Unequally Weighted Sliding-Window Belief Propagation for Binary LDPC Codes

Zhaotun Feng, Bowei Shan, and Yong Fang(✉)

School of Information Engineering, Chang'an University,
Xi'an, People's Republic of China
{2018124076,bwshan,fy}@chd.edu.cn

Abstract. In this paper, an Unequally Weighted Sliding-Window Belief Propagation (UW-SWBP) algorithm was proposed to decode the binary LDPC code. We model the important of overall beliefs of variable nodes in a sliding window as Gaussian distribution, which means central nodes play a more importance role than the nodes on both sides. The UW-SWBP demonstrates better performance than SWBP algorithm in both BER and FER metrics.

Keywords: UW-SWBP · LDPC · Overall belief

1 Introduction

Thanks to extremely fast data speed, very low latency and ubiquitous coverage, the Fifth Generation (5G) technology promises a fundamental ecosystem for the Internet of Things as a Services (IoTaaS). In 2018, 5G New Radio (NR) has used Low-Density Parity Check (LDPC) Codes as the code for forward error correction [1]. LDPC codes was originally invented by Gallager [2] in 1962. Due to lack of high computing ability at that time, it was not recognized by information community for nearly thirty years. In 1990s, MacKay at Cambridge [3] and Spielman at MIT [4] independently rediscovered the LDPC codes and showed its near-Shannon-limit performance.

In today's communication society, LDPC codes play a key role in many different areas. Thereafter, how to design a LDPC decoder with best performance is an important issue. In 2012, a Sliding-Window Belief Propagation (SWBP) algorithm [5] for decoding binary LDPC codes was introduced by Fang. Many experiments [6,7] shows that SWBP can outperform standard belief propagation (BP) algorithm. As the state-of-the-art LDPC decoder, SWBP can be easily accelerated by GPU as well [8].

The idea of SWBP is based on belief propagation between variable nodes and check nodes of LDPC, while SWBP initialize the variable nodes with the

Supported by organization x.

B. Li et al. (Eds.): IoTaaS 2019, LNICST 316, pp. 582–588, 2020.
https://doi.org/10.1007/978-3-030-44751-9_49

optimum seeds which are computed by averaging the overall beliefs of the variable nodes in a well-chosen window size. Experiments results demonstrated that, as long as the window size is selected properly, this technique is highly effective [5–7]. With the advance of this research, we discover that some improvement can be made in the basic SWBP algorithm, which regards all the variable nodes as same weights. In fact, different variable nodes in the window should have different importance. The closer to the central nodes, the more important the nodes should be. In this paper, we present an Unequally Weighted Sliding-Window Belief Propagation (UW-SWBP) algorithm. In this algorithm, the optimum seeds are obtained by weighted averaging overall beliefs, and the weights decreases from the center node to the nodes on both sides. The experiment results show that UW-SWBP outperforms standard SWBP for decoding binary LDPC codes.

This paper is organized as follow. Section 2 reviews the SWBP algorithm. Our proposed UW-SWBP is introduced in Sect. 3. Section 4 presentes the experiment results and Sect. 5 concludes this work.

2 Review on Related Works

2.1 Notations

Let $\mathbf{x} = (x_1, \ldots, x_n)^T$ be the binary source codeword and $\mathbf{y} = (y_1, \ldots, y_n)^T$ be the binary received codeword, where n is the length of the LDPC code. Let $\mathbf{H}$ be an $m \times n$ parity check matrix, and $\mathbf{s} = (s_1, \ldots, s_n)^T = \mathbf{Hx}$ be the syndrome. In this paper, we only consider Binary Symmetric Channel (BSC) with cross probability $\mathbf{p} = (p_1, \ldots, p_n)$, where $p_i = \Pr(x_i \neq y_i)$.

2.2 SWBP Algorithm

SWBP algorithm is consisted of two steps: standard BP algorithm and computing the optimum seeds. Standard BP algorithm has been depicted in many articles. Reader nay refer to [5] for details.

The optimum seeds $\tilde{\mathbf{p}} = (\tilde{p}_1, \ldots, \tilde{p}_n)^T$of variable nodes can be computed by averaging the overall beliefs of neighboring variable nodes in a size-l (an odd) window around x_i.

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

where $\mathbf{r} = (r_1, \ldots, r_n)^T$ is the overall belief of variable node i, and $u = \lfloor l/2 \rfloor$. Ref. [5] also presented an stepwise form of Eq. (1):

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

How to determine a proper window size l^* is the key task in SWBP. In [5,6], the mean squared error (MSE) τ^2 between r and $\tilde{p}$ is calculated.

$$\tau^2 = \frac{1}{n}\sum_{i=1}^{n}(r_i - \tilde{p}_i)^2. \tag{3}$$

Then we search all l one by one from 1 to n and only that gives the smallest τ^2 is selected as l^*.

The complexity of above algorithm is $\mathcal{O}(n^2)$. To reduce it, a fast window size algorithm is presented in [7]. This search-free algorithm first calculates the Fast Fourier Transform (FFT) of $\mathbf{r}$ as $f(\theta)$, where $\theta \in [1 : n]$. Then let

$$A = \sum_{\theta=1}^{n} |f(\theta)|, \tag{4}$$

where $|f(\theta)|$ is the modulus of $f(\theta)$. Then the proper window size

$$l^* \approx \frac{n}{A}\sum_{\theta=1}^{n} |f(\theta)|/\theta, \tag{5}$$

Obviously, the complexity of this new algorithm is reduced to $\mathcal{O}(n \log_2 n)$.

3 UW-SWBP Algorithm

Our proposed UW-SWBP is based on SWBP. The step of Standard BP of these two are identical, while the step of computing the optimum seeds is improved in UW-SWBP to achieve better performance. In Eq. (1), every overall beliefs of variable nodes r_i in window have the same importance account for $\tilde{p}_i$. In fact, when we want to calculate the optimum seeds of a node, a natural idea is to highlight the characteristics of the node, so we believe that nodes in different locations should play different degrees of importance.In other words, the central node should play a more important role than the nodes on both sides of the window. As a result, we modify Eq. (1) to a weighted form.

$$\tilde{p}_i = \frac{-r_i * w_i + \sum_{i'=1}^{n}(r_{i'} * w_{i'})}{\sum_{i'=1}^{n} w_{i'}}, \tag{6}$$

where the weight of variable nodes $w(x)$ follows Gaussian distribution

$$w(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{|x-\mu|}{2\sigma^2}}. \tag{7}$$

We illustrate equal and unequal weights distribution in Fig. 1. Obviously unequal weights distribution fulfills our expectations of the importance of different nodes.

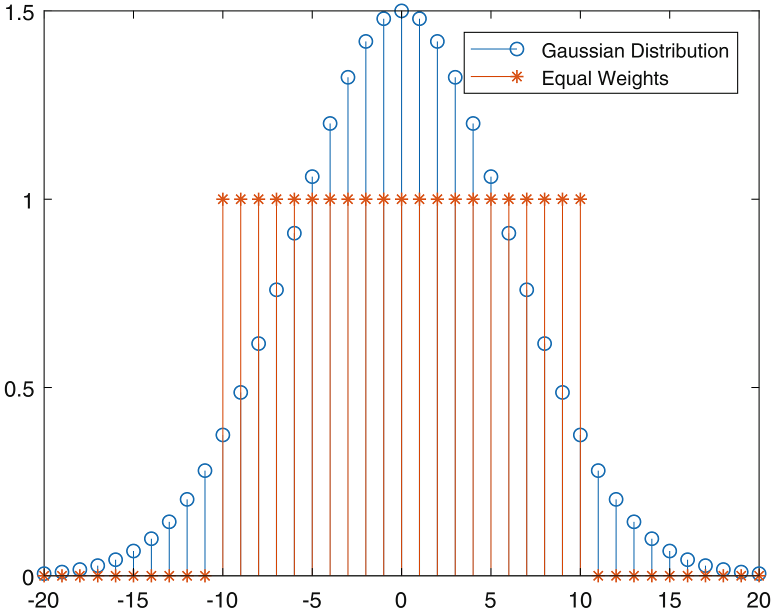

Fig. 1. Different type of weight distribution.

We can find that parameter σ plays an key role in computing the optimum seeds. Because σ determines the size of the window and the importance of each node. The main task of UW-SWBP is to estimate an optimum σ^*. Same as SWBP algorithm, we first calculate the MSE τ^2 between r_i and $\tilde{p}_i$ by Eq. (3). Then we calculate the following problem:

$$\sigma^* = \arg\min_{\sigma} \tau^2, \tag{8}$$

which can be solved by searching all σ from 1 to n and only that gives the smallest τ^2 is choosed as σ^*. We show the UW-SWBP algorithm as follow.

Algorithm 1. UW-SWBP Algorithm.

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

Initialization: $b_i^{(0)} = 0$, $q_{ij}^{(0)} = pmf_{ij}$
for $l = 1 : \text{MAX\_ITERATION}$ **do**
    Standard BP algorithm [5]
    Computing overall beliefs $\mathbf{r} = (r_1, \ldots, r_n)^T$ for variable nodes:
    Computing optimum parameter σ^* by equation (6)
    Refining local bias probability $\tilde{\mathbf{p}} = (\tilde{p}_1, \ldots, \tilde{p}_n)^T$ by equation (4) and equation (5)
    Hard decision:
    $\widetilde{x_i} = \begin{cases} 0, & \text{if} \quad r_i \leq 0.5 \\ 1, & \text{if} \quad r_i > 0.5 \end{cases}$
    if $\mathbf{H}\widetilde{\mathbf{x}} = \mathbf{s}$ **then**
        quit loop
    end if
end for

4 Experimental Results

In this section, we will use numerical simulation to carry out experiments. The aim of this paper is to evaluate the performance of UW-SWBP algorithm. Our testing platform is configured with Intel(R) Core(TM) E7500 CPU 2.93GHz(2 Cores) main frequency, 4GB memory, and 64-bit Windows-10 operation system. All algorithms are developed in MATLAB 2017b environment.

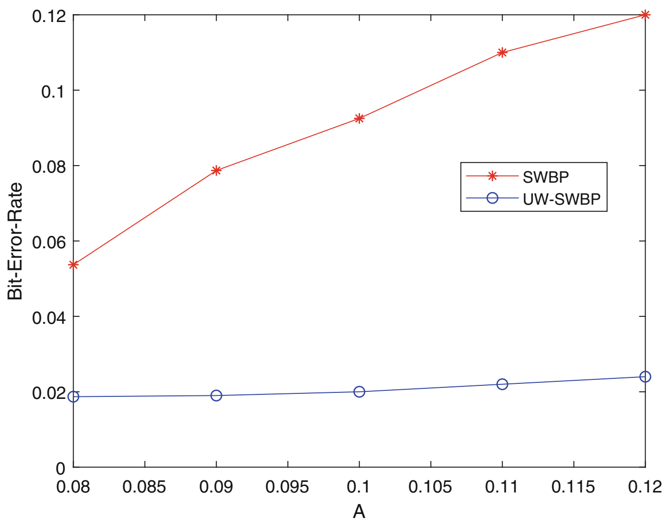

Fig. 2. Bit-Error-Rate under two algorithms

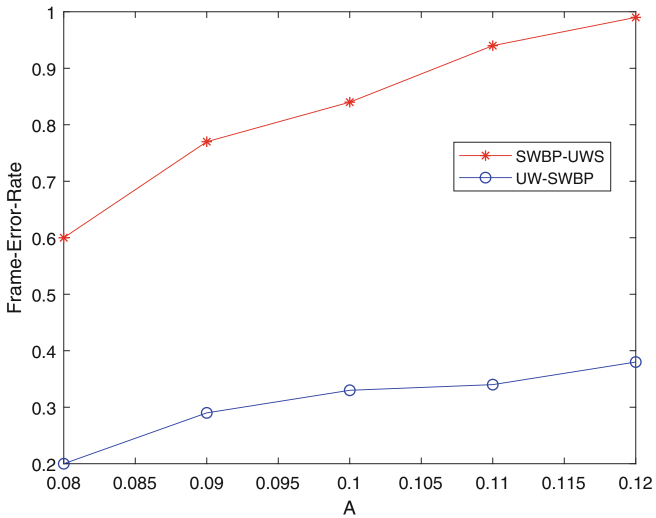

Fig. 3. Frame-Error-Rate under two algorithms

We construct a LDPC code with codeword length $n = 1024$ and information bit number $k = 512$. The local bias probability varies according to $p_i = A(1 + \sin(2\pi i/n))$. We test five value of A, *i.e.*, $A \in \{0.08, 0.09, 1.00, 1.05, 1.10\}$ in our experiment. The Bit-Error-Rate (BER) and Frame-Error-Rate (FER) are used as the metrics to compare these two algorithms. The experimental results are illustrated in Figs. 2 and 3.We can see that the BER and FER of UW-SWBP are much lower than SWBP when the initial value A=0.08. Moreover, with the increase of A value, the BER and FER of SWBP show unstable and drastic changes, and the increase trend of UW-SWBP has been very flat. We find that under different A, UW-SWBP outperforms SWBP in both BER and FER.

5 Conclusion

In this paper, we propose a unequally weighted Sliding-Window Belief Propagation (UW-SWBP) algorithm to decode binary LDCP codes. We assume that the importance of the overall belief of a variable node is subject to a Gaussian distribution. The UW-SWBP algorithm can improve the local deviation probability with higher precision. Experiment results show that UW-SWBP has better performance that SWBP algorithm.

References

1. Richardson, T., Kudekar, S.: Design of low-density parity check codes for 5G new radio. IEEE Commun. Mag. **56**(3), 28–34 (2018)

2. Gallager, R.G.: Low-density parity-check codes. IRE Trans. Inf. Theory **8**(1), 21–28 (1962)
3. Mackay, D.J.C., Neal, R.M.: Near Shannon limit performance of low density parity check codes. Electron. Lett. **32**(6), 457–458 (1997)
4. Sipser, M., Spielman, D.A.: Expander codes. IEEE Trans. Inform. Theory **42**(11), 1710–1722 (1996)
5. Fang, Y.: Ldpc-based lossless compression of nonstationary binary sources using sliding-window belief propagation. IEEE Trans. Commun. **60**(11), 3161–3166 (2012)
6. 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)
7. Fang, Y., Yang, Y., Shan, B., Stankovic, V.: Joint source-channel estimation via sliding-window belief propagation. IEEE Trans. Commun (2019). (submitted)
8. Shan, B., Fang, Y.: GPU Accelerated parallel algorithm of sliding-window belief propagation for LDPC codes. Int. J. Parallel Prog (2019). (Online First). https://doi.org/10.1007/s10766-019-00632-3