Sliding-Window Belief Propagation with Unequal Window Size for Nonstationary Heterogeneous Source

Jiao Fan, Bowei Shan, and Yong Fang(✉)

School of Information Engineering, Chang'an University, Xi'an, China
{2018124056,bwshan,fy}@chd.edu.cn

Abstract. This paper presents a Sliding-Window Belief Propagation with Unequal Window Size (SWBP-UWS) algorithm to deal with the nonstationary heterogeneous source. In this algorithm, the entire source is divided into several sections according to its variation and each optimum window size is individually determined by each section. The experimental results show this algorithm outperforms the SWBP algorithm.

Keywords: SWBP algorithm · Belief propagation · LDPC code

1 Introduction

1.1 Background

Reliable communication paves the way for the development of modern IoT as a service and 5G technology. As an important error-correcting code, Low-Density Parity-Check (LDPC) codes [1,2] could protect source codes from noisy channel. Hence, LDPC codes have played a key role in various networking and communication system.

Decoding LDPC codes is a very important problem. In 2012, the Sliding-Window Belief Propagation (SWBP) technique was proposed by Fang [3] to decode LDPC codes. Thereafter, a lot of experiments [4,5] had showed that SWBP possesses many advantages compared with standard Belief Propagation (BP) algorithm. It can exactly trace time-varying channel state, demonstrate near-limit performance, is easy to be implemented, is robust to initial parameters, and is convenient to be parallelized. Recently, Shan [6] has used Graphics Processing Units (GPUs) to accelerate a parallel version of SWBP algorithm and achieved a high speed-up ratio.

All above works are based on an assumption that source are transmitted on the channel with smoothly time-varying, e.g. sinusoidally-varying, homogeneous state. Here, "homogeneous state" means the frequency of the time-varying functions keeps constant. While in the real communication system, the source and channel state usually varies arbitrarily. To better model the source, different frequency of the time-varying functions should be considered, which we name as

B. Li et al. (Eds.): IoTaaS 2019, LNICST 316, pp. 375–381, 2020.
https://doi.org/10.1007/978-3-030-44751-9_31

heterogeneous source. In Fang's SWBP algorithm, the *uniform sized* windows are imposed to calculate the initial local bias probability of each variable nodes for belief propagation. This set up is effective for the homogeneous source. Whether it is suitable for the heterogeneous source still need to be investigated.

1.2 Contribution

The main contribution of this paper includes: (a) A new Sliding-Window Belief Propagation with Unequal Window Size (SWBP-UWS) algorithm is proposed to deal with binary LDPC codes; (b) the size of sliding window is directly determined by the channel parameters in frequency domain; (c) smoothly sinusoidally-varying channel states with two different frequency are studied. The numerical simulated experiments demonstrate that our proposed algorithm performs better than original SWBP.

The rest of this paper is organized as follows: Sect. 2 briefly reviews the SWBP algorithm. Sect. 3 introduces our proposed model. In Sect. 4, experiment results are reported and discussed. Section 5 concludes this paper and gives the future possible development.

2 Preliminary

2.1 Notations

The LDPC code is specified by an $m \times n$ parity check matrix $\mathbf{H}$. We give the following notations to describe SWBP model.

- $\mathbf{x} = (x_1, \ldots, x_n)^T$ is the binary source, and $\tilde{\mathbf{x}}$ is the estimate of $\mathbf{x}$;
- $\mathbf{y} = (y_1, \ldots, y_n)^T$ is the Side Information (SI) available at receiver side;
- $\mathbf{s} = (s_1, \ldots, s_m)^T = \mathbf{H}\mathbf{x}$ is the syndrome;
- $\mathbf{p} = (p_1, \ldots, p_n)$, where p_i is the local bias probability, $p \triangleq \frac{1}{n}\sum_{i=1}^{n} p_i$ is the global bias probability, and $\tilde{\mathbf{p}}$ is the estimate of $\mathbf{p}$;
- α_{ij} is the belief propagated from variable node x_i to check node s_j;
- β_{ji} is the belief propagated from check node s_j to variable node x_i;
- $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m)^T$ is the overall belief of variable nodes;
- $\mathcal{M}_i$ is the set of indices of check nodes connected to variable node x_i
- $\mathcal{N}_j$ is the set of indices of variable nodes connected to check node s_j

The SWBP algorithm includes three phases: the standard BP, calculating window size and refining local bias probability.

2.2 Steps of Standard BP

(1) Initialization:

$$\sigma_i^{(0)} = (1 - 2y_i) \log \frac{(1 - p_i)}{p_i}, \quad \text{and} \quad \beta_{ji}^{(0)} = 0 \tag{1}$$

(2) Variable nodes to check nodes BP:

$$\frac{1-\alpha_{ij}}{\alpha_{ij}} = \frac{1-\sigma_i}{\sigma_i} \Big/ \frac{1-\beta_{ji}}{\beta_{ji}}. \tag{2}$$

(3) Check nodes to variable nodes BP:

$$1 - 2\beta_{ji} = (1 - 2s_j) \prod_{i' \in \mathcal{M}_j \backslash i} (1 - 2\alpha_{i'j}) \tag{3}$$

(4) Computing overall brief of variable nodes:

$$\frac{1-\sigma_i}{\sigma_i} = \frac{1-\tilde{p}_i}{\tilde{p}_i} \Big/ \prod_{j' \in \mathcal{N}_i} \frac{1-\beta_{j'i}}{\beta_{j'i}}. \tag{4}$$

(5) Hard decision:

$$\tilde{x} = \begin{cases} 0, & \sigma_i \leq 0.5 \\ 1, & \sigma_i > 0.5 \end{cases} . \tag{5}$$

If $\mathbf{H}\tilde{\mathbf{x}} = \mathbf{s}$, the decoding process is successfully terminated; otherwise, more iteration is required.

2.3 Calculating Window Size

At the beginning of each BP iteration, all variable nodes need be seeded with local bias probability $\mathbf{p}$. Hence, estimating an appropriate $\tilde{\mathbf{p}}$ is a key issue. For a nonstationary smooth source, we can treat any segment of this source in a small window as approximately stationary. Therefore, $\tilde{\mathbf{p}}$ can be obtained by taking variable node x_i as center, and averaging it's neighbor's overall belief in a window with size h.

In fact, a suitable h is a tradeoff between two factors: on one hand, h should be large enough that neighbor nodes' information can be taken into account as much as possible; on the other hand, h should be rather small to ensure that source keeps stationary in this window. In [3], an adaptively searching algorithm was proposed to calculating window size h. In this algorithm, $\tilde{\mathbf{p}}$ is first computed by (6)

$$\tilde{p}_i = \frac{-\sigma_i + \sum_{i'=\max(1,i-u)}^{\min(i+u,n)} \sigma_{i'}}{\min(i+u,n) - \max(1,i-u)}, \tag{6}$$

where $u = \lfloor h/2 \rfloor$ is the half window size. Then, the mean squared error (MSE) between $\tilde{\mathbf{p}}$ and $\boldsymbol{\sigma}$ is computed by (7)

$$\tau = \frac{1}{n} \sum_{i=1}^{n} (\tilde{p}_i - \sigma_i)^2. \tag{7}$$

At last, an appropriate h is obtained when MSE reaches the smallest value, which is named h^*. In [3], all possible $h \in \{1, 2, \ldots, n\}$ are tested to find out the h^*.

2.4 Refining Local Bias Probability

Once h^* is determined, local bias probability can be refined by (6), which can be straightforwardly deduced to follows.

$$\tilde{p}_i = \begin{cases} \tilde{p}_{i-1} + \dfrac{\sigma_{i-1} - \sigma_i + \sigma_{i+u} - \tilde{p}_{i-1}}{i+u-1}, & 2 \leq i \leq (1+u) \\ \tilde{p}_{i-1} + \dfrac{\sigma_{i-1} - \sigma_i + \sigma_{i+u} - \sigma_{i-u-1}}{h-1}, & (2+u) \leq i \leq (n-u) \\ \tilde{p}_{i-1} + \dfrac{\sigma_{i-1} - \sigma_i - \sigma_{i-u-1} + \tilde{p}_{i-1}}{n-i+u}, & (n-u+1) \leq i \leq n \end{cases} . \tag{8}$$

3 Model Description

SWBP in Sect. 2 can only tackle the homogeneous source or channel state. In this section, we present the SWBP-UWS algorithm to deal with heterogeneous source. The steps of standard BP are identical with that in Sect. 2. The method of calculating window size in Sect. 2 is very time consuming. In [5], the window size was directly obtained by the frequencial characteristics of variable nodes' global bias probability. We borrow this novel idea into our new algorithm.

3.1 Calculating Different Window Size

Assuming the source is consist of l sections, i.e. $\{x_1, x_2, \ldots, x_n\} = \{x_1, \ldots, x_{n_1}\} \cup \{x_{n_1+1}, \ldots, x_{n_2}\} \cup \ldots \{x_{n_{l-1}+1}, \ldots, x_{n_l}\}$, where $n = n_l$. The frequency of time-varying function in one section is different from that in another section, but keeps constant within itself. In SWBP-UWS algorithm, the best window size for each section is independently computed by the overall beliefs within this section. Let $|\cdot|$ be the cardinality of the set. For the ith section, the algorithm first calculates the Fast Fourier Transform (FFT) of $\mathbf{r}^i$ as $f^i(\theta)$), where $\theta \in [n_i + 1 : n_{i+1}]$, and $|\theta| = W_i$. Then let $D_i = \sum_{\theta=1}^{W_i} |f^i(\theta)|$. Then the best window size for ith section is calculated by:

$$h_i^* \approx \frac{W_i}{D_i} \sum_{\theta=1}^{W_i} |f^i(\theta)|/\theta \tag{9}$$

3.2 SWBP-UWS Algorithm

Once all best window sizes $h_i^*, i \in [1, ..., l]$ are computed, the local bias probability for each section can be computed by Eq. (8). Then we combine them to an optimum local bias probability $\tilde{\mathbf{p}}$, which will be the seeds for next BP iteration. The SWBP-UWS algorithm is summarized as follow.

Algorithm 1. SWBP-UWS Algorithm.

Require: Overall beliefs, $\mathbf{r}$; Side information, $\mathbf{y}$; Syndrome, $\mathbf{s}$;
Ensure: Estimated source, $\tilde{x}$;

```
Initialization: b_i^(0) = 0, q_ij^(0) = pmf_ij
for l = 1 : MAX_ITERATION do
    Steps of Standard BP [3]
    Computing overall beliefs r = (r_1, ..., r_n)^T for varialbe nodes:
    Dividing overall beliefs r into l sections
    Computing each best window size h_i^* by equation (9)
    Computing each optimum local bias probability by equation (8)
    Combining local bias probabilities in each section to optimum local bias probability p̃
    Hard decision:
    x̃_i = { 0, if r_i ≤ 0.5 ; 1, if r_i > 0.5 }
    if Hx̃ = s then
        quit loop
    end if
end for
```

Hard decision: $\widetilde{x_i} = \begin{cases} 0, & \text{if } r_i \leq 0.5 \\ 1, & \text{if } r_i > 0.5 \end{cases}$

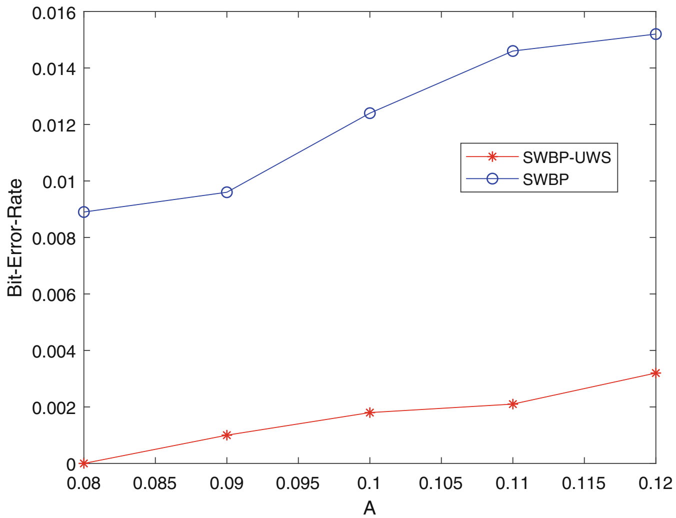

Fig. 1. Bit-error-rate under two algorithms.

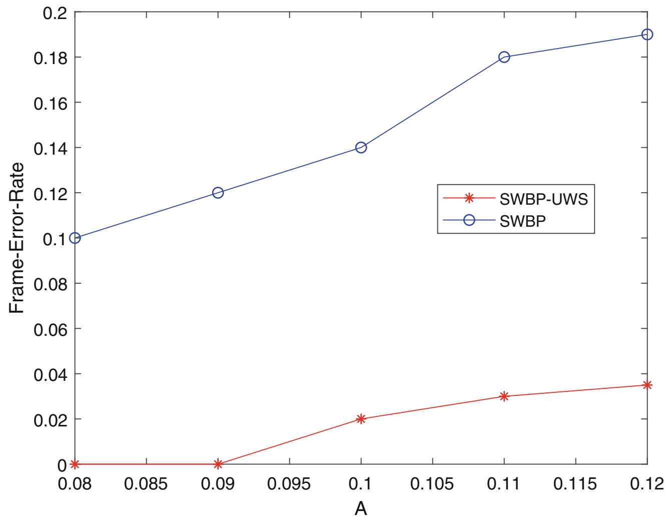

Fig. 2. Frame-error-rate under two algorithms.

4 Experimental Results

To evaluate the performance of SWBP-UWS for nonstationary heterogeneous source, a regular length n-1024 LDPC code is constructed. The local bias probability varies according to $p_i = p_{i1} + p_{i2}$, where $p_{i1} = A(1 + \sin(2\pi f_1 i_1/n))$, $i_1 \in [1, ..., 512]$ and $p_{i2} = A(1 + \sin(2\pi f_2 i_2/n))$, $i_2 \in [513, ..., 1024]$. To satisfy the heterogeneous source, we set $f1_1 : f_2 = 1 : 3$. Let five values of A be tested, *i.e.*, $A \in \{0.08, 0.09, 0.10, 0.11, 0.12\}$.

The experimental results are illustrated in Figs. 1 and 2, from which we find that under different A SWBP-UWS outperforms SWBP algorithm in both BER and FER.

5 Conclusion

To handle the nonstationary heterogeneous source, we present a Sliding-Window Belief Propagation with Unequal Window Size (SWBP-UWS) algorithm. In this algorithm, the entire source is divided into several sections according to its variation and each optimum window size is individually determined by each section. We perform numerical experiments to evaluate our algorithm. Comparing SWBP, SWBP-UWS demonstrates better performance in both BER and FER.

References

1. Gallager, R.G.: Low-density parity-check codes. IRE Trans. Inf. Theory **8**(1), 21–28 (1962)
2. Mackay, D.J.C., Neal, R.M.: Near Sshannon limit performance of low density parity check codes. Electron. Lett. **32**(6), 457–458 (1997)
3. Fang, Y.: Ldpc-based lossless compression of nonstationary binary sources using sliding-window belief propagation. IEEE Trans. Commun. **60**(11), 3161–3166 (2012)
4. Fang, Y.: Asymmetric Slepian-Wolf coding of nonstationarily-correlated M-ary sources with sliding-window belief propagation. IEEE Trans. Commun. **61**(12), 5114–5124 (2013). https://doi.org/10.1109/TCOMM.2013.111313.130230
5. Fang, Y., Yang, Y., Shan, B., Stankovic, V.: Joint source-channel estimation via sliding-window belief propagation. IEEE Trans. Commun. (2019, submitted)
6. Shan, B., Fang, Y.: GPU Accelerated parallel algorithm of sliding-window belief propagation for LDPC codes. Int. J, Parallel Program. (2019)