MIN-SUM 복호화 알고리즘을 이용한 LDPC 오류정정부호의 성능분석

정회원 허 준*

Convergence of Min-Sum Decoding of LDPC codes under a Gaussian Approximation

Jun Heo* ***Regular Member***

요 약

최근에 소개된 density evolution 기법은 sum-product 알고리즘에서 LDPC 부호가 갖는 성능의 한계를 분석하였다[1]. 또한, Iterative decoding 알고리즘에서 전달되는 정보가 Gaussian 확률분포를 갖는 점을 이용하여 기존의 density evolution 기법을 단순화 시킨 연구결과가 소개되었다[2]. 한편, LDPC 부호의 한계 성능을 sum-product가 아닌 min-sum 알고리즘에서 분석한 결과가 최근에 발표되었다[3]. 본 논문에서는 이러한 일련의 연구 결과를 바탕으로 min-sum 알고리즘을 이용하면서 Gaussian 확률 분포 특성을 이용한 density evolution 기법을 소개한다. 제안된 density evolution 기법은 기존의 방법보다 적은 계산으로 정확한 threshold를 구할 수 있으며, 그 결과가 numerical simulation 결과와 잘 일치함을 나타내었다.

ABSTRACT

Density evolution was developed as a method for computing the capacity of low-density parity-check(LDPC) codes under the sum-product algorithm [1]. Based on the assumption that the passed messages on the belief propagation model can be approximated well by Gaussian random variables, a modified and simplified version of density evolution technique was introduced in [2]. Recently, the min-sum algorithm was applied to the density evolution of LDPC codes as an alternative decoding algorithm in [3]. Next question is how the min-sum algorithm is combined with a Gaussian approximation. In this paper, the capacity of various rate LDPC codes is obtained using the min-sum algorithm combined with the Gaussian approximation, which gives a simplest way of LDPC code analysis. Unlike the sum-product algorithm, the symmetry condition [4] is not maintained in the min-sum algorithm. Therefore, the variance as well as the mean of Gaussian distribution are recursively computed in this analysis. It is also shown that the min-sum threshold under a gaussian approximation is well matched to the simulation results.

* 건국대학교 전자공학부 (junheo@konkuk.ac.kr)
논문번호 : 020499-1125, 접수일자 : 2002년 11월 25일

I . Introduction

A density evolution technique was recently developed for the sum-product decoding algorithm of LDPC codes and it explains the capacity and convergence issues of the iterative decoding. In [1], the density evolution was used to track the density of extrinsic message between the variable nodes and check nodes of LDPC code on various channel conditions. An simplified version of the density evolution algorithm was introduced under a Gaussian approximation in [2]. The Gaussian approximation was based on the well known fact that the extrinsic information can be well represented as a Gaussian random variable as the number of iteration increases. The message passing expression for sum-product algorithm was developed in [5] and its approximation was also introduced. This approximation was used for the density evolution based on the min-sum algorithm in [3]. The probability density function for the min-sum approximation was derived and the computed min-sum capacities(thresholds) were compared to those of the sum-product algorithm.

As pointed in [3], next question is to combine the Gaussian approximation with the min-sum algorithm. Empirically we know that performance of sum-product algorithm is better than that of min-sum algorithm. It is interesting to know if this characteristic is maintained under the Gaussian approximation for the passed message. In this paper, the density evolution of min-sum algorithm under the Gaussian approximation is presented and the capacities of both decoding algorithms are compared. If we assume the symmetry condition $f(x)=f(-x)e^{x}$ for a Gaussian probability density function $f(x)$ as in [4],[2] under the sum-product algorithm, the variance of Gaussian density can be represented by two times of the mean. However, the symmetry condition does not hold in the min-sum algorithm. Therefore, the variance as well as the mean should be computed during the iterative density evolution process.

II. Sum-Product algorithm under a Gaussian approximation

A (d_v, d_c) LDPC code can be represented by a bipartite graph which consists of variable nodes with d_v edges and check nodes with d_c edges. A (3,6) LDPC code bipartite graph is shown in Fig. 1. The exchanged messages on the graph are the Log-Likelihood Ratios (LLRs).

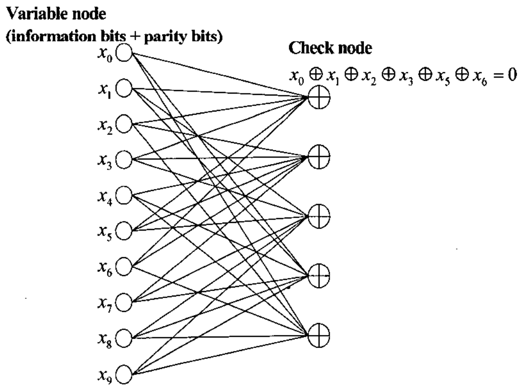


Fig. 1. Bipartite graph of (3,6) regular LDPC code

The output messages from a variable node is represented by v and the messages from a check node is represented by u. Based on the sum-product algorithm, the messages from the variable and check nodes are represented by

$$v=\lambda_c+\sum_{i=1}^{d_v-1} u_i \quad (1)$$

$$\tanh(\frac{u}{2})=\tanh(\frac{v_1}{2})\ldots\tanh(\frac{v_{d_c}-1}{2}) \quad (2)$$

where λ_c represents a received message from channel and the message from the node receiving the output message is excluded. Because of the independent and identically distributed assumption [1],[2] for messages, we omit any time index in the message representation. With the gaussian approximation [2], it is only needed to compute the mean and variance of exchanged messages recursively. In addition, based on the symmetry condition $f(x)=f(-x)e^{x}$ [1] for the density $f(x)$ of an LLR message, the variance can be represented as two times the mean. As a result, we need only to track the mean of a gaussian density.

By taking expectation on both sides of eq. (1), the message from a variable node is

$$\overline{v}=\overline{\lambda}_c+(d_v-1)\overline{u}$$

where $\overline{x}$ denotes the mean of random variable x. Similarly, we can take the expectation on both sides of eq. (2). To simplify the notation, we define $\phi(x)$ as:

$$\phi(x)=\int_{-\infty}^{\infty}\tanh\left(\frac{y}{2}\right)\frac{1}{\sqrt{4\overline{y}}}e^{-\frac{(y-\overline{y})^2}{4\overline{y}}}dy = E\left[\tanh\left(\frac{y}{2}\right)\right]$$

then, the expectation on the message from a check node can be represented as:

$$\phi(\overline{u})=\phi(\overline{v}_1)\cdots\phi(\overline{v}_{d_c-1})$$

Then we can obtain $\overline{u}$ by taking inverse function $\phi^{-1}(\cdot)$. To calculate the $\phi(\cdot)$ and $\phi^{-1}(\cdot)$ effectively, we use the following approximation which was used in [2].

$$\phi(x)\cong\begin{cases}1-\frac{1}{2}\sqrt{\frac{\pi}{x}}e^{-\frac{x}{4}}\left(2-\frac{1}{7x}-\frac{3}{x}\right): & x>10\\ 1-e^{\alpha x^{\gamma}-\beta} & : x\leq 10\end{cases}$$

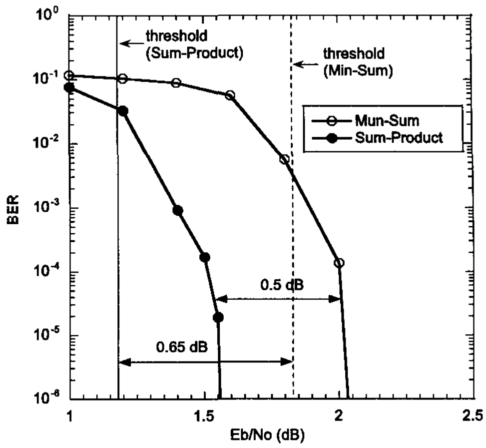


Fig. 2. Simulation results and thresholds of regular (3,6) LDPC code for block size $N=10000$ with Min-Sum and Sum-Product algorithm

d_v	d_c	rate	$\sigma^{GA}_{\Sigma-\Pi uct}$ ($\frac{E_b}{N_o}dB$)	$\sigma^{GA}_{\min-\Sigma}$ ($\frac{E_b}{N_o}dB$)
3	6	0.5	0.8740 (1.17)	0.8100 (1.83)
4	8	0.5	0.8318 (1.06)	0.7405 (2.61)
5	10	0.5	0.7907 (2.04)	0.6942 (3.17)
3	5	0.4	0.9999 (0.97)	0.9036 (1.85)
4	6	1/3	1.0036 (1.73)	0.8631 (3.04)
3	4	0.25	1.2517 (1.06)	1.0827 (2.32)
4	10	0.6	0.7437 (1.78)	0.6767 (2.60)
3	9	2/3	0.7047 (1.79)	0.6800 (2.10)
5	12	0.75	0.6294 (2.26)	0.6165 (2.44)

Table 1. The Threshold on MIN-SUM and SUM-PRODUCT algorithm for various pairs (d_v, d_c)

where $\alpha=-0.4527$, $\beta=0.0218$, and $\gamma=0.86$. By initially setting $\overline{u}$ to zero, we can recursively update the mean of each message until it converges to a finite value or goes to infinity. When it goes to infinity, it means that the density tends to a point mass at infinity or equivalently, the probability of error tends to zero. The threshold is calculated as the maximum noise level (i.e., minimum SNR) from the channel such that the probability of error tends to zero [1],[2].

The thresholds, which are calculated by the above method, are shown in Table 1 and the simulation results for the case of $(d_v, d_c) = (3, 6)$ is presented in Fig. 2. The computed threshold is well matched to the waterfall region of simulation curve.

Ⅲ. Min-Sum algorithm under a Gaussian approximation

The density evolution techniques in the literature can be classified based on the way of tracking the density function of extrinsic information. One approach is a quantization of a density function and apply the density evolution equations approximately [1]. The other approach is to quantize messages and track probability mass function [6]. To avoid these complex computations, a Gaussian approximation approach was introduced in [2], which reduces an infinite dimensional problem into a one dimensional problem. With this Gaussian approximation, it is only necessary to track the mean and variance of the exchanged extrinsic information.

Combining this Gaussian approximation with the recent work in [3], a simpler min-sum density evolution technique is presented. For regular (d_v, d_c) LDPC codes the min-sum message passing algorithm at variable and check nodes is represented by

$$v = \lambda_c + \sum_{i=1}^{d_v - 1} u_i \quad (3)$$

$$u = sign(v_1 \cdots v_{d_c - 1}) \min[|v_1|, \cdots, |v_{d_c - 1}|] \quad (4)$$

where v is the Log Likelihood Ratio (LLR) output from variable node and u is the LLR output from check node and λ_c represents the received message from the channel.

This min-sum message passing algorithm at a check node was shown in [5][7][3]. According to this expression, the output message from a check node can be represented by the sign of the product of the incoming messages and the minimum absolute value among the incoming messages, where the message from the node receiving the output message is excluded.

On the min-sum density evolution with a Gaussian assumption, our goal is to track the mean $\overline{v}, \overline{u}$ and variance σ_v^2, σ_u^2 of the LLRs during the iterative decoding. With the independent and identically distribution assumption for the exchanged messages, $\overline{v}$ and σ_v^2 can be obtained from (3) as:

$$\overline{v} = \overline{\lambda_c} + (d_v - 1)\overline{u} \quad (5)$$

$$\sigma_v^2 = \sigma_{\lambda_c}^2 + (d_v - 1)\sigma_u^2 \quad (6)$$

The calculation of $\overline{u}$ and σ_u^2 is more involved. When $d_c = 3$, the probability density function (pdf) of the output message from a check node based on (4) was derived in [3] as:

$$f_u(x) = f_{v_1}(x)(1 - F_{v_2}(x)) + f_{v_2}(x)(1 - F_{v_1}(x)) + f_{v_1}(-x)F_{v_2}(-x) + f_{v_2}(-x)F_{v_1}(-x), \quad x > 0 \quad (7)$$

$$f_u(x) = f_{v_1}(x)(1 - F_{v_2}(-x)) + f_{v_2}(x)(1 - F_{v_1}(-x)) + f_{v_1}(-x)F_{v_2}(x) + f_{v_2}(-x)F_{v_1}(x), \quad x > 0 \quad (8)$$

where $f(x)$ and $F(x)$ are the pdf and the cumulative density function (cdf) respectively. For the case of $d_c > 3$, $f_u(x)$ is obtained by recursive update with additional pdf $f_{v_i}(x)$ up to $i = d_c - 1$ as:

$$f_u(x) = G(\cdots G(G(f_{v_1}, f_{v_2}), f_{v_3}), \cdots, f_{v_{d_c - 1}})$$

where $G(\cdot)$ a shorthand notation of (12)-(13), which is the pdf of the output message for the case of $d_c = 3$.

Once we have $\overline{v}^{(l-1)}$, $\sigma_v^{2(l-1)}$ at $l-1$ th iteration, the mean and variance at l th iteration are tracked as[1)]

$$\overline{v}^{(l-1)}, \sigma_u^{2(l-1)} \rightarrow f_v^{(l-1)}, F_v^{(l-1)} \rightarrow f_u^{(l-1)}$$
$$\rightarrow \overline{u}^{(l-1)}, \sigma_u^{2(l-1)} \rightarrow \overline{v}^{(l)}, \sigma_v^{2(l)}$$

Using the Gaussian assumption on v, the pdf and cdf $f_v^{(l-1)}$, $F_v^{(l-1)}$ are obtained from $\overline{v}^{(l-1)}, \sigma_v^{2(l-1)}$. Then, the pdf $f_u^{(l-1)}$ is obtained by (5)-(6). The mean and variance $\overline{u}^{(l-1)}, \sigma_u^{2(l-1)}$ are numerically calculated from $f_u^{(l-1)}$. Finally, $\overline{v}^{(l)}$, $\sigma_v^{2(l)}$ at l th iteration is obtained by (7)-(8). The iterative mean and variance tracking is initialized by the received message from channel as:

$$\overline{v}^{(0)} = \overline{\lambda}_c$$
$$\sigma_v^{2(0)} = \sigma_{\lambda_c}^2 = 2\overline{\lambda}_c$$

This recursive calculation is executed for a sufficiently large number of iterations (e.g., 1000) to determine whether the message converges to correct codewords at a certain channel noise level. The convergence is determined when the probability error, equivalently the tail part ($v<0$) of the pdf $f_v(x)$ goes to zero. The threshold is defined as the maximum channel noise level for which the message converges. Table 1 shows the computed thresholds $\sigma_{\min-\Sigma}^{GA}$ based on the described min-sum algorithm for various pairs (d_v, d_c). The corresponding E_b/N_o values in dB are shown inside parentheses. For comparison, the thresholds $\sigma_{\Sigma-\Pi uct}^{GA}$ of sum-product algorithm are also shown in the samemanner. Fig. 2 shows the simulation results and thresholds for regular (3,6) LDPC code for block size N=10000 with min-sum and sum-product algorithms. The difference of thresholds (0.65dB) is well matched to the difference of simulation performance (0.5dB). The gap (0.15dB) between the thresholds and simulation performance is due to the fact that the thresholds is obtained on the infinite block size and iteration numbers.

Fig. 3 shows how the mean of output message is evolved from the mean of input message at a check node based on two different massage evolution algorithms. A check node which has two inputs and one output(i.e.,three edges) is considered. Fig. 3 shows the output message mean when one of the input mean varies from 0 to 40 and the other of input mean is fixed at 20. We can see the output mean converges to 20 which is minimum value of two input means for both cases. However, the output mean of sum-product algorithm converges faster then that of min-sum algorithm.

Ⅳ. Conclusion

In this paper, the density evolution of min-sum algorithm under the Gaussian

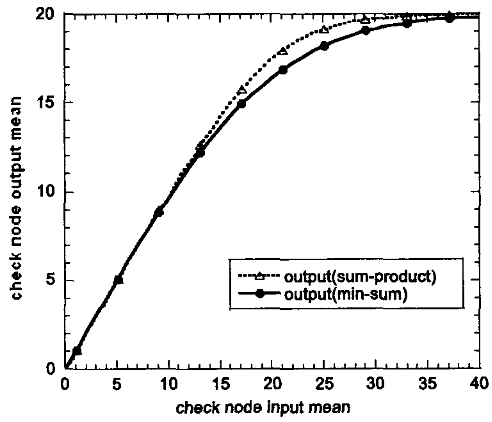


Fig. 3. Evolution of input message at a check node for two massage evolution algorithm (sum-product vs. min-sum)

1) 1))) In [3] and [8], the pdfs are tracked as : $f_v^{(l-1)} \rightarrow f_u^{(l-1)} \rightarrow f_v^{(l)}$ without Gaussian assumption.

approximation was presented and applied to various rate LDPC codes. This simplified new evolution technique can be used to obtain the capacities(thresholds) of the LDPC code. The capacities were computed based on the min-sum algorithm and were compared with the thresholds based on the sum-product algorithms. It was also shown that the computed threshold was well matched to the simulation result.

ACKNOWLEDGEMENT

This paper was supported by Konkuk University in 2003.

REFERENCES

[1] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599-618, February 2001

[2] S. Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 657-670, February 2001

[3] A. Anastasopoulos, "A comparison between the sum-product and the min-sum iterative detection algorithm based on density evolution," *Proc. Globecom Connf.*, pp. 1021-1025, 2001

[4] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619-637, February 2001

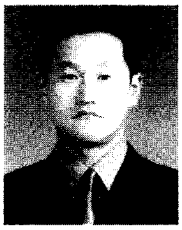
[5] J. Hagenauer, D. Offer, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429-445, March 1996.

[6] S. Y. Chung, G. D. Forney, T. J. Richardson, and R. L. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *IEEE Communication letters,* vol. 5, pp. 58-60, February 2001.

[7] L. Ping, S. Chan, and K. L. Yeung, "Iterative decoding of multidimensional concatenated single parity check codes," *Proc. International Conf. Communications*, pp. 131-135, June 1998.

[8] X. Wei and A. Akansu, "Density evolution for low-density parity-check codes under max-log-map decoding," in *Electronics Letters*, vol. 37, pp. 1125-1126, August 2001

허 준(Jun Heo) 정회원



1989년 2월 : 서울대학교 전자공학과 졸업
1991년 2월 : 서울대학교 전자공학과 석사
2002년 8월 : 미국 USC대학 전기공학과 공학박사
1991년 1월 ~ 1997년 8월: LG전자 중앙연구소
2003년 3월 ~ 현재: 건국대학교 전자공학부 조교수

<주관심분야> 통신공학, 부호이론