Adaptive Slepian-Wolf Decoding Based On Expectation Propagation

Lijuan Cui, Shuang Wang and Samuel Cheng

***Abstract*—A major difficulty that plagues the practical use of Slepian-Wolf (SW) coding (and distributed source coding in general) is that the precise correlation among sources needs to be known *a priori*. However, belief propagation (BP) algorithm cannot adapt efficiently to the statistical change of the correlation. This paper proposes an adaptive SW decoding scheme which can perform online time-varying correlation estimation at the bit-level by incorporating expectation propagation (EP) algorithm. Moreover, we compare the proposed EP-based approach with Monte Carlo method using particle filtering (PF) algorithm. Our results show that the proposed EP estimator obtains the comparable estimation accuracy with less computational complexity than the PF method.**

***Index Terms*—Adaptive decoding, Distributed algorithms, Source coding, Data compression**

I. INTRODUCTION

Slepian-Wolf (SW) coding is a technique to losslessly compress correlated remote sources separately and decompress them jointly [1]. Numerous channel coding based SW coding schemes have been proposed [2], [3]. However, the fundamental assumption is that the correlation statistics needs to be known accurately *a priori*. Actually in many real-world applications, such as sensor networks, the correlation statistics among sensors cannot be obtained easily and may vary over both space and time. Since the decoding performance of distributed source coding (DSC) relies on the knowledge of correlation significantly, the design of an online correlation estimation scheme becomes an important research topic both in theoretical study and practical applications [4]–[6].

Here we consider asymmetric SW coding of two binary correlated sources X and Y, where the relationship is modeled as a virtual binary symmetric channel (BSC) with a time-varying crossover probability. For the crossover probability estimation, many algorithms were studied in the literature. In [4], the residual redundancies in LDPC syndromes are used to estimate the crossover probability between two correlated binary sources using Mean-Intrinsic-LLR. However, this algorithm works only for highly correlated sources. In [5], [6], the expectation maximization (EM) algorithm was used to estimate the crossover probability. However, the crossover probability is assumed to be constant and does not change within each codeword block. In [7] we considered adaptive correlation estimation with a single factor graph. The algorithm can handle sources with both weak and strong correlations and the statistics may vary dynamically. However, since the correlation parameter is continuous and cannot be parametrized in general, we incorporated Monte Carlo step into standard BP. The resulting particle based BP (PBP) algorithm handles correlation estimation well but with significant computational overhead as the computational complexity of BP algorithm increases exponentially with the alphabet size.

In this letter, we consider another possible workaround for approximate inference. Instead of using Monte Carlo techniques, we explore the possibility of deterministic approximation, in particular, through the use of expectation propagation (EP). Comparing to Monte Carlo techniques, deterministic approximation typically is much faster but is less flexible. It may not work for all problems and is also more mathematically involved. In the following, we demonstrate how EP can be used for adaptive SW decoding. Moreover, we compare the performance of the EP estimator with PBP estimator [7]–[9] under the same setup. Our simulation results show that the proposed EP estimator obtains the comparable estimation accuracy with less computational complexity comparing to PBP. Further, the EP estimator does not depend on the initial estimation of crossover probability and offers a good real-time estimation for the crossover probability. As a result, a lower decoding error rate is possible comparing to a standard BP-based SW decoder.

II. FACTOR GRAPH CONSTRUCTION OF SW DECODER WITH CROSSOVER PROBABILITY ESTIMATION

A factor graph for both SW decoding and correlation tracking is illustrated in Fig. 1. The factor graph is more or less the same as that used in our prior work [7]. Note that regions II and III in Fig. 1 alone contribute the same factor graph as that used in traditional LDPC-based SW decoding. A block of X $(x_1, x_2, \cdots, x_N)$ is compressed into M syndrome bits $s_1, s_2, \cdots, s_M$, thus resulting in $M : N$ compression. Here, x_i and y_i are realizations of nodes X_i and Y_i, respectively. In Region III, syndrome factor nodes h_k, $k = 1, 2, \ldots, M$ take into account the constraints imposed by the received syndrome bit s_k. In Region II, factor node f_i plays a role of providing a predetermined likelihood $p(y_i|x_i, \rho)$ to variable node X_i for LDPC-based SW decoding, where ρ denotes the crossover probability. According to the relationship between x_i and y_i in BSC, the corresponding factor function of f_i is defined as

$$f_i(\rho, x_i, y_i) = \rho^{x_i \oplus y_i}(1-\rho)^{1 \oplus x_i \oplus y_i}, \quad (1)$$

where $\oplus$ is the bitwise sum of two elements.

However, in reality, crossover probability may vary over time, denoted by ρ_t, and the perfect knowledge of time-varying crossover probability may not always be available at the decoder. Thus, in the case without feedback channels, it is necessary to perform an online crossover probability estimation to avoid the degradation of decoding performance. It also means that each factor node f_i will periodically update the likelihood $p(y_i|x_i, \rho_t)$ for the respective bit variable node X_i when a new crossover probability estimate of ρ_t is available, instead of using a predetermined likelihood $p(y_i|x_i, \rho)$.

Lijuan Cui, Shuang Wang, Samuel Cheng are with the School of Electrical and Computer Engineering, University of Oklahoma, Tulsa, OK 74135. E-mail:{lj.cui, shuangwang, samuel.cheng}@ou.edu

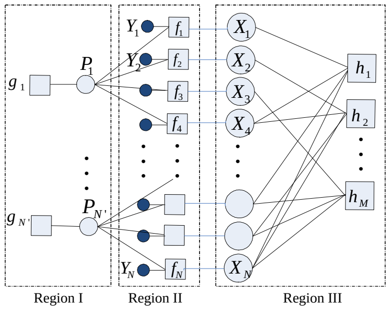

Fig. 1. Factor graph of adaptive SW decoding.

To enable the online estimation of time-varying crossover probability ρ_t, we introduce extra variable nodes P_j and factor nodes g_j, $j = 1, 2, \ldots, N'$ (see Region I of Fig. 1). Here, we call the number of factor nodes in Region II connecting to each variable node P_j the connection ratio C [1], which is equal to four in Fig. 1. In Region I, each variable node P_j is used to model the time-varying crossover probability ρ_t of a block of C number of code bits. Moreover, the factor function $g_j(\rho_j)$ of factor node g_j corresponds to the *a priori* distribution for variable ρ_j and will be discussed later. Consequently, by introducing crossover probability estimation in Region I, likelihood factor function (1) will be updated to

$$f_i(\rho_j, x_i, y_i) = \rho_j^{x_i \oplus y_i}(1-\rho_j)^{1 \oplus x_i \oplus y_i}. \tag{2}$$

III. Posterior approximation on factor graph

In Bayesian inference, the estimation of crossover probability ρ_j corresponds to the estimation of its posterior distribution, i.e. $p(\rho_j|\mathbf{y}_j)$, where $\mathbf{y}_j = (y_i | i \in \mathcal{N}^{\backslash g_j}(P_j))$, and $\mathcal{N}^{\backslash g_j}(P_j)$ represents the set of all neighbors' indices for a variable node P_j except the index of g_j. According to the Bayes' rule, the posterior distribution over variable ρ_j in Fig. 1 can be written as

$$\begin{aligned} p(\rho_j|\mathbf{y}_j) &= \frac{1}{Z_j} \prod_{i \in \mathcal{N}^{\backslash g_j}(P_j)} p(\rho_j) p(y_i|\rho_j) \\ &= \frac{1}{Z_j} \prod_{i \in \mathcal{N}^{\backslash g_j}(P_j)} \int_{x_i} p(\rho_j) p(x_i) p(y_i|x_i, \rho_j) \\ &= \frac{1}{Z_j} g(\rho_j) \prod_{i \in \mathcal{N}^{\backslash g_j}(P_j)} \sum_{x_i} f(\rho_j, x_i, y_i) m_{X_i \to f_i}(x_i), \end{aligned} \tag{3}$$

where $Z_j = \int_{\rho_j} \prod_{i \in \mathcal{N}^{\backslash g_j}(P_j)} p(\rho_j)p(y_i|\rho_j)$ is a normalization constant, $p(\rho_j) = g_j(\rho_j)$, $p(y_i|x_i, \rho_j) = f(\rho_j, x_i, y_i)$, the *a priori* distribution $p(x_i)$ is captured by the message $m_{X_i \to f_i}(x_i)$ with binary sources x_i taking 0 or 1 defined in [10]. Moreover, according to message passing algorithm [10], the posterior distribution (3) can be written as

$$p(\rho_j|\mathbf{y}_j) = \frac{1}{Z_j} m_{g_j \to P_j}(\rho_j) \prod_{i \in \mathcal{N}^{\backslash g_j}(P_j)} m_{f_i \to P_j}(\rho_j). \tag{4}$$

However, direct evaluation of the posterior distribution through (4) would be infeasible, since the message $m_{f_i \to P_j}(\rho_j) = \sum_{x_i \in [0,1]} f(\rho_j, x_i, y_i) m_{X_i \to f_i}(x_i)$ has two terms and the product of all the messages $\prod_{i \in \mathcal{N}^{\backslash g_j}(P_j)} m_{f_i \to P_j}(\rho_j)$ has total 2^C number of terms, where $C = |\mathcal{N}^{\backslash g_j}(P_j)|$ is the connection ratio, and it can be a large number (e.g. 50 to $10,000$ in our study). To solve this problem, we resort to EP algorithm to approximate the posterior in the following section.

[1]To estimate a stationary crossover probability, we can set the connection ratio equal to the code length.

A. Posterior Approximation Using EP

Briefly speaking, EP algorithm attempts to seek an approximate posterior distribution restricted in exponential family that can be very close to the true posterior distribution through minimizing the Kullback Leibler (KL) divergence between the true distribution and the approximation [11]. In our problem, we assume an approximation to $p(\rho_j|\mathbf{y}_j)$ of (4) in the form $q(\rho_j|\mathbf{y}_j) = \frac{1}{Z_j} \tilde{m}_{g_j \to P_j}(\rho_j) \prod_{i \in \mathcal{N}^{\backslash g_j}(P_j)} \tilde{m}_{f_i \to P_j}(\rho_j)$, where each original message $m_{f_i \to P_j}(\rho_j)$ in (4) is replaced by an approximating message $\tilde{m}_{f_i \to P_j}(\rho_j)$ belonging to a tractable distribution in exponential family. The approximation of each message $\tilde{m}_{f_i \to P_j}(\rho_j)$ is achieved by minimizing KL divergence in the context of all the remaining messages.

Note that Beta distribution is defined as

$$\text{Beta}(x, \alpha, \beta) = \frac{1}{\text{beta}(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}, \tag{5}$$

where α and β are shape parameters, $\text{beta}(\alpha, \beta)$ represents Beta function. By comparing (2) with Beta distribution (5), likelihood function (2) can be represented in terms of Beta distribution with respect to variable ρ_j as parameter:

$$\begin{aligned} f_i(\rho_j, x_i, y_i) = &\ \text{beta}\left((x_i \oplus y_i) + 1, (1 \oplus x_i \oplus y_i) + 1\right) \\ &\times \text{Beta}\left(\rho_j, (x_i \oplus y_i) + 1, (1 \oplus x_i \oplus y_i) + 1\right). \end{aligned} \tag{6}$$

The original EP algorithm proposed by Minka is applied to approximate a mixture of Gaussian distributions in clutter problem [11]. In this paper, our problem is to estimate time-varying crossover probability given observations from neighboring factor nodes (see (3) and (6)). Thus, EP is extended to approximate a mixture of Beta distributions instead of Gaussian distributions in our problem.

In addition, to approximate a posterior distribution of crossover probability, it is computationally favorable to choose a conjugate prior for the likelihood function based on Bayesian theorem. Since Beta distribution is the conjugate prior for itself, we choose $g(\rho_j) = \text{Beta}(\rho_j, \alpha_j^0, \beta_j^0)$ as the prior distribution with the prior parameter α_j^0 and β_j^0.

EP algorithm processes in the following. For the ease of exposition, we denote by $q(\rho_j)$ the approximated posterior distribution instead of $q(\rho_j|\mathbf{y}_j)$ in the rest of this letter.

1. Initialize the prior messages for the crossover probability variables

$$g_j(\rho_j) = \text{Beta}(\rho_j, \alpha_j^0, \beta_j^0) = z_j^0 \rho_j^{\alpha_j^0 - 1}(1-\rho_j)^{\beta_j^0 - 1}, \tag{7}$$

with $\alpha_j^0 = 2$, $\beta_j^0 = \frac{\alpha_j^0 - 1}{\rho^0} + 2 - \alpha_j^0$ and $z_j^0 = \frac{1}{\text{beta}(\alpha_j^0, \beta_j^0)}$, where ρ^0 is the initial crossover probability for SW decoding, β_j^0 and α_j^0 are shape parameters for Beta distribution.

2. Initialize the likelihood messages from the channel output

$$\tilde{m}_{f_i \to P_j}(\rho_j) = z_{ij} \rho_j^{\alpha_{ij} - 1}(1-\rho_j)^{\beta_{ij} - 1} \tag{8}$$

with $\beta_{ij} = 1$, $\alpha_{ij} = 1$ and $z_{ij} = 1$, where the values selection for the above parameters guarantee that ρ_j in (8) is equality likely before LDPC decoding.

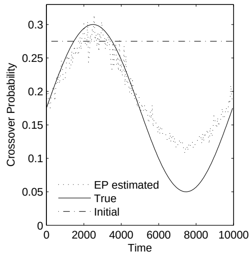

Fig. 2. Estimation of time-varying crossover probability.

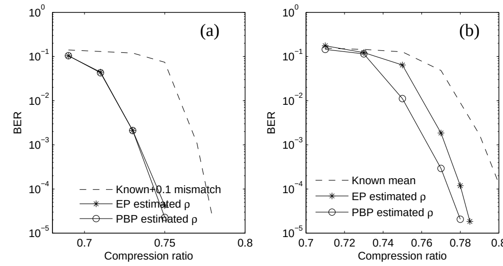

Fig. 3. BER vs. compression ratio for SW decoder with (a) constant crossover probability and (b) time-varying crossover probability, respectively.

3. Initialize the posterior probability distributions of the crossover probability variables

$$q(\rho_j) = Z_j \rho_j^{\alpha_j^{\text{new}}-1}(1-\rho_j)^{\beta_j^{\text{new}}-1}, \tag{9}$$

with $\alpha_j^{\text{new}} = \alpha_j^0$, $\beta_j^{\text{new}} = \beta_j^0$ and $Z_j = z_j^0$.
4. Until all $\tilde{m}_{f_i \to P_j}(\rho_j)$ converge:
For each variable node P_j
For each factor node f_i, where $i \in \mathcal{N}^{\setminus g_j}(P_j)$
4.1 Remove $\tilde{m}_{f_i \to P_j}(\rho_j)$ from the posterior $q(\rho_j)$

$$\alpha_j^{\text{tmp}} = \alpha_j^{\text{new}} - (\alpha_{ij} - 1); \ \beta_j^{\text{tmp}} = \beta_j^{\text{new}} - (\beta_{ij} - 1) \tag{10}$$

4.2 Update α_j^{new} and β_j^{new} according to moment matching

$$\alpha_j^{\text{new}} = \frac{m_1(m_1 - m_2)}{m_2 - m_1^2}; \ \beta_j^{\text{new}} = \alpha_j^{\text{new}}(\frac{1}{m_1} - 1) \tag{11}$$

$$m_1 = \frac{(\alpha' + y) + \beta'^{(2y-1)} \left(\alpha'(\alpha'+1)\right)^{(1-y)} Lr(x)}{(\alpha' + \beta' + 1)\left(1 + (\frac{\beta'}{\alpha'})^{(2y-1)} Lr(x)\right)}, \tag{12}$$

$$m_2 = \frac{(\alpha'+1)\left(\alpha' + 2y + \left(\alpha'(\alpha'+2)\right)^{(1-y)} \beta'^{(2y-1)} Lr(x)\right)}{(\alpha'+\beta'+2)(\alpha'+\beta'+1)\left(1 + (\frac{\beta'}{\alpha'})^{(2y-1)} Lr(x)\right)}, \tag{13}$$

where for simplify notations, $\alpha' = \alpha_j^{\text{tmp}}$, $\beta' = \beta_j^{\text{tmp}}$, and $Lr(x) = \frac{m_{X_i \to f_i}(1)}{m_{X_i \to f_i}(0)}$.
4.3 Set approximated message

$$\begin{aligned} \alpha_{ij} &= \alpha_j^{\text{new}} - (\alpha_j^{\text{tmp}} - 1); \ \beta_{ij} = \beta_j^{\text{new}} - (\beta_j^{\text{tmp}} - 1); \\ z_{ij} &= Z_j \frac{\text{beta}(\alpha_j^{\text{tmp}}, \beta_j^{\text{tmp}})\text{beta}(\alpha_{ij}, \beta_{ij})}{\text{beta}(\alpha_j^{\text{new}}, \beta_j^{\text{new}})}. \end{aligned} \tag{14}$$

IV. Results

First, we study the performance of the proposed EP estimator for time-varying crossover probability. Fig. 2 shows the estimated results of a sinusoidally changing correlation, where the crossover probability changes sinusoidally from 0.05 to 0.3, $N = 10,000$, $C = 50$, and ρ_0 is 0.1 above the mean of true crossover probability. It can be seen that the proposed EP estimator achieves the comparable estimation accuracy with the PBP estimator, even though the initial crossover probability is far away from the mean of the time-varying crossover probability.

Then, we study the decoding performances of SW decoder with and without EP/PBP estimator in terms of constant crossover probability (see Fig. 3(a)) and time-varying crossover probability (see Fig. 3(b)), respectively. The following results are obtained over 10,000 trails with $N = 10,000$. The EP/PBP estimator starts working after 50 BP iterations and then is performed periodically every 20 BP iterations until BP decoder successfully decodes the codeword or reaches its maximum number of iterations (i.e. 150 in our experiment). From Fig. 3(a), we can see that there is no obvious difference of performance between the proposed EP based decoder and PBP decoder for constant crossover probability, and we observe a small performance gap between EP based decoder and PBP decoder for time-varying crossover probability in Fig. 3(b). On the other hand, the decoding performance of SW decoders with EP/PBP estimator are significantly better than that of the standard BP-based SW decoder in both Fig. 3(a) and 3(b). It is important to point out that the additional computational overhead of the proposed EP based decoder is less than 10% of the standard BP-based SW decoder. In the following comparison, the proposed EP and PBP algorithms are implemented in MATLAB with Java, and executed on an intel i7 CPU with 100 trails. In our simulation, EP based decoder requires about 68s, while PBP decoder needs 814s. We can see that EP based decoder is much faster than PBP based decoder. Thus, we believe that proposed SW decoder with EP estimator is a practical improved alternative of the standard BP-based SW decoder and PBP decoder.

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: design and construction," in *Proc. DCC*, 1999, pp. 158–167.

[3] A. Yedla, H. Pfister, and K. Narayanan, "Universality for the noisy slepian-wolf problem via spatial coupling," *Arxiv preprint arXiv:1105.6374*, 2011.

[4] Y. Fang, "Crossover probability estimation using mean-intrinsic-LLR of LDPC syndrome," *IEEE Commun. Lett.*, vol. 13, no. 9, p. 679, 2009.

[5] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Maximum Likelihood BSC parameter estimation for the Slepian-Wolf problem," *IEEE Commun. Lett.*, vol. 15, no. 2, pp. 232–234, 2011.

[6] A. Zia, J. Reilly, and S. Shirani, "Distributed Parameter Estimation with Side Information: A Factor Graph Approach," in *IEEE International Symposium on Information Theory, 2007*, pp. 2556–2560.

[7] L. Cui, S. Wang, S. Cheng, and M. Yeary, "Adaptive binary Slepian-Wolf decoding using particle based belief propagation," *IEEE Transactions on Communications*, vol. 59, no. 9, pp. 2337 – 2342, 2011.

[8] L. Cui, S. Wang, and S. Cheng, "Adaptive Distributed Source Coding over Erasure Channels using Particle-based Belief Propagation," in *IEEE Global Telecommunications Conference*, 2010.

[9] S. Wang, L. Cui, S. Cheng, Y. Zhai, M. Yeary, and Q. Wu, "Noise adaptive LDPC decoding using particle filtering," *IEEE Transactions on Communications*, vol. 25, no. 99, pp. 1–4, 4 2011.

[10] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[11] T. Minka, "Expectation propagation for approximate Bayesian inference," in *Uncertainty in Artificial Intelligence*, vol. 17, 2001, pp. 362–369.