# ADAPTIVE SLEPIAN-WOLF DECODING USING LAPLACE PROPAGATION

*Lijuan Cui, Shuang Wang, Samuel Cheng* *

School of Electrical and Computer Engineering
University of Oklahoma
Tulsa, OK, 74135

*Lina Stankovic, Vladimir Stankovic*

University of Strathclyde
Dept. Electronic and Electrical Engineering
Glasgow G11XW, UK

## ABSTRACT

Accurately estimating correlation between sources has significant impact on the performance of Slepian-Wolf (SW) coding. In this paper, we propose a low complexity estimator based on Laplace propagation for exploiting the source correlation at the decoder side, by modeling the correlation estimation as a Bayesian inference problem. Through simulations, we show that the proposed algorithm can simultaneously reconstruct a compressed source and estimate both stationary and time-varying joint correlation between the sources at the bit level. Furthermore, comparing to the conventional SW decoder, the proposed approach can achieve a better decoding performance under varying correlation statistics and the proposed estimator shows a very fast convergence speed and low complexity compared with state-of-the-art sampling approaches.

***Index Terms*—** Source coding, Adaptive decoding

## 1. INTRODUCTION

Slepian-Wolf (SW) coding refers to lossless independent compression and joint decompression of correlated sources [1]. Numerous SW coding schemes have been proposed (e.g., [2, 3]) based on advanced channel coding. However, the fundamental assumption in these designs is that the correlation statistics needs to be known accurately *a priori* at the decoder side. However, in many applications, such as sensor networks, the correlation statistics cannot be obtained easily and may vary over both space and time. Since decoding performance of distributed source coding (DSC) relies heavily on the knowledge of correlation, the design of an online correlation estimation scheme becomes an important research topic both theoretically and practically [4, 5, 6].

The estimation of the correlation statistics can be depicted as a learning or an inference problem using graphical models (our previous works, e.g. [7]), especially the factor graph. In context of Bayesian inference, such a process is equivalent to computing the posterior distribution of correlation parameter. Therefore, given the side information (SI), $Y$, the posterior distribution of correlation parameter $\rho$, (that models the correlation between source $X$ and SI $Y$) can be written as

$$p(\rho|Y) = \frac{p(\rho)p(Y|\rho)}{p(Y)}, \tag{1}$$

where $p(\rho)$ is the prior distribution of the correlation parameter $\rho$, $p(Y|\rho)$ is likelihood function, and $p(Y)$ is the model evidence/normalization factor, which could be expressed in the following marginalization form

$$p(Y) = \int p(\rho)p(Y|\rho)d\rho. \tag{2}$$

Unfortunately, the exact posterior distribution (1) and the normalization factor (2) are computationally tractable only in few special cases, such as when (i) the prior distribution $p(\rho)$ and likelihood function $p(Y|\rho)$ are both Gaussian (e.g., using Kalman filter) or (ii) the unknown $\rho$ is a discrete variable with small alphabet size (e.g., using belief propagation (BP)). Unfortunately, in practice, the source correlation $\rho$ is usually a non-Gaussian distributed real variable, and thus approaches are needed to find a good approximate solution.

During the past decade, there have been many different approximate inference methods proposed in the research community. In general, these approximation methods can been summarized into the following two categories: stochastic and deterministic approximations. Stochastic techniques, such as sequential Monte Carlo (also known as the sampling method or particle filters (PF)), can be adopted and applied to most scenarios although they are computational demanding. Deterministic techniques, including the Laplace's method, Expectation propagation, and variational approximations, provide some low complexity alternatives based on analytical approximations. Compared with Monte Carlo techniques, deterministic approximation typically is much faster but is less flexible, since it can only approximate unimodal posterior distributions. Moreover, it is also more mathematically involved.

Although a key DSC-design requirement is a low complexity encoder, many applications demand fast and low complexity decoding. Therefore, the study of low complexity online estimation schemes is becoming increasingly important and a viable solution is deterministic approximation. Among

*This work was supported in part by NSF under grant CCF 1117886

existing deterministic approximation methods, Laplace propagation (LP) algorithm offers a ultra-low computational complexity. In this paper, we propose an online correlation estimation scheme based on LP, which is carried out jointly with decoding of a factor graph-based SW code. In addition, we compare the performance of the proposed LP estimator with sampling based estimator [11]. Our simulation results show that LP estimator offers a ultra-low computational complexity, as well as maintains a comparable estimation performance.

## 2. RELATED WORK

In this paper we consider the asymmetric SW coding problem (or source coding with SI), where the relationship between two binary correlated sources $X$ and $Y$ is modeled as a virtual binary symmetric channel (BSC). For the BSC crossover probability estimation, several algorithms have been studied in the literature. In [4], the residual redundancies in low-density parity-check (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, and the crossover probability is assumed to be constant and does not change within each codeword block. In [7], the authors considered the problem of adaptive correlation estimation based on factor graph, which can handle sources with both weak and strong correlations and the correlation statistics may vary dynamically. However, the proposed scheme in [7] is based on the stochastic approximation method, which involves a significant computational overhead at the decoder. To reduce the impact of decoding delay, it is necessary to keep the estimation complexity as low as possible, but without sacrificing decoding performance (i.e., estimation accuracy). This is what motivates the work in this paper, where we explore an LP based correlation estimator for the SW decoding problem.

## 3. THE PROPOSED SCHEME

### 3.1. Factor graph construction of the SW decoder with crossover probability estimation

We consider the asymmetric SW coding case with two binary correlated sources $X$ and $Y$, where the correlation is modeled as a virtual BSC with time-varying crossover probability.

The traditional LDPC-based SW decoding factor graph is shown in Fig. 1 (Regions II and III). Here, a block $(x_1, x_2, \cdots, x_N)$ is compressed into $M$ syndrome bits $s_1, s_2, \cdots, s_M$, thus resulting in $M : N$ compression, and $x_i$ and $y_i$ are realizations of $X_i$ and $Y_i$, respectively. Syndrome factor nodes $h_k$, $k = 1, 2, \ldots, M$ shown in Region III take into account the constraints imposed by the received syndrome bit $s_k$. Factor node $f_i$ shown in Region II plays the role of providing a predetermined likelihood $p(y_i|x_i, \rho)$ to variable node $X_i$ for LDPC-based SW decoding, where $\rho$ 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

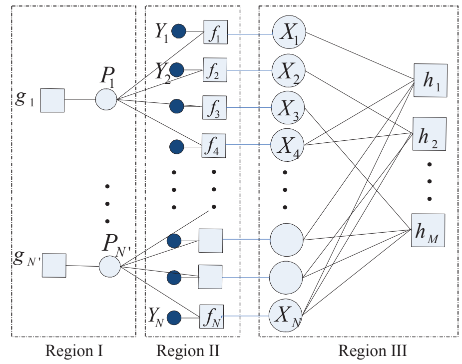


**Fig. 1**. Factor graph of adaptive SW decoding.

$$f_i(\rho, x_i, y_i) = \rho^{x_i \oplus y_i}(1-\rho)^{1 \oplus x_i \oplus y_i}, \tag{3}$$

where $\oplus$ is the bitwise sum of two elements. With the defined factor functions, source $X$ can be decoded by performing standard BP on Regions II and III in Fig. 1.

In practice, however, it is not easy to obtain perfect knowledge of crossover probability at the decoder side. In addition, another important practical issue is that the crossover probability may vary over time, i.e, $\rho(t) = \rho_t$. Thus, in the case without feedback channels, it is imperative 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 corresponding bit variable node $X_i$ when a new crossover probability estimate of $\rho_t$ is available, instead of using a predetermined likelihood.

To enable the online estimation of time-varying crossover probability $\rho_t$, 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. Each variable node $P_j$ in Region I is used to model the time-varying crossover probability $\rho_t$ of a block of $C$ code bits. Moreover, the factor function $g_j(\rho_j)$ of factor node $g_j$ corresponds to the *a priori* distribution for variable $\rho_j$. Consequently, by introducing the crossover probability estimation in Region I, the likelihood factor function (3) becomes:

$$f_i(\rho_j, x_i, y_i) = \rho_j^{x_i \oplus y_i}(1-\rho_j)^{1 \oplus x_i \oplus y_i}. \tag{4}$$

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

### 3.2. Posterior approximation of crossover probability

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

$$\begin{aligned} p(\rho_j|\mathbf{y}_j) &= \frac{1}{Z_j} \prod_{i \in \mathcal{N}^{\setminus g_j}(P_j)} p(\rho_j) p(y_i|\rho_j) \\ &= \frac{1}{Z_j} \prod_{i \in \mathcal{N}^{\setminus g_j}(P_j)} \int_{x_i} p(\rho_j) p(x_i) p(y_i|x_i, \rho_j) \\ &= \frac{1}{Z_j} g(\rho_j) \prod_{i \in \mathcal{N}^{\setminus g_j}(P_j)} \sum_{x_i} f_i(\rho_j, x_i, y_i) m_{X_i \to f_i}(x_i), \end{aligned} \tag{5}$$

where $Z_j = \int_{\rho_j} \prod_{i \in \mathcal{N}^{\setminus g_j}(P_j)} p(\rho_j) p(y_i|\rho_j)$ is a normalization constant, $p(\rho_j) = g_j(\rho_j)$, $p(y_i|x_i, \rho_j) = f_i(\rho_j, x_i, y_i)$, 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 [8]. Moreover, according to message passing algorithm [8], the posterior distribution (5) can be written as

$$p(\rho_j|\mathbf{y}_j) = \frac{1}{Z_j} m_{g_j \to P_j}(\rho_j) \prod_{i \in \mathcal{N}^{\setminus g_j}(P_j)} m_{f_i \to P_j}(\rho_j). \tag{6}$$

However, to infer the parameter $\rho_j$, direct evaluation of the posterior distribution through (6) is infeasible, since the message

$$m_{f_i \to P_j}(\rho_j) = \sum_{x_i \in [0,1]} f(\rho_j, x_i, y_i) m_{X_i \to f_i}(x_i) \tag{7}$$

has two terms and the product of all the messages

$$\prod_{i \in \mathcal{N}^{\setminus g_j}(P_j)} m_{f_i \to P_j}(\rho_j) \tag{8}$$

has a total of terms $2^C$, where $C$ can be a large number. In the following section, we resort to an approximate inference LP algorithm to solve this problem.

## 4. LP FOR POSTERIOR APPROXIMATION

### 4.1. LP algorithm

Laplace's method [9], also known as the saddle-point approximation, approximates a density function $p(\theta|D)$ with parameter $\theta$ by a Gaussian $q(\theta|D)$ around its peak, where $D$ is a set of observations. The mean and variance of the Gaussian are approximated, respectively, as

$$\begin{aligned} m &= \text{argmax}_\theta \log p(\theta|D) =: \hat{\theta} \\ v^{-1} &= -\partial_\theta^2 \log p(\theta|D)|_{\theta=\hat{\theta}}. \end{aligned} \tag{9}$$

This method, when applied directly to our problem, results in a large computation cost. However, if the possibility density function can be written as a product of several terms $p(\theta|D) = \prod_i f_i(\theta)$, each of which only contains a small number of variables (e.g., $p(\rho_j|\mathbf{y}_j)$ in (6)), then the approximate distribution $q(\theta)$ of $p(\theta|D)$ can be obtained by iteratively finding the approximate of each term. Here, the approximate $\tilde{f}_i(\theta)$ of each true term $f_i(\theta)$ is achieved by performing Laplace approximate on $q_i(\theta) = f_i(\theta) \prod_{j \neq i} \tilde{f}_j(\theta)$ for all $i$ until convergence. This is referred to as the LP algorithm [10].

### 4.2. Prior function

In our problem of the crossover probability estimation, the LP algorithm needs to optimize each message term of the posteriori in (6). The message $m_{g_j \to P_j}(\rho_j)$ corresponds to the prior distribution of $\rho_j$ defined by the prior function $g(v_j)$. If the prior probability distribution and posterior distribution are in the same exponential family, which is named conjugate prior for the likelihood, it is computationally favorable to obtain the posterior. Thus, we choose a conjugate prior for the likelihood function (4).

Likelihood function (4) can be represented in terms of Beta distribution Beta$(x, \alpha, \beta)$ with respect to variable $\rho_j$ as parameter:

$$\begin{aligned} f_i(\rho_j, x_i, y_i) = \;& \text{beta}\left((x_i \oplus y_i) + 1, (1 \oplus x_i \oplus y_i) + 1\right) \\ & \times \text{Beta}\left(\rho_j, (x_i \oplus y_i) + 1, (1 \oplus x_i \oplus y_i) + 1\right), \end{aligned} \tag{10}$$

$$\text{Beta}(x, \alpha, \beta) = \frac{1}{\text{beta}(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}, \tag{11}$$

where $\alpha$ and $\beta$ are shape parameters and beta$(\alpha, \beta)$ is Beta function. 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 shape parameters $\alpha_j^0$ and $\beta_j^0$. Here, we let $\alpha_j^0 = 2$ and $\beta_j^0 = \frac{\alpha_j^0 - 1}{\rho^0} + 2 - \alpha_j^0$, the values which guarantee the mode of prior distribution to be equal to the initial crossover probability $\rho_j^0$.

### 4.3. LP for posterior approximation

The proposed LP algorithm for crossover probability estimation is shown next. Here, we drop the dependency of $\rho_j$ on $\mathbf{y}_j$ in (6) for simplicity.

1. Initialize the prior term

$$\tilde{m}_{g_j \to P_j}(\rho_j) = \mathcal{N}(\rho_j, m_j^0, v_j^0) \tag{12}$$

with

$$\begin{aligned} m_j^0 &= \frac{\alpha_j^0 - 1}{\alpha_j^0 + \beta_j^0 - 2}, z_j^0 = \frac{1}{\sqrt{2\pi v_j^0}} \\ v_j^0 &= \frac{(-1 + m_j^0)^2 (m_j^0)^2}{-1 + \alpha_j^0 (-1 + m_j^0)^2 + 2m_j^0 + (-2 + \beta_j^0)(m_j^0)^2} \end{aligned} \tag{13}$$

Initialize the approximation term

$$\tilde{m}_{f_i \to P_j}(\rho_j) = \mathcal{N}(\rho_j, m_{ij}, v_{ij}) \tag{14}$$

with $m_{ij} = 0$, $v_{ij} = \infty$, $z_{ij} = 1$.

2. Initialize the mean and variance of the approximate normal distribution of posterior $q(\rho_j)$

$$m_j^{\text{new}} = m_j^0, v_j^{\text{new}} = v_j^0 \tag{15}$$

3. For each variable node $P_j$

For each factor node $f_i$, where $i \in \mathcal{N}(P_j)$

(a) Remove $\tilde{m}_{f_i \to P_j}(\rho_j)$ from the posterior $q(\rho_j)$, we get $q^{\setminus P_j}(\rho_j) = \mathcal{N}(\rho_j, m_j^{\text{tmp}}, v_j^{\text{tmp}})$

$$\begin{aligned} v_j^{\text{tmp}} &= \left(\frac{1}{v_j^{\text{new}}} - \frac{1}{v_{ij}}\right)^{-1} \\ m_j^{\text{tmp}} &= v_j^{\text{tmp}}\left(\frac{m_j^{\text{new}}}{v_j^{\text{new}}} - \frac{m_{ij}}{v_{ij}}\right) \end{aligned} \tag{16}$$

(b) Update $q^{\text{new}}(\rho_j)$ according to the Laplace's method and get $m_j^{\text{new}}$ and $v_j^{\text{new}}$

$$\begin{aligned} v_j^{\text{new}} &= \left(v_c + \frac{1}{v_j^{\text{tmp}}}\right)^{-1}, Z_j = \frac{1}{\sqrt{2\pi v_j^{\text{new}}}} \\ m_j^{\text{new}} &= \frac{m_j^{\text{tmp}} + \frac{(-Lr(x))^{1-y}}{1-Lr(x)} \pm \sqrt{\triangle}}{2} \end{aligned} \tag{17}$$

where $v_c = \frac{(Lr(x)-1)^2}{(Lr(x)^{(1-y)}(-1+m_j^{\text{new}}) - Lr(x)^y m_j^{\text{new}})^2}$, $\triangle = (m_j^{\text{tmp}} - \frac{(-Lr(x))^{1-y}}{1-Lr(x)})^2 + 4v_j^{\text{tmp}}$, and the $\pm$ for $\sqrt{\triangle}$ is decided by the values of $y$ and $Lr(x) = \frac{m_{X_i \to f_i}(0)}{m_{X_i \to f_i}(1)}$.

(c) Set approximated message

$$\begin{aligned} v_{ij} &= \left(\frac{1}{v_j^{\text{new}}} - \frac{1}{v_j^{\text{tmp}}}\right)^{-1} \\ m_{ij} &= v_{ij}\left(\frac{m_j^{\text{new}}}{v_j^{\text{new}}} - \frac{m_j^{\text{tmp}}}{v_j^{\text{tmp}}}\right) \\ z_{ij} &= Z_j \frac{1}{\sqrt{v_{ij}}} \frac{v_j^{\text{tmp}}}{\sqrt{v_j^{\text{tmp}} - v_j^{\text{new}}}} \exp\frac{(m_j^{\text{new}} - m_j^{\text{tmp}})^2}{2(v_j^{\text{tmp}} - v_j^{\text{new}})} \end{aligned} \tag{18}$$

## 5. RESULTS

In this section, the decoding performance of standard BP, LP based BP decoders, and particle based BP (PBP) decoder in [11] are presented in the presence of a crossover probability mismatch. Here, we consider two different scenarios, constant crossover probability mismatch and time-varying crossover probability.

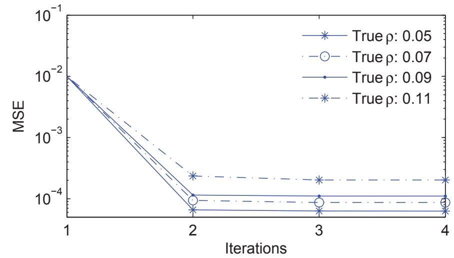


**Fig. 2**. The performance of the LP estimator for different values of constant crossover probability.

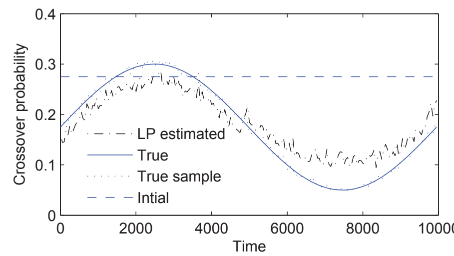


**Fig. 3**. Estimation of the time-varying crossover probability using the proposed LP estimator .

We first study the estimation accuracy of the proposed LP estimator for constant crossover probability in Fig. 2. In this case, SW coding based on a regular LDPC code with code rate $R = 0.7$ and code length $N = 1000$ is used in our simulation. Since the constant crossover probability is considered, there is only one variable node in Region I, which means that the connection ratio $C$ is set to 1000. All the results are obtained over 100 trials. Initial crossover probabilities used for BP decoding are always for 0.1 above true crossover probabilities. The maximum number of iterations for BP decoding is equal to 50. Since we only focus on estimation accuracy, the LP estimator is only used once at the end of the BP decoding. To achieve the best decoding performance, new estimates can be sent back to the SW decoder periodically, and this setup will be studied later in this section. We can see from Fig. 2 that the LP estimator always converges within 3 or 4 iterations.

Second, we study the performance of the proposed LP estimator for time-varying crossover probability. Fig. 3 shows the estimate of a sinusoidally changing crossover probability, where $N = 10,000$, $C = 50$, the minimum and maximum value of sinusoidal signal are 0.05 and 0.3, respectively, and the initial value $\rho_0 = 0.25$ is for 0.1 above the mean of true crossover probability. It can be seen that the proposed LP estimator provides an accurate estimate even though the initial crossover probability is far away from the mean.

Finally, in Figs. 4 and 5, we study the decoding performance of an LDPC based SW decoder with and without LP/PBP estimator. The following results are obtained over 10,000 trials and with the codeword length of $10^4$ bits. The LP/PBP estimator starts after 50 BP iterations and is applied

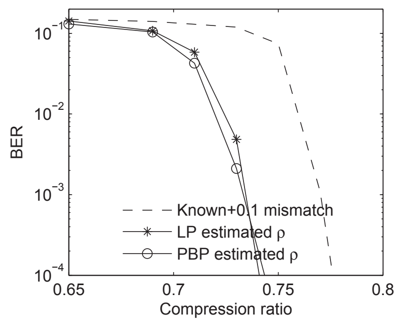


**Fig. 4**. Bit error rate (BER) vs. compression ratio for the LDPC-based SW decoder using 1) BP assuming crossover probability for $0.1$ above the true value; 2) LP-based BP; 3) PBP decoder for a constant crossover probability.

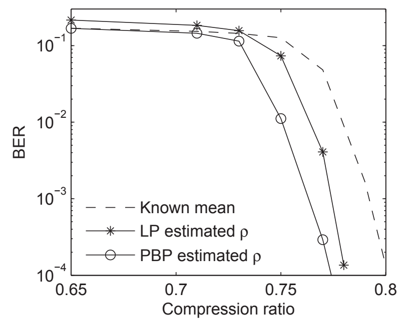


**Fig. 5**. Bit error rate (BER) vs. compression ratio for the LDPC-based SW decoder using 1) BP assuming mean crossover probability; 2) LP-based BP; 3) PBP decoder for a time-varying crossover probability.

periodically every 20 BP iterations until the BP decoder successfully decodes the codeword or reaches its maximum number of 150 iterations. In Fig. 4, the crossover probability is considered as constant and the initial crossover probabilities are for $0.1$ above the true crossover probabilities for all schemes. We can observe a large performance gap between the BP decoder and LP-based BP decoder in the presence of a crossover probability mismatch, while the performance of the proposed LP based decoder is similar to that of the PBP decoder. In Fig. 5, decoding performance with time-varying crossover probability is investigated, where a sinusoidally changing crossover probability is described as the aforementioned model in Fig. 3. We can see that the proposed LP based decoder obtained a much better performance than the BP decoder with the mean of time-varying crossover probability, and there is only a small performance gap between the LP based decoder and PBP decoder. Note that the additional computational overhead introduced by the proposed LP estimator is very small. In addition, compared to the PBP decoder, which mainly uses Monte Carlo techniques, the proposed LP based decoder has ultra-low computational complexity.

## 6. CONCLUSION

For a DSC problem with unknown correlation statistics, we proposed a factor graph associated with an LP estimator for SW decoding and stationary/time-varying correlation estimation. The experimental results show that the proposed scheme is not sensitive to the initial knowledge of the source correlation and can precisely track both the stationary and the time-varying source correlation. Moreover, the proposed scheme yields a better decoding performance than the standard BP algorithm and shows a comparable performance to the significantly more complex PBP decoder. Finally, the proposed LP estimator also shows a very fast convergence speed with only few iterations and ultra-low complexity compared to the PBP decoder.

## 7. 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.D. 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, pp. 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.P. Reilly, and S. Shirani, "Distributed parameter estimation with side information: A factor graph approach," in *IEEE Int. Symp. Inform. 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 Trans. Commun.*, vol. 59, no. 9, pp. 2337 – 2342, 2011.

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

[9] C.M. Bishop, *Pattern recognition and machine learning*, vol. 4, Springer New York:, 2006.

[10] A. Smola, SVN Vishwanathan, and E. Eskin, "Laplace propagation," *Advances in neural information processing systems*, vol. 16, 2003.

[11] S. Cheng, S. Wang, and L. Cui, "Adaptive Slepian-Wolf decoding using particle filtering based belief propagation," in *Annual Allerton Conf.*, 2009, pp. 607–612.