Design of Non-Binary Decoders Using the Role Model Framework

Jossy Sayir
j.sayir@ieee.org
Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK

Abstract—The concept of a message-passing decoder for non-binary low-density parity-check (LDPC) codes that operates on symbol rankings instead of probability distributions is investigated. The optimal Bayesian conversion from ranks back to probabilities is derived, and the hypothetical performance of a rank-based algorithm is measured by converting the rank-valued messages back to probabilities and implementing the node operations in the probability domain. While the performance of a pure rank-based decoder is disappointing, further analysis and performance evaluation shows that a hybrid decoder that switches to the probability domain for its final iterations exhibits a performance at par with the best known low complexity decoders without causing a significant increase in the overall number of iterations.

I. INTRODUCTION

Non-Binary low-density parity-check (LDPC) codes are of practical interest due to their superior performance in particular for moderate block lengths. However, the complexity of message passing decoding for non-binary codes has been a hurdle in the way of their adoption in communication systems and standards. For binary codes, there exist a wide range of decoders starting from the full sum-product decoder in the probability domain and down to the very low complexity Gallager A and B [1] algorithms that operate on 1-bit messages. For non-binary codes, the complexity of operations in the sum-product decoder grows sub-quadratically in the alphabet size. Various simplifications like the Extended Min-Sum (EMS) algorithm [2] bring the complexity down somewhat but none sufficiently to make non-binary codes perform better than binary codes at equal complexity. There is certainly no equivalent to the very simple Gallager decoders for non-binary codes.

This paper will present an attempt to develop Gallager-type message passing decoders for non-binary LDPC codes. The role of the sign of the log likelihood ratio represented by the bit-valued message in the Gallager algorithms is taken by the ranked list of symbols in order of decreasing extrinsic probability. We will study the performance and mutual information analysis of an idealized message-passing algorithm that performs Bayesian operations in the variable and check nodes under the constraint of having to transmit ranked lists of symbols over the edges of the factor graph. Implementing these Bayesian operations under constraint is made possible by use of the role model framework introduced by the author in [3] and shown in the non-parametric case to coincide with Monte Carlo integration in [4].

We will show that the performance of our algorithm is insufficient and proceed to demonstrate how and why a hybrid decoder that switches from the rank-only decoder to a full sum-product decoder in its last iterations exhibits a good performance at par with the best known simplified decoders for non-binary LDPC codes without giving rise to a significant increase in the number of iterations required. Our algorithm does not quite fulfill the promise of this paper’s title as it remains in the hypothetical domain and we have not as yet studied techniques for its low complexity implementation. We do however believe that our results justify further research on rank message passing.

II. RANK DECODING

We are interested in message passing algorithms defined on factor graphs where the messages are constrained to being ranked list of symbols over GF(q), i.e., one of the q! possible permutations of the list (0, 1, …, q − 1).

We resort to the common assumptions of asymptotic message independence under random interleaving and block length tending towards infinity. This would allow us in principle to apply Bayesian theory to the problem of finding the optimal mapping, with one caveat. If only the input messages were constrained to being rank-valued, then Bayesian theory could be directly applied to computing an extrinsic q-ary probability distribution given the d − 1 rank-valued input observations to a node. However, once we introduce a constraint on the output alphabet, then applying Bayesian theory is not evident. This is equivalent to a quantization constraint on the representation of the Bayesian output distribution. Defining the right optimality criterion for such a quantization is a difficult problem.

Supposing that we had no constraint on the output alphabet of a node, it is worth investigating how Bayesian theory would be applied. Let us denote by R2, R3, …, Rd the rank-valued input messages to a node, which we assume to be observations of variables X2, X3, …,Xd, and a possible additional channel message Y1 (the latter is only present for variable nodes and is not constrained to ranks). The aim of the Bayesian decoder is to compute, for any realization y1 and r2…rd, the extrinsic...
distribution
\[ P(x_1|y_1, r_2 \ldots r_d) = \sum_{x_{2} \ldots x_d} P(x_1 \ldots x_d|y_1, r_2 \ldots r_d) \]
\[ \propto \sum_{x_{2} \ldots x_d} P(x_1 \ldots x_d) P(y_1, r_2 \ldots , r_d|x_1 \ldots x_d) \]
\[ \propto \sum_{x_{2} \ldots x_d} f(x_1 \ldots x_d) P(x_1|y_1) \prod_{i=2}^{d} P(x_i|r_i) \]

where \( f(.) \) is an indicator random variable for the node constraint, i.e., 1 if all its values are equal and 0 otherwise for variable nodes, or 1 if the weighted sum in \( \text{GF}(q) \) is 0 and zero otherwise for check nodes. The optimal node operation without output constrain therefore converts the rank-valued messages \( R_i \) back to a-posteriori probability distributions \( P_{X_i|R_i}(.|r_i) \), then implements the normal sum-product algorithm operation based on these probability-valued messages.

A picture emerges of a message passing decoder where messages are optimally converted between the symbol ranking domain and the extrinsic probability domain. This is illustrated in Figure 1. We will now make two working assumptions:

- the optimal quantization from extrinsic probability distribution to symbol ranking is the one that assigns symbols their ranks in order of decreasing extrinsic probability, i.e., retains the ranks of the distribution while discarding the probability values;
- under the independence assumption (which holds for tree graphs or for asymptotically long blocks with random interleaving), the algorithm that optimally converts between ranks and probabilities and performs optimal operations in the probability domain is optimal in the sense of achieving the best possible error probability after decoding under the message constraints.

While both assumptions appear intuitively plausible, we are unable at this point to prove or disprove them. Indeed the validity of the first assumption depends on what we mean by “optimality” of quantization in this context. If both assumptions are true, then the performance of this algorithm is a bound for any algorithm that operates on messages in the symbol ranking domain. Should one or both assumptions be wrong, then the performance is only an indication of what can be achieved and it is possible that alternative node operations could improve the performance.

We have now clarified how we plan to implement a message-passing decoder with rank-valued messages:

1) convert rank-valued input messages \( R_i = r_i \) to a-posteriori distributions \( P_{X_i|R_i}(.|r_i) \);
2) apply the equivalent sum-product operation to these probability-valued messages;
3) convert the resulting probability-valued message back to a rank-valued message by retaining the symbol-ranking of the probability distribution and discarding the values of the probabilities.

The question that remains to be solved is how to implement the first step in an efficient manner in order to assess the achievable performance. Note that a symbol ranking can take up to \( q! \) possible values, so accumulating statistics for, e.g., \( q = 64 \), is not feasible. The next section will establish a few simple theoretical results that we will use to evaluate these statistics efficiently and to implement our hypothetical algorithm.

### III. M, R Decomposition

Consider the setup in Figure 2. A \( q \)-ary random variable \( X \) is transmitted over a discrete memoryless channel. The output \( Y \) of the channel is given to a Bayesian decoder that computes the a-posteriori probability distribution \( P_{X|Y}(.|y) \) for the received value \( y \). This distribution is divided into two variables:

- \( M \) is a permutation of the vector \( P_{X|Y}(.|y) \), ordered in order of decreasing probabilities, i.e., \( M_1 \geq M_2 \geq \ldots \geq M_q \). \( M \) is a function of \( Y \).
- \( R \) is a permutation of \( \{0, 1, \ldots , q - 1\} \) such that \( P_{X|Y}(R_1|y) \geq P_{X|Y}(R_2|y) \geq \ldots \geq P_{X|Y}(R_q|y) \) if there is more than one permutation that fulfills these inequalities, then a permutation will be chosen uniformly...
at random among the possible permutations. \( R \) is thus not
strictly a function of \( Y \).

We will now state a number of simple lemmas omitting the proofs that will be used as a basis to implement our rank only decoder.

**Lemma 1:**
\[
I(X;Y) = I(X;MR)
\]  
(1)

In other words, \( M \) and \( R \) contain all the information about \( X \) present in \( Y \). It shows that, although the relation between \( Y \) and \((M, R)\) is not, strictly speaking, a bijective function, it does behave like one with respect to mutual information.

When the channel fulfills certain symmetry conditions, we will be able to state some additional results. We include the following standard information theoretic definitions:

**Definition 1:** A channel is said to be uniformly dispersive if the probabilities of the \( J \) transitions leaving an input symbol have, when put in decreasing order, the same values \( p_1 \geq p_2 \geq \ldots p_J \) for each of the \( q \) input letters.

**Definition 2:** A channel is said to be uniformly focusing if the probabilities of the \( K \) transitions to an output letter have, when put in decreasing order, the same values for each of the \( J \) output letters.

Although the last two definitions are for discrete channels, they can be extended to channels with continuous output alphabets.

**Definition 3:** A uniformly dispersive and focusing channel is said to be strongly symmetric.

**Definition 4:** A channel with input \( X \) and output \( Y \) is said to be weakly symmetric if its output alphabet can be partitioned into \( L \) subsets and a \( L \)-ary random variable \( Z \) independent of \( X \) can be constructed such that \( P_{Y|X,Z=z} \) defines a strongly symmetric channel for every \( z \).

We are now ready to state the relevant lemmas linking channel symmetry with \( M, R \) decomposition:

**Lemma 2:** For a weakly symmetric channel, \( X \) and \( M \) are independent, i.e.,
\[
I(X;M) = 0.
\]  
(2)

This lemma is intuitively pleasing because it bolsters the case for the \( M, R \) decomposition to mirror the \([L],\text{sign}(L)\) decomposition used in the analysis and simplification of binary decoders, where \( L \) is the log-likelihood ratio. For the latter decomposition, it is well known that \( I(X;|L|) = 0 \) for binary input symmetric memoryless channels [5].

**Lemma 3:** For a weakly symmetric channel, \( M \) and \( R \) are independent, i.e.,
\[
I(M;R) = 0.
\]  
(3)

We will use this result in the next section for to estimate the a-posteriori probability distribution given a rank-valued message.

**IV. ROLE MODEL FRAMEWORK AND MONTE CARLO INTEGRATION**

The problem we wish to address is to estimate the a-posteriori probability \( P(x|r) \) for a given rank-valued message \( r \). Since the number of possible symbol rankings is \( q! \), simply transmitting known symbols \( x_1, x_2, \ldots \) in a simulation and recording the joint frequencies of \( X \) and \( R \) is not feasible in practice. In principle, it is possible to determine the joint statistics of \( X \) and \( R \) analytically or numerically based on the known joint statistics of \( X \) and \( Y \). However, this would require \( q! \) multi-dimensional integrations over subregions of the definition space for the observations \( Y \), which is clearly not practical.

In [3], we introduced the “role model framework” to build an estimator based on inferior observations (in our case \( R \)) by learning from a role model estimator with superior observations (in our case \( Y \)). The main result is summarized in the following theorem:

**Theorem 1 (The “role model” theorem):** If \( X, Y \) and \( Z \) form a Markov chain \( X - Y - Z \), then
\[
E_{P_{X|Y,Z}} [D(P_{X|Y}|Q_{X|Z})] = H(X|Z) - H(X|Y) + E_{P_{X|Z}} [D(P_{X|Z}|Q_{X|Z})]
\]  
(4)

where \( Q_{X|Z} \) denotes a possibly mismatched statistical estimator for \( X \) based on \( Z \). In particular,
\[
E_{P_{X|Y,Z}} [D(P_{X|Y}|Q_{X|Z})] \geq H(X|Z) - H(X|Y)
\]  
(5)

with equality if and only if \( Q_{X|Z} = P_{X|Z} \) for all \( z \) for which \( P(z) > 0 \). In other words, the expression is minimized if and only if the estimator is optimal in a Bayesian sense.

The theorem gives a recipe for adapting an estimator based on \( Z \) to a Bayesian estimator based on \( Y \):

- at every time instant, measure the information divergence between the a-posteriori distributions computed by the Bayesian estimator and by the mismatched estimator;
- compute the expected divergence via time-averaging;
- adapt the parameters of the mismatched estimator so as to converge towards the minimum expected divergence.

While this approach can incorporate any parametric constraint on the mismatched estimator, in the non-parametric case where the mismatched estimator is allowed to be any discrete probability distribution on \( X \) for any value \( Z = z \), it is observed in [4] that the method coincides with Monte Carlo integration of the expression
\[
P(z|x) = \sum_y P(xy|z)
\]  
(6)

\[
= \sum_y P(y|z)P(x|y)
\]  
(7)

where the expression is evaluated as time average of \( P(x|y) \) to compensate for the unknown \( P(y|z) \).

The role model or Monte Carlo approach is particularly useful when the alphabet size for \( X \) and \( Z \) is manageable, and allows one to circumvent the difficulty of performing analytical or numerical integrations over the definition space of \( Y \). In the case of interest to us, however, \( Z \) is defined over an alphabet of size \( q! \), which is far from manageable for most alphabet sizes \( q \) of interest, e.g., \( q = 64 \). This is where the
symmetry properties introduced in the previous section can help. We can write
\[ P(x|r) = \sum_m P(xm|r) \]
\[ = \sum_m P(m|r)P(x|mr) \]
\[ = \sum_m P(m)\pi_r(m) \]
\[ = \pi_r[E[M]] \]
where we have used Lemma 3 in the last step, and \( \pi_r(.) \) denotes the permutation specified by the symbol ranking \( r \). Therefore, we see that, for weakly symmetric channels, \( P(x|r) \) can be determined simply by permuting the vector \( E[M] \) according to the ranking specified by \( r \). In conclusion, to reconstruct the optimal a-posteriori distribution given the rank-valued messages supposing a weakly symmetric message distribution, all we need to do is to estimate the average ranked probability distribution of the message prior to dropping the probabilities. This average ranked distribution \( E[M] \) can either be pre-computed offline, or transmitted as one extra probability-valued message that applies to all rank-valued messages in the graph. Although we have no formal proof of this at this stage, if the communication channel is weakly symmetric and the code is linear, it seems reasonable to assume that the message distribution will fulfill the weak symmetry condition throughout the iterations.

V. PERFORMANCE MEASUREMENTS

Figure 3 shows the performance of a rank-only decoder for an LDPC code of rate 1/2, length 142, defined over GF(64), transmitted over 6 binary input additive white Gaussian noise (AWGN) channels. For comparison, the performance of the classical sum-product decoder and of the extended min-sum (EMS) decoders from [2] with parameters (13,3) (bottom curve) and (7,2) (top curve) are also plotted. Note that the complexity per iteration of the EMS (13,3) is almost as high as that of the sum-product decoder and it is only the EMS (7,2) that achieves a considerable reduction of complexity per iteration. The rank-only decoder starts off better than both EMS at low signal-to-noise ratios \( E_b/N_0 \), but soon exhibits an error floor that degrades its performance to a point where it is overtaken by the EMS (13,3) and even by the much simpler EMS (7,2).

In order to investigate the reasons for this disappointing performance, we plotted the EXIT chart of the rank-only decoder, represented in Figure 4. The EXIT chart assumes a message distribution as for the output of a 64QAM channel both for variable and check node input messages. As this is clearly a very rough approximation, this EXIT chart would be unsuitable for computing a threshold or for code design. It may however be sufficiently accurate to give at least an indication of why we are observing such a bad error floor. The communication channel for the variable nodes is assumed to be 64QAM with 8.5 dB. For check nodes, the graph shows a loss of extrinsic mutual information with respect to the sum-product check node, which is to be expected and would result in a worse threshold but not give rise to an error floor. For the variable nodes however, the graph shows a slight loss over the whole range, but the key observation here is that the variable node curve fails to rise to the (6,6) point. This explains the error floor, because no matter how much a-priori information is supplied by the check nodes, the variable node is incapable of achieving a mutual information of 6 and therefore incapable of essentially determining the code digits with an error probability of 0. This means that there will always be a residual error probability even at high \( E_b/N_0 \).

As a first attempt at improving the performance, we tried to relax the rank-only requirement to allow one “soft” value to qualify the steepness of \( M \) in addition to the rank-valued message. We tried various measures of steepness, including the entropy \( -\sum m_i \log m_i \), the difference between the largest and smallest probabilities \( m_1 - m_q \), the difference of the logs \( \log m_1/m_q \) and many others. All single soft values gave the same result: the variable node curve is raised by the soft value,
but the failure to climb to the (6,6) point remains. Figure 5 shows the measurement where the soft value is $m_1 - m_q$, whose performance was almost indistinguishable from when the soft value is the entropy $-\sum m_i \log m_i$. Note that the apparent violation of the data processing theorem stemming from the larger mutual information for the rank + soft value decoder than for the optimal sum-product variable node is misleading, because the a-priori messages are not the same. If you take an a-priori probability-valued message and convert it to a rank-only message, you get a much lower a-priori mutual information. Therefore, the curve for the rank + soft value decoder has a warped x-axis with respect to the sum-product decoder, allowing it to appear to provide a better extrinsic mutual information for, e.g., $I_{av} = 3$ bits.

Our second attempt at improving the performance of our rank-based algorithm is to construct a hybrid algorithm that switches back to the sum-product algorithm for its last iterations. We use the rank-only decoder to progress along the x-axis of the variable nodes in the EXIT chart until we reach an information threshold $I_\theta$. Once we reach this threshold, we switch to the sum-product algorithm. Note that it is not obvious that this method would work as the sum-product algorithm may be unable to recover from the message degradation resulting from the rank-only processing in the previous iterations. Nevertheless, the performance of the hybrid decoder is very encouraging and gets better the lower we choose the threshold $I_\theta$. Its bit error performance is plotted in Figure 6.

Of course this promising performance of the hybrid decoder would be misleading if the decoder performed only a few iterations as a rank-based decoder and achieved its good performance by appending a number of probability-based iterations comparable to the original sum-product decoder. Figure 7 shows that this is not the case: for example, for $I_\theta = 5.5\text{dB}$, only 2-3 iterations are performed in the probability domain, and the increase in overall iterations is less pronounced than for the EMS decoder.

VI. CONCLUSION

We have studied the hypothetical performance of a decoder operating on messages constrained to symbol rankings. We have shown how to build this decoder optimally under certain assumptions. While the performance measured for a practical code is disappointing, we have shown that a modified decoder that switches to the sum-product decoder for its last iterations exhibits a performance at par with the best known reduced complexity decoders. This remains a theoretical exercise at this stage, because the decoder presented operates in the probability domain within nodes and therefore does not achieve a complexity reduction if implemented as proposed. We believe that we have made the case for further research around this type of decoders, in particular with respect to simple implementations of rank-to-rank node operations and in finding alternative methods for overcoming the weakness of the optimal rank-based variable node operation in the last iterations or at high $E_b/N_0$.

REFERENCES