Polar码基本原理

一、polar码简介

Polar码在2009年首次被Arikan提出[1]。它以理论可达香农限，以及简单的编译码算法等优点迅速成为编码界的研究热点。目前，polar码主要研究方向有以下几个：一是polar码的码字构造。Polar码的原理基础是信道极化：在无限码长下，以一定的编码方式进行编码，则编码后的“比特信道”将极化为“好信道”和“坏信道”。这里的“比特信道”实际上是串行消除译码时的信道。以码长4为例，比特信道分别为“、、、”。但在有限码长下，“比特信道”极化将不完全，即只有部分信道是“好信道”或者是“坏信道”，当中还有部分信道介于“好信道”和“坏信道”之间，这里我们将其称为“不完全极化信道”。如何在不完全极化信道中挑选出相对较好的“比特信道”就是polar码码字构造问题。二是polar码译码算法研究。目前polar码译码算法主要有SC算法[1]、SCL算法[2, 3]、CA-SCL算法[4]、BP算法[5]、SCAN算法[6]，以及各种算法的简化版本[7-13]。其中SC算法最初由Arikan提出，但其在码长有限长情形下，性能一般。SCL是SC算法性能提升的改进版本，原理是在SC的基础上提供了多条路径；而CA-SCL是在SCL的基础上对信息比特进行了循环冗余校验。它通过简单的校验就可以带来性能的极大提升。目前基于CA-SCL译码算法，polar码性能已经优于LDPC码。SC、SCL、以及CA-SCL算法均是硬输出算法，即最终输出的是0、1比特序列而不是对应的LLR值。为了在一些联合设计中使用polar码，就必须提供polar码的软输出算法，即输出对应比特的LLR值。BP和SCAN算法就是软输出算法，由于BP和SCAN算法在消息传递和递归规则上不同，导致了两种算法的译码延时和收敛速度不同。在译码延时上，BP消息传递采用“洪水”规则[12]，SCAN采用类SC算法的串行消除规则，因此BP算法的译码延时好于SCAN算法。但是SCAN算法的收敛速度明显好于BP算法。一般而言，BP算法需要40-50次的迭代过程，而SCAN算法1次迭代就能达到稍好于SC算法的性能。在性能表现上，这几种算法排序如下：CA-SCL>state-of-the-art(LDPC、Turbo)>SCL>BP=SCAN>SC。三是polar码在具体通信信道以及通信场景中的应用，如衰落信道、MIMO信道等。为了读者能够迅速了解和掌握polar码的基本原理，下面对polar码编码原理以及SC、SCL以及BP和SCAN译码算法作简要介绍。Polar码码字构造研究方法可以参考文献[14-18]。

二、信道极化原理

Polar码的基本原理是信道极化。Arikan在他的文章中[1]指出如果对二进制对称信道进行特定的“组合”和“拆分”，则拆分后的“比特信道”将呈现极化现象：一部分“比特信道”的对称信道容量趋近于1，而其余部分“比特信道”的对称信道信道容量趋近于0。如果发送端将源信息比特放置在“好的比特信道上”，而在“坏的比特信道上”放置固定比特，如0，同时在接收端采用连续消除译码算法，则该码字的码率 在码长 时将可以达到信道容量。

**2.1 信道组合**

所谓的信道组合其实并不是指将实际的物理信道组合起来。实际通信系统中，信源端发送符号将经过一个特定的物理信道，如果信源端无记忆，而实际物理信道在一段时间内保持不变并且也是无记忆的话，在某一个物理信道上连续发送个符号将等效于发送的个符号经过了信道。如果我们在发送端对这个符号进行特定方式的组合，则将等效于信道的组合。



图 1 信道示例

如图 1所示，左图是发送端发送两个符号至对称无记忆信道情景。由于信源独立，并且信道也是无记忆的，因此，此时信道可以表示为右图所示，即等效为两个独立对称无记忆信道。



图 2 N=2信道组合

进一步，如果我们对发送的信息源符号以一定方式进行组合，如图 2所示，并记组合后的信道为，则：



类似于图 2所示过程，我们称其为信道组合。进一步，我们可以将两个信道组合为一个信道，如图 3所示。依次往复，直至递归到两个信道组合为信道，如图 4所示。其中是reverse-shuffle置换矩阵，满足关系。



图 3  信道示例



图 4  信道

**2.1 信道拆分**

信道组合不是物理意义上的将信道组合起来，同样，信道拆分也不是将实际物理信道拆分开来。考虑发送端发送信源信息，接收端接收符号为。如果按照极大似然估计译码，则译码信息为。显然，当较大时，译码复杂度将不可承受。为此考虑连续串行消除译码算法（Arikan证明其译码复杂度为），即先根据译码，然后在假设完全译码准确的基础上，根据译码。依次类推。最后假设译码正确的基础上根据  译码。

此时称信道、和为拆分后的比特信道。若定义拆分后的比特信道为，则：



**证明：**



综上可知，拆分后的比特信道实际上就是连续消除串行译码时对应的各个逻辑信道。

**2.3 信道极化定理**

Arikan指出拆分后的比特信道将呈现极化趋势，一部分比特信道变“好”，另一部分比特信道变“差”。并且随着增大，极化趋势将更明显。

**（信道极化定理）：**给定任意 ，其比特信道将随增大而呈现极化特性，当时：

* + 1. 对于任意，有；
    2. 对于任意，有；
    3.  。

其中为对称信道容量，表示数目。

如图所示是时，SNR=0dB时，BEC信道对应的各个“比特信道”的对称信道容量

由于polar码选择在“比特信道”容量为1的比特信道上发送信源信息（“比特信道”容量为0的比特信道上发送固定信息，如0），当码长为并且时，“比特信道”容量为1的比特信道数量有个，因此该码字所对应的信息量为。平均每个码元对应的信息量就为。故polar码是第一个能在理论上证明达到信道容量的码字。

二、polar码构造原理

三、polar码编码原理

以polar码码长为例，在找到polar码的固定比特位置和信息比特位置后，采用如图 5所示的编码方式。

生成矩阵可以表示为：





其中为bit-inverse矩阵[1]。

|  |
| --- |
| 图 5 N=8时polar码编码因子图 |

四、SC译码原理

Polar码的SC译码原理只需要弄清楚两点：第一是单位因子图上的概率传递公式，二是递归顺序。首先笔者先对单位因子图上的概率传递公式进行介绍。

3.1 单位因子图概率传递公式

如图 6所示是polar码的单位因子图，在该因子图上有8个值，分别是代表向左传递的LLR值，代表向右传递的硬比特信息。

传递公式为：



其中 表示：









|  |
| --- |
| 图 6 SC译码单位因子图 |

3.2 递归顺序



图3 递归顺序举例

首先，译码器应计算标号为1的点的左消息LLR值，而由图2和公式1.3知道，此时必须知道标号为2和3点的LLR值，同理要计算2的LLR应计算4和5的LLR值，而3需要计算6和7的LLR值，而4、5、6、7的LLR值可以直接根据信道传输过来的LLR值直接进行计算。通过以上递归过程则得到了1点的LLR值。此时先判定该点是否是固定比特，如果是固定比特则直接译为0（显然）；如果是信息比特，则根据刚计算得到的LLR值进行判决，LLR值大于0判为0、LLR值小于0判为1。然后可以根据公式1.5计算标号为8的LLR值。注意由于此时点8对应于一个单位因子图的左下方，右边两个点即点2和3的LLR值在计算点1的LLR值已经计算得到。因此不需要进行递归。计算得到点8的LLR值后，同样对其进行判决。注意在计算得到单位因子图左下方点的LLR值后，需要向右传递硬判决比特值。即根据公式1.6和1.7计算得到点2和3的比特值。而此时2和3为单位因子图的左上方，因子不再需要向右传递硬判决比特值。再考虑点9LLR值计算，由公式1.3可知，必须首先计算点10和11的LLR值。而点10和11对应于一个单位因子图的左下方，根据公式1.5和点2和点3的硬判决比特值进行计算。不再需要往下递归。再计算点12的LLR值，点12LLR值可以根据公式1.5和点9的硬判决比特值进行计算，然后进行判决。由于点12为单位因子图的左下方，因此需要向右传递硬判决比特值。此时根据公式1.6和1.7计算点10和点11的比特值，由于点10和点11均为单位因子图的左下方，因此需要继续向右传递比特值至点4、5、6和7。此时点4、5、6、7均为单位因子图的左上方，因此不需要继续向右传递比特值。同理直至SC译码器将的LLR值全部计算出来，并判决得到比特值。LLR值计算的顺序见图3标号所示。

3.3 节点标号

在理解了上述SC译码原理后就可以进行译码了。然而，为了编程实现上述描述的LLR值递归过程以及硬判决比特值向右传递过程，需要对图3所示中的点进行标号，以便编程时按照某种顺序进行。



图4 节点标号

Polar码因子图中的每个点都可以用一个来表示。其中 代表因子图的哪一层。一个码长为 为的因子图含有 层，从右端也即码字端向消息端分别表示为。表示为红色方框的序号，对于第 层含有的红色方框数为个，分为标为 。而 表示为红色方框中点的序号如图4中 层方框标号为的方框中均有四个点，分别标为。显然对于 层，每一个红色方框中均有 个点，分别标号为。因此对于每一个点的左消息LLR值可以表示为每一个点的比特值可以表示为。

因此下面的SC译码算法就很容易理解了。

|  |
| --- |
| **Algorithm1** SC Main Function |
| **Input:**  the LLR of received signal **y**  **Output:** decoded message  **for**  updateLLR(n,)  **if**   is frozen **do**  set  **else**    **end**  **if**   is odd **do**  updateB(n,)  **end**  **end**  **return** |

|  |
| --- |
| **Algorithm2** SC updateLLR(,) |
| **if**  **then** **return**    **if**  **then** updateLLR(,)  **for**  **do**  **if**  **then**    **else**    **end**  **end** |

|  |
| --- |
| **Algorithm3** SC updateB(,) |
| **if**  is odd **then**  **for**  **do**      **end**  **if**  is odd **then** updateB(, )  **end** |

五、SCL译码原理



图5 SCL译码原理图

如图5所示是SCL译码原理图。在这里我们假设 均是信息比特。在SC译码下， 的LLR值将被计算，进而判决出是0还是1。然后根据判决的 值利用公式1.5计算 的LLR值，同时利用公式1.6和1.7对信息比特向右进行传递。此时如果 译码错误，则将错误将向后传递。为了改进错误向后传递情况，首先计算出 的LLR值，继而不进行判决，假设 可能为0或者为1。当 是0时，利用公式1.5对 的LLR值进行计算，并用公式1.6和1.7向右传递信息比特。同样当 是1时，利用公式1.5对 的LLR值进行计算，并用公式1.6和1.7向右传递信息比特。同理，再计算出的两种可能LLR值后也不进行判决，假设其为0或者1，这样继续计算 。依次往复，对于一个码长为 的译码系统，路径数将达到 个，此时的计算量太大。为此，需要对路径进行删减。而删减准则则需要一个量度来决定哪些路径被剪。

考虑译到 时，此时的概率为：



因此此时的概率值越大，表示译码为时可能性最大。路径应该保存下来。而如果是基于LLR的SCL算法，路径度量可以采用



具体证明详见论文[3]。

六、SCAN译码原理

虽然SC译码时，信道接收符号的LLR值将向左传递成为信息比特的LLR值，但是该LLR值不是真正意义上的“软输出”。该值是一个畸形的LLR值，不能用在一些联合设计系统上。原因如下：

考虑译 时，我们计算的是概率值 ，此时我们假设是等概率为0和1.

而译 时，我们计算的是概率值 ，此时我们假设是确定值。

而实际软输出算法中，我们不应有以上两种假设，即完全无信息和拥有完全信息。我们需要给每一个信息比特一个概率值，并进行逐步迭代，最终获得每一个信息比特的LLR值。

经理论推导，在一个单位因子图中概率值满足一下关系：





图6 SCAN算法单位因子图概率传递图示

而传递规则完全参照SC算法。详细算法步骤参照论文[19]。BP算法则是按照“洪水”规则，详见论文[20]

七、笔者注

其实关于上述polar码的译码原理，最重要的只要掌握SC算法就可以，毕竟其他所有算法都是在它的基础上改进而来的。而polar码的SC算法，最复杂的就是递归顺序和标号，因为它每一个点采用了三个维度来刻画。当然只要知道图4结构，理解了这三个维度的物理意义，就不难理解和掌握SC算法了。关于SCL算法，复杂的在于路径如何增删，以及如何将保存的L值和B值附到路径上。由于笔者能力有限，在该方面描述上可能会越解释越糊涂，所以读者最好参考论文[2]以及笔者编写的程序[见附件]。笔者在当时理解SCL算法时，开始并不了解路径度量准则，因为论文[2]将它放在了论文末尾，而读者在读这里之前早已经晕头转向了。一个比较笨但是绝对实用的方法理解SCL算法就是跟进程序，调制码长为8，一步一步走，结合论文[2]中的各个子算法和程序中的子函数对照起来看。

## 参考文献

[1] Arikan E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels [J]. IEEE Transactions on Information Theory. 2009, 55 (7): 3051–3073.

[2] Tal I, Vardy A. List Decoding of Polar Codes [J]. IEEE Transactions on Information Theory. 2015, 61 (5): 2213–2226.

[3] Balatsoukas-Stimming A, Parizi M B, Burg A. LLR-based successive cancellation list decoding of polar codes [C]. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP): 3903–3907.

[4] Zhang Q, Liu A, Pan X, et al. CRC Code Design for List Decoding of Polar Codes [J]. IEEE Communications Letters. 2017, 21 (6): 1229–1232.

[5] Arikan E. A performance comparison of polar codes and Reed-Muller codes [J]. IEEE Communications Letters. 2008, 12 (6): 447–449.

[6] Fayyaz U U, Barry J R. Polar codes for partial response channels [C]. In 2013 IEEE International Conference on Communications (ICC): 4337–4341.

[7] Giard P, Balatsoukas-Stimming A, Sarkis G, et al. Fast low-complexity decoders for low-rate polar codes [J]. Journal of Signal Processing Systems. 2016: 1–11.

[8] Lin J, Sha J, Li L, et al. A high throughput belief propagation decoder architecture for polar codes [C]. In 2016 IEEE International Symposium on Circuits and Systems (ISCAS): 153–156.

[9] Lin J, Yan Z, Wang Z. Efficient Soft Cancelation Decoder Architectures for Polar Codes [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2017, 25 (1): 87–99.

[10] Lin J, Xiong C, Yan Z. Reduced complexity belief propagation decoders for polar codes [C]. In 2015 IEEE Workshop on Signal Processing Systems (SiPS): 1–6.

[11] Alamdar-Yazdi A, Kschischang F R. A Simplified Successive-Cancellation Decoder for Polar Codes [J]. IEEE Communications Letters. 2011, 15 (12): 1378–1380.

[12] Yuan B, Parhi K K. Early Stopping Criteria for Energy-Efficient Low-Latency Belief-Propagation Polar Code Decoders [J]. IEEE Transactions on Signal Processing. 2014, 62 (24): 6496–6506.

[13] Fayyaz U U, Barry J R. Low-Complexity Soft-Output Decoding of Polar Codes [J/OL]. IEEE Journal on Selected Areas in Communications. 2014, 32 (5): 958–966. http://­ieeexplore.ieee.org/­ielx7/­49/­6803816/­06804940.pdf?tp=&arnumber=6804940&isnumber=6803816.

[14] Vangala H, Viterbo E, Hong Y. A Comparative Study of Polar Code Constructions for the AWGN Channel [J]. Mathematics. 2015.

[15] Cheng F, Liu A, Zhang Q, et al. Codes design based on EXIT chart for polar coded BICM-ID [C]. In 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). March 2017: 1129–1133.

[16] Mondelli M, Hassani S H, Urbanke R. Construction of polar codes with sublinear complexity [C]. In 2017 IEEE International Symposium on Information Theory (ISIT). June 2017: 1853–1857.

[17] Sun S, Zhang Z. Designing Practical Polar Codes Using Simulation-Based Bit Selection [J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems. 2017, PP (99): 1–1.

[18] 吴道龙. 极化码构造与译码算法研究 [D]. 西安: 西安电子科技大学, 2016.

[19] Fayyaz U U. Polar code design and decoding for magnetic recording [D]. [S. l.]: Georgia Institute of Technology, 2014.

[20] Xu J, Che T, Choi G. XJ-BP: Express Journey Belief Propagation Decoding for Polar Codes [C]. In 2015 IEEE Global Communications Conference (GLOBECOM): 1–6.