

LDPC Sum-Product Decoding Based on Synchronized Information

Yangyang Chen, Lenan Wu

Radio Engineering Department of Southeast University, Nanjing Jiangsu
Email: southernduck@seu.edu.cn

Received: May 22nd, 2015; accepted: Jun. 8th, 2015; published: Jun. 12th, 2015

Copyright © 2015 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Low density parity check code (LDPC) is a subclass of linear block code with sparse check matrix. Sum-product algorithm is efficient for decoding LDPC theoretically, but not practically due to its complexity. Improved algorithms including minimum sum algorithm reduce computational complexity by approximation but performance is degraded. In this paper, we introduce corrected sum-product algorithm based on synchronized information, which improves the practicability by reducing the iterations without performance degradation.

Keywords

Sum-Product Algorithm, Low Density Parity Check Code, Linear Block Code, Channel Coding

利用同步信息修正的LDPC和积译码

陈洋洋, 吴乐南

东南大学信息科学与工程学院, 江苏 南京
Email: southernduck@seu.edu.cn

收稿日期: 2015年5月22日; 录用日期: 2015年6月8日; 发布日期: 2015年6月12日

摘 要

低密度奇偶校验码(LDPC)是一类具有稀疏校验矩阵的线性分组码, 和积算法是理论可行的LDPC软解码

算法, 但其实用受制于计算复杂度。最小和算法及其它已知改进算法采用近似来有效简化每次迭代的过程, 以部分性能为代价, 在一定程度上弥补了前述缺陷。本文提出了基于同步信息修正的和积算法, 在保持原算法每次迭代复杂度以保证性能的前提下, 利用同步信息对算法进行修正, 有效减少了算法迭代次数, 因此降低了计算复杂度, 提高了实用性。

关键词

和积算法, 低密度奇偶校验码, 线性分组码, 信道编码

1. 引言

数字信号在传输过程中由于受到信道噪声和干扰的影响容易被错判。采用均衡可纠正由乘性干扰引起的码间串扰, 而加性干扰的影响则需要用其它办法解决。若在合理选择调制制度、解调方法以及发送功率后, 仍无法抵抗加性干扰对解调性能的大幅下降, 就应该考虑采用差错控制技术。

信道编码是物理层上的差错控制技术, 以添加冗余降低信息传递速率为代价提高传输可靠性。根据冗余加到信息中方式的不同, 纠错编码分为两类: 分组码和卷积码, 均获得了实际应用。历史上人们更喜欢用卷积码, 因其可使用软判决 Viterbi 译码, 而多年来分组码则一直被认为无法使用高效的软判决译码。然而, 关于线性分组码软判决译码算法的理论和实际, 近年来都获得了很大发展。1962 年, Gallager 给出了一类称为低密度奇偶校验码(LDPC)的分组码和两种迭代概率译码算法。随后, Tanner 将 Gallager 的概率译码算法扩展到更一般的情况, 用子码代替简单的单奇偶校验方程组来定义奇偶校验[1]。

LDPC 码有多种译码方法。根据消息迭代过程中传送消息形式的不同, 可将 LDPC 的译码方法分为硬判决译码和软判决译码。若在译码过程中传送的消息是比特值, 称之为硬判决译码, 若传递的消息是与后验概率相关的消息, 称之为软判决译码。软译码算法主要有和积算法、最小和算法、简化的最小和算法、归一化的最小和算法等。后三者主要通过简化每次迭代的过程, 达到简化计算的目的, 取得了一定的效果[2]。

本文提出了基于同步信息修正的和积算法, 在保持原算法每次迭代复杂度以保证性能的前提下, 利用同步信息对算法进行修正, 有效减小了算法迭代次数, 因此减小了计算复杂度, 提高了算法的实用性。

2. 和积算法译码流程[1] [3]

初始化: 设定最大迭代次数 I , 初始迭代次数 $i=1$, 计算变量节点来自于信道的初始概率似然比消息, 对每一个变量节点 i 以及与其相邻的校验节点 $j \in C(i)$, 设定变量节点向校验节点传递的初始消息:

$$L^{(0)}(q_{ij}) = L(p_i) \quad (1)$$

步骤 1:

1) 水平方向, 对所有的校验节点 j 和与其相邻的变量节点 $i \in R(j)$ 。第 1 次迭代时, 计算变量节点传向校验节点的消息:

$$\tanh\left(\frac{1}{2}L^{(l)}(r_{ji})\right) = \prod_{i \in R_j \setminus i} \tanh\left(\frac{1}{2}L^{(l-1)}(q_{ij})\right) \quad (2)$$

2) 垂直方向, 对所有的节点变量 i 和与其相邻的校验节点 $j \in C(i)$ 。第 1 次迭代时, 计算校验节点传向变量节点的消息:

$$L^{(l)}(q_{ij}) = L(p_i) + \prod_{j \in C_i \setminus j} L^{(l)}(r_{ji}), L^{(l)}(q_i) = L(p_i) + \prod_{j \in C_i} L^{(l)}(r_{ji}) \quad (3)$$

步骤 2: 判决, 并判定是否应该结束迭代译码过程:

- 1) 判决, 若 $L^{(l)}(q_i) > 0$, 则 $c_i = 1$ 。从而得到判决后的输出码字。
- 2) 若 $Hc^i = 0$ 或 $i = I$, 结束译码, 转至步骤 3, 否则转至步骤 1。

步骤 3: 输出译码结果。

3. 同步信息修正和积算法译码流程

初始化: 设定最大迭代次数 I , 初始迭代次数 $i = 1$, 计算变量节点来自于信道的初始概率似然比消息, 对每一个变量节点 i 以及与其相邻的校验节点 $j \in C(i)$, 按式(1)设定变量节点向校验节点传递的初始消息并将帧头中码元“1”对应的初始软信息置为 $-T$, 码元“0”对应的初始软信息置为 T 。

步骤 1:

1) 水平方向, 对所有的校验节点 j 和与其相邻的变量节点 $i \in R(j)$ 。第 1 次迭代时, 按式(2)计算变量节点传向校验节点的消息。

2) 垂直方向, 对所有的节点变量 i 和与其相邻的校验节点 $j \in C(i)$ 。第 1 次迭代时, 按式(3)计算校验节点传向变量节点的消息。

3) 利用同步信息对两个消息矩阵 Q 和 R 进行修正。假设同步有效, 那么帧头数据将是确定的, 于是在每次迭代的过程中不必更新相应消息的可信度。采取的策略是固定前述两个矩阵相应列的消息不变, 如此便可避免有效信息的减小并且可以持续向其它节点输出修正信息。具体做法为: 若帧头中某一位确定为“0”, 那么消息固定为 T , 若某一位为“1”, 则消息固定为 $-T$ 。

步骤 2: 判决, 并判定是否应该结束迭代译码过程:

- 1) 判决, 若 $L^{(l)}(q_i) > 0$, 则 $c_i = 1$ 。从而得到判决后的输出码字。
- 2) 若 $Hc^i = 0$ 或 $i = I$, 结束译码, 转至步骤三, 否则转至步骤一。

步骤 3: 输出译码结果。

4. 性能分析

当信噪比高于 3 dB 时, 改进算法不仅不增大误码率, 而且提高了原算法的平均计算收敛速度, 为硬件资源的复用提供了一定的理论依据, 仿真结果如图 1 所示。可以看到, 在信噪比为 3 dB 时, 原算法的

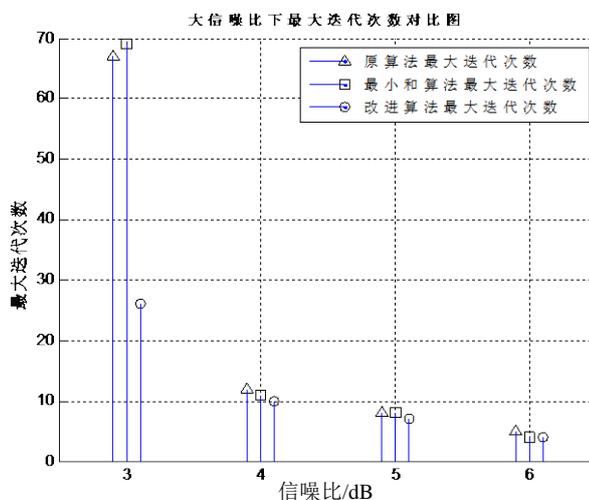


Figure 1. Max iteration under large SNR

图 1. 大信噪比下最大迭代次数对比图

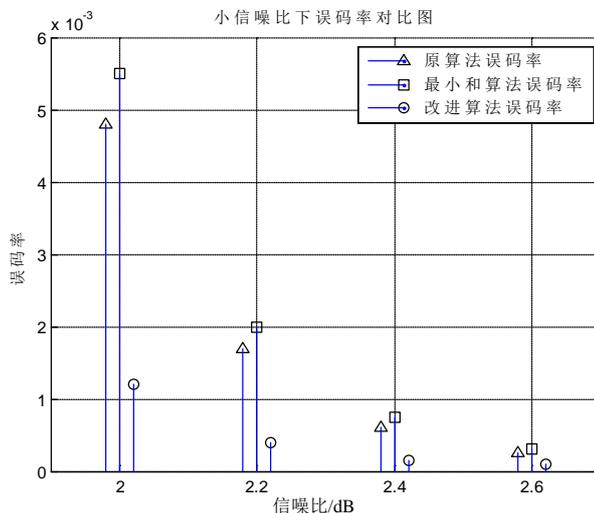


Figure 2. BER under small SNR
图 2. 小信噪比下误码率对比图

最大迭代次数为 67 次，最小和算法的最大迭代次数为 69 次，而改进算法是 26 次，证明了改进算法的高效性。

当信噪比低于 3 dB 时，改进算法降低了误码率，提高了通信系统的可靠性。仿真结果如图 2 所示。

5. 结论

无论是连续传输模式还是突发传输模式，物理层均进行组帧和解帧操作。帧同步是可靠传输的基础，而帧同步的实现意味着帧头识别的完成。如前述，在帧同步准确时，帧头数据是确定的，在每次迭代的过程中不更新相应消息的可信度，以此避免有效信息的减小并且可持续向其它节点输出修正信息。此时，该算法在高信噪比下能提高计算效率且在低信噪比下能降低误码率。而若帧同步出错，则物理层传输的错误已经形成，只能依靠上层进行纠正，在此基础上该算法并不会使情况恶化。因此，该算法锦上添花，能改善系统的整体性能。

参考文献 (References)

[1] 黄海艺 (2013) 低密度奇偶校验 (LDPC) 码改进译码算法研究. 华南理工大学, 广州.
 [2] 孙斌, 王钢, 杨文超, 等 (2015) 一种改进型 LLR BP 算法的 LDPC 译码研究. *无线电工程*, 3, 002.
 [3] 龚莉萍, 陈云榕, 胡凯 (2009) LDPC 编译码技术研究. *通信技术*, 7, 10-12.