Efficient Construction of Successive Cancellation Decoding of Polar Codes using Logistic regression Algorithm

A.Albert raj¹, D.Judson², A.Beno³
Dept. of ECE, Sri Krishna College of Engineering and Technology¹,
St.Xaviers Catholic College of Engg², Dr. S. A. College of Engineering³
albert@skcet.ac.in¹, Judson_d2001@yahoo.co.in², benoaustin@gmail.com³

Abstract. In recent days, polar codes have attracted the attention of Industry and Academia. It is one of a linear block error correction codes. Polar codes are one of the capacity achieving algorithms over a wide range of channels, making them very attractive from a theoretical perspective. Polar codes are shown to be instances of concatenated codes. It has been shown that the effect of a polar code can be enhanced by showing the multistage decoding algorithm with log-likelihood based Successive List decoding (SCL). However, SCL decoders are no longer optimal in terms of frame error rate. It is believed that the machine learning based algorithm can improve the FER for the decoded bit. Thus, the logistic regression algorithm is proposed in this paper. The proposed method offers an optimized solution to the decoded bits. The simulation result shows that with sufficient training, this method can provide a lower frame error rate for different code lengths.

Keywords: Polar Code, Successive Cancelation, Channel polarization, Erasure channel, Logistic regression

1. Introduction

In the recent past, a number of research is being done in 5G wireless networks. Various channel codes are available to encode and decode the data. Turbo code and LDPC code are the key players for 3G and 4G wireless networks. However, they are unproven codes for 5G applications. Polar code has proven to be the best in channel coding theory for 5G wireless because of its low complexity and best decoding performance. Polar code [1] is one of the best error correcting codes with an explicit and efficient construction [2] with the capacity of encoding and decoding algorithms. They can achieve the channel capacity asymptotically for the code length 'n', when the underlying channel is memory less and has a discrete input alphabet [3]. Up until now, it is proved that the polar codes are provably achieved channel capacity with tractable decoding complexity. In addition, for 5G applications, these codes are one of the best and efficient [4, 5] algorithms. Therefore, they are seen as a breakthrough in coding and information theory.

2. Several techniques have been proposed to construct Successive Cancellations in polar codes. SC decoding is one of the promising algorithms which are sequential in nature that process bit by bit. A method of efficient construction of polar codes as well as degrading and upgrading is presented and analysed in [6]. Trifonov et al. [7] have shown that the approximation of Gaussian density evolution enables one to accurately predict the performance of polar codes. Improving the complexity of the channel coding for a new partial order on the synthetic channels of polar codes is presented in [8]. An approximate computation architecture and efficient channel computation for encoders and decoders for polar code are presented and subsequently implemented in [9]. It is observed that Reed Muller
codes have a lower probability of error than polar codes in MAP decoding. Keeping this in mind, a Reed-Muller code is attempted [10] to improve the performance of finite to polar codes. To show the advantages of short length codes, the maximization of minimum method is proposed in Bioglio et al. [11]. It is not necessary that the polar code generation with SC decoding algorithm gives the best error correction result under variants of SC decoding [12]. However, heuristic methods such as Monte-Carlo for Belief propagation algorithm [13-17] were implemented for polar codes and deal with these issues. Various methods were analysed in a number of research articles including [18-20] in which blind detection algorithms were implemented for polar codes. Artificial-intelligence based techniques have developed as the assuring components to generate a polar code. In general, both the encoding and the decoding process can experience considerable overhead and complexity, while the performance of decoders are closely attached to the characteristics of the polar code.

3. The rest of the paper is structured as follows. Section II briefly introduces the basic encoding algorithm for polar codes. Brief Introduction about the polar decoder is given in Section III. Section IV deals with the successive cancellation of decoder. Introduction about the Logistic Regression algorithm and the decoding process using Logistic Regression algorithm for the decoder, is briefed in Section V. Section VI details an implementation result. Finally, Section VII draws the conclusions.

2. Encoding

One of the big difference between LDPC code and polar code is the decoder. Unlike LDPC codes, Polar code decode each bit sequentially. The implementation occurs due to channel polarization. Channel polarization is a totally new effect observed by Arikan[1]. The most basic construction of the polar code is the so-called polar transform or Arikan transform, it has roots in Information theory and describes it in Information theoretic terms. The polarization (2x2) matrix or transform kernal is given as $G = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. This matrix enables the encoding of a two-bit input vector $u = [u_0; u_1]$ with the code word $d = [d_0; d_1]$ having $\begin{align*} d_0 &= u_0 \oplus u_1 \\ d_1 &= u_1 \end{align*}$. Similarly, the polar transform can be performed for any power of 2 ($G^4, G^8$).This transform can also be represented using a tree format which is much like a Turner code in LDPC as in figure 1 where $u_1$ and $u_2$ are the leaf nodes. $U^2$ is the root node where the basic transform is performed. ($u_1$ is xor-ed with $u_2$ and $u_2$ value is retained)

![Figure 1: Tree representation for polar encoder for 2 bits](image1)

Much like 2 bit transform, 4 bit transform can also be represented using the tree format as in figure 2

![Figure 2: Polar encoder for the 4 bit transform](image2)
In general, the polar codes are represented by PC (N, K), where K & N are the message and code word, respectively. In a polar code, the message is encoded for that its contents are first placed in the K dependable sub-channel indices corresponding to G, which initiate the information set 1. The enduring N − K bits are always set to zero. Then the encoding will be carried out with the matrix multiplication: 

\[ x = uG \]

along with the message with expanded frozen bits.

3. Polar Decoder

As in the encoding, the binary tree can also be used for decoding. This terminology is called simplified technique for Successive Cancellation. Successive cancellation is the first polar decoder proposed by Arikan (1). SC decoding is a very efficient decoding which is based on soft input and soft output in nature. And we ignore the soft output to make the hard decision. Normally, the decoder takes the received vector and puts out one code word. However, in the case of Successive Cancellation List decoding, it will display several possible code word lists as outputs. Typically, a number of researchers use cyclic redundancy check (CRC) for this purpose. It is another error detection code as it has very good error detection properties and simple to use. In the list decoding, the CRC is used in the physical layer.

The polar decoder receives a polar encoded message that has been transmitted over an AWGN channel. Due to the noise on the AWGN channel, the received message value contain signal offsets compared to the message that was sent. Polar Codes are designed to correct these offsets and reconstruct the correct message from the received data.

Most algorithms found were based on SC with additional complexity for better performance. Therefore, the Successive Cancellation decoding is chosen (see Figure 3) as the decoder to be constructed for this article. It was also intended to implement Successive Cancellation lists, since this has a significant code gain compared to SC.

![Figure 3: Polar Decoder](image)

4. Successive Cancellation Decoder

Because of the way the polar Codes are arranged, the received values, r, from the channel can be represented as a binary tree. In SC, each node in the binary tree has to do some operations.

a) Receive some beliefs from the parent node and send some information to the left child
b) Once it receives the decision from the left child, then it can send some beliefs to the right child
c) After receiving the decision from the right child, it combines them and sends them to the parent node. These are the three different operations that these nodes will do. Let’s start with the operation from the interior node (the nodes other than leaf nodes in the binary node). All three different operations that these interior nodes can perform are (L, R, U operations) are given in Figure 4.
In this figure, the interior nodes receive some beliefs \((L)\). Once the value is received by the left node, it will do some minsum operation \((f(r_1, r_2) = \text{sgn}(r_1)\text{sgn}(r_2), \min|r_1||r_2|)\). So there are \(m/2\) beliefs will be sent to the child node. After the child on the left completes its operations, it sends the value back to the right child. Now the \(G\) operation will be performed in the right child. It is a kind of repetition-like operation. The \(G\) operation will be performed for \(m/2\) values as follows
\[
G(r_1, r_2, b) = r_2 + (1 - 2b)r_1.
\]  

Now the decision \(\hat{u}_2\) from the right child will send back to their parent. Now that the parent would have received the \(\hat{u}_1\) and \(\hat{u}_2\) from left and right child leaves. The parent node will perform \(\hat{u}_{1+} \hat{u}_2, \hat{u}_2\) operation for the \(m\) value.

**A) Decision in the leaf node**

In the leaf node, the signal may be frozen or message bit. If the \(i^{th}\) value is frozen then the decision \(\hat{u}_1 = 0\). If \(i\) is a message position \(\hat{u}_1 = 0\) if \(L(u_1) \geq 0\); \(\hat{u}_1 = 1\) if \(L(u_1) < 0\); here is the sequence of decoder operation

1) At every node (if not leaf) do step L and go to the left child and perform minsum operation
2) When decisions are received from the left child, do step \(G\) (repetitive operations) and go to the right child
3) When the decision is received from right child do step \(U(\hat{u}_{1+} \hat{u}_2, \hat{u}_2)\) and go to parent.

The SC decoding for \(N=8\) is implemented in Figure 5. The construction of a binary tree is made of nodes and leaf nodes, where a node always has two children. A leaf node is a childless node. The operations in nodes can be divided into four states: Leaf, Left, Right and Up, where the leaves represent the decoded message bits.

Let us analyze the decoding operation on this simple diagram (see Figure 6).
It is clear from fig (7) that ‘a’ becomes zero when c=d= 0 or 1, so
\[ P(a=0) = P(c=0)P(d=0) + P(c=1)P(d=1) \]----------------------(2)
Similarly ‘a’ becomes one when c and d are 0/1 and 1/0 respectively. Therefore
\[ P(a=1) = P(c=0)P(d=1) + P(c=1)P(d=0). \]------------------------ (3)
Now let’s assume the likelihood ratio of ‘a’ as
\[ \text{LR}(a) = \frac{P(a=0)}{P(a=1)} \]------------------------------------------(4)
Equating (1 and 2), we get
\[ \text{LR}(a) = \frac{P(c=0)P(d=0)+P(c=1)P(d=1)}{P(c=0)P(d=1)+P(c=1)P(d=0)} \]--------------------------------------(5)
And
\[ \text{LR}(a) = \frac{1+ \text{LR}(c)\text{LR}(d)}{\text{LR}(c)+\text{LR}(d)} \]------------------------------------------(6)

Same way we shall calculate b
\[ \text{LR}(b) = [\text{LR}(c)]^{1-2a} \text{LR}(d) \]------------------------------------------(7)

At the receiver side, due to noise, let us imagine the signal received as follows
\[ \text{LR}(c) = \frac{P(c=0)}{P(c=1)} = 10 \text{ and } \text{LR}(d) = \frac{P(d=0)}{P(d=1)} = 0.1. \]

After substituting this value in equations 5 and 6, we get
\[ \text{LR}(a) = \frac{1+ \text{LR}(c)\text{LR}(d)}{\text{LR}(c)+\text{LR}(d)} = \text{LR}(a) \frac{1+10 \times 0.1}{10+0.1} = \text{LR}(a) = 0.2 \]------------------------------------------(8)
\[ \text{LR}(b) = 10^{1-2x1} \times 0.1 \to \text{LR}(b) = 0.01 \]------------------------------------------(9)

We have to decide the value of bits ‘a’ and ‘b’ is either 0 or 1. For this purpose, we will go for the one of the best classification algorithm, Logistic regression. The next section deals with the logistic regression and the decoder based on this algorithm. In the similar way, the decoding for 8 bits (see Figure 7) and the subsequent higher order bits are calculated.

Figure 7: Eight bit decoding circuit
5. Decoding using Logistic regression

Machine learning algorithm is trained using a labelled or unlabeled training dataset to produce a set of model. This model will give a prediction. This prediction is expected to give an accurate value. One of the best algorithms for Binary classification is Regression. In our case, the decoding can be performed using Logistic regression algorithm.

5.1 Signal Classification for Decoder output

Logistic Regression is used for predicting the outcome of a categorical criterion variable after the signal is experienced with noise (LR(a)=0.2(7) and LR(b)=0.001(8). In our case, the numerical values of 0 and 1 are assigned to the two outcomes of a binary variable. As we know, 1 stands for positive and 0 stands for negative response. Then the mean of this variable will be taken and considered as the proportion of positive responses. If p is the outcome of 1, then 1-p is 0 for the probability of ‘a’. Mathematically, the transformation can be written as

\[
y = \log\left(\frac{p}{1-p}\right) \tag{10}
\]

we model the log odds as a linear function of the explanatory variable:

\[
\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x \tag{11}
\]

where \( \beta_0 \) and \( \beta_1 \) are the parameters of the logistic regression, \( x \) is a variable and \( p \) is a binomial proportion. To find the estimates \( b_0 \) and \( b_1 \), we match the ‘0’ and ‘1’ model equations with the corresponding data equations. Similarly, the \( b_0 \) is the log odds for the ‘0’; This is the difference between log off of 0 and 1. This way, all incoming values for the decoders are classified either as 0/1.

6. Result

To illustrate the performance of the proposed polar code construction based on Logistic Regression method, different polar code constructions are evaluated under the pure Successive Cancellation decoder. The BER performance of the proposed Logistic regression algorithm and some existing SC based algorithms are simulated for different bit position (P), \( N = 1024, K = 512 \) (see Figure 8). Simulation result shows that the FER rate is better for higher bit position. The performance of FER for the learned code is simulated using Matlab (see Figure 10). As we can see in the figure, FER decreases while SNR increases for all kind of decoders. It can be also seen that expression (13) provides a very good estimate for the decoding error probability.

Further, while comparing the FER/SNR by logistic regression with SC decoder, it is found that the former performs slightly better than the SC and SCL decoder (see Figure 9) and its advantage is more pronounced in the high SNR region.

Figure 8: FER performance of the SC decoder for different bit precision (P), \( N = 1024, K = 512 \).
7. Conclusion

In this work, we have constructed the Successive Cancellation technique for the polar codes. We built the Logistic Regression algorithm to support the fifth generation wireless standard. In addition, we implemented the Matlab simulation for the SC decoder. It is found that this algorithm perfectly predicts the values for the decoding. Furthermore, the simulation results show that the proposed Logistic regression based decoding algorithm has better BER and FER performance than the existing SC algorithms. However, as the complexity of the Machine learning based algorithms are more, this architecture may be further be optimized by reducing the computation complexity. This will be the object of future research.

Reference
[1] E. Arıkan, “Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels,” IEEE Transactions on Information Theory, vol. 55, no. 7, pp. 3051–3073, July 2009.
[2] 3rd Generation Partnership Project (3GPP), “Multiplexing and channel coding,” 3GPP 38.212 V.15.3.0, 2018.
[3] K. Niu, K. Chen, J. Lin, and Q. T. Zhang, “Polar codes: Primary concepts and practical decoding algorithms,” IEEE Communications magazine, vol. 52, no. 7, pp. 192–203, July 2014.
[4] H. Vangala, E. Viterbo, and Y. Hong, “A comparative study of polar code constructions for the AWGN channel,” in arXiv preprint arXiv: 1501.02473, Jan. 2015.
[5] R. Mori and T. Tanaka, “Performance of polar codes with the construction using density evolution,” IEEE Communications Letters, vol. 13, no. 7, pp. 519–521, July 2009.
[6] I. Tal and A. Vardy, “How to construct polar codes,” IEEE Transactions on Information Theory, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.
[7] P. Trifonov, “Efficient design and decoding of polar codes,” IEEE Transactions on Communications, vol. 60, no. 11, pp. 3221–3227, Nov. 2012.
[8] M. Mondelli, S. H. Hassani, and R. Urbanke, “Construction of polar codes with sublinear complexity,” in IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, June 2017.
[9] C. Condo, S. A. Hashemi, and W. J. Gross, “Efficient bit-channel reliability computation for multimode polar code encoders and decoders,” in IEEE International Workshop on Signal Processing Systems (SiPS), Lorient, France, Oct. 2017.
[10] M. Mondelli, S.H. Hassani, and R.L. Urbanke, “From polar to Reed- Muller codes: A technique to improve the finite-length performance,” IEEE Transactions on Communications, vol. 62, no. 9, pp. 3084–3091, Sept. 2014.
[11] V. Bioglio, F. Gabry, I. Land, and J.-C. Belfiore, “Minimum-distance based construction of multi-kernel polar codes,” in IEEE Global Communications Conference (GLOBECOM), Singapore, Dec. 2017.
[12] M. Mondelli, S. H. Hassani, and R. L. Urbanke, “Scaling exponent of list decoders with applications to polar codes,” IEEE Transactions on Information Theory, vol. 61, no. 9, pp. 4838–4851, Sept. 2015.
[13] J. Guo, M. Qin, A. Guilleren F’abregas, and P. H. Siegel, “Enhanced belief propagation decoding of polar codes through concatenation,” in IEEE International Symposium on Information Theory (ISIT), 2014, Honolulu, HI, USA, June 2014.
[14] U. U. Fayyaz and J. R. Barry, “Low-complexity soft-output decoding of polar codes,” IEEE Journal on Selected Areas in Communications, vol. 32, no. 5, pp. 958–966, May 2014.
[15] J. Vijila, A. Albert Raj 2020 “Ameliorate Security by Introducing Security Server in Software Defined Network” Journal of Computers, Materials & Continua Vol 62 No3 pp 1077-1096.
[16] A Albert Raj, A Beno R Jaisakthi 2020 “Security Enhancement of Information using Multilayered Cryptographic Algorithm” IEEE -4th International Conference on Trends in Electronics and Informatics (ICOEI) pp-348-353.
[17] Dr. A. Albert Raj, J. Vijila, 2020 “Design of Indoor Air Quality Monitoring System to Ensure a Healthy Universe” International Conference on Smart Electronics and Communication (ICOSEC) pp 1123-1127.
[18] P. Yuan, T. Prinz, and G. Bocherer, “Polar code construction for list decoding,” in IEEE International ITG Conference on Systems, Communications and Coding (SCC), Rostock, Germany, Feb. 2019.
[19] C. Condo, S. A. Hashemi, A. Ardakani, F. Ercan, and W. J. Gross, “Design and implementation of a polar codes blind detection scheme,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 66, no. 6, pp. 943–947, June 2018.
[20] Weixiang Li; Weiming Zhang; Li Li; Hang Zhou; Nenghai Yu Designing “Near-Optimal Steganographic Codes in Practice Based on Polar Codes” IEEE Transactions on Communications Year: 2020, Volume: 68, Issue: 7.