EFFICIENT LDPC CODES FOR JOINT SOURCE-CHANNEL CODING

Haggai Kfir, Ido Kanter

Minerva Center and Department of Physics,
Bar-Ilan University Ramat-Gan, 52900 Israel

ABSTRACT

Belief Propagation (BP) decoding of LDPC codes is extended to the case of Joint Source-Channel coding. The uncompressed source is treated as a Markov process, characterized by a transition matrix, T, which is utilized as side information for the Joint scheme. The method is based on the ability to calculate a prior for each decoded symbol *separately*, and re-estimate this prior dynamically after every iteration of the BP decoder. We demonstrate the implementation of this method using MacKay and Neel's LDPC algorithm over $GF(q)$, and present simulation results indicating that the proposed scheme is competitive with separate scheme, even when advanced compression algorithms (such as AC, PPM) are used. The extension to 2D (and higher) arrays of symbols is straight-forward. Finally, the ability of using the proposed scheme with the lack of side information is briefly sketched.

1. INTRODUCTION

Shannon separation theorem [1, 2], states that source coding and channel coding can be performed separately and sequentially, while maintaining optimality. However, this is true only in the case of asymptotically long blocks of data. Thus, considerable interest has developed in various schemes of joint source-channel coding, where the inherent redundancy of the source is utilized for error correction, possibly with the aid of some side information (see, for instance, [3]). Combining the two processes may be motivated by reducing total complexity of the procedure, by some gain in the overall performance, and improving the resilience to single bit errors, which may corrupt entire blocks in the case of compressed data.

Shannon's lower bound for the channel capacity of a binary symmetric channel (BSC) with flip probability f, bit error rate p_b and source entropy $H(src)$ per bit is given by [1]:

$$C = \frac{1 - H_2(f)}{H(src) - H_2(p_b)}, \tag{1}$$

where $H_2(x) = -x log_2(x) - (1-x) log_2(1-x)$, is the entropy of x, and C is the maximal *Rate* between the source length k and the transmitted length m.

In this paper we propose an extension of the Low-Density-Parity-Check codes (LDPC) [4], primarily designed for i.i.d. sequences, to the case of uncompressed data. Our approach is to regard the sequence as driven from some memoryless stationary Markov process with a finite alphabet $s_n \in \{0, 1, 2...q-1\}$, and transition matrix T of dimensions $q \times q$ that describes the probability of transition from symbol i to symbol j: $t_{ij} = P(s_{n+1} = j \mid s_n = i) \equiv P(j \mid i)$. According to this point of view, neighboring symbols are *correlated*, and any information about symbols s_{n-1} and s_{n+1}, immediately implies some knowledge about s_n, too. The Markov Entropy (per symbol) of such a process is given by:

$$H = -\sum_{i=0}^{q-1} P(i) \sum_{j=0}^{q-1} P(j \mid i) log_2[P(j \mid i)] \tag{2}$$

where $P(i)$ is the stationary solution of the Markov process.

2. MN ALGORITHM

Our joint source-channel scheme is based on MacKay and Neel's algorithm (a thorough introduction may be found in [5]), a variant of the earlier Gallager code [4]. Although originally proposed for the binary field, extending the MN algorithm to higher finite fields is straight-forward as demonstrated in [6]. The original motivation for moving to higher fields was reducing the number of edges (and short loops) in the code's graph. However, for our purpose, this enables us to regard the source sequence as a Markov process with a richer alphabet, of $q = 2^i$ symbols, (i being an integer) [11]. The algorithm consists of two sparse matrices known both to the sender and the receiver: $A(m \times k)$, and $B(m \times m)$, where k is the source block length and m is the transmitted block length, the code rate being $R = k/m$. All non-zero elements in A and B are from $\{1, 2...q-1\}$, and B must be invertible. Encoding of a source vector s into a codeword t is performed (over $GF(q)$) by:

$$t = B^{-1} \cdot A \cdot s. \tag{3}$$

During transmission, noise n is added to t, therefore the received vector is $r = t + n$. Upon receipt, the decoder computes $z = B \cdot r$, and faces the following syndrome:

$$z = B{\cdot}r = B{\cdot}(t{+}n) = B{\cdot}(B^{-1}{\cdot}A{\cdot}s{+}n) = [AB]{\cdot}x, \quad (4)$$

where square brackets denote concatenation of matrices, and x is a concatenation of s and n. The decoding problem may be solved using the Belief Propagation (BP) (or sum - product) algorithm. BP is an iterative algorithm with two alternating steps, horizontal pass (check→ variable messages) and vertical pass (variable → check messages) [5, 6]. During the vertical pass, some prior knowledge is assigned to each decoded symbol, according to the assumed statistics (for the i.i.d. case this would simply be: $Pr(s_i = k) = 1/q$ for all the symbols). The key point here is that one can re-estimate and re-assign these priors after every iteration *individually for each decoded symbol* [7]. The outcome of each iteration is an a-posteriori probability $Q_i^a = Pr(x_i = a)$, for each symbol (both source and noise).

A proper construction of the matrices A and B is crucial in order to ensure capacity-achieving performance. In this work we follow Kanter and Saad (KS) constructions, described in [8]. The decoder is linear, with complexity $O(kqu)$ (per iteration), where u is the average number of checks per symbol [9, 10].

3. DERIVING THE DYNAMICAL BLOCK PRIORS

Consider three successive symbols s_{i-1}, s_i, s_{i+1} in a sequence generated by a Markov process with transition matrix T, and alphabet $GF(q)$. The probability of the triplet a, b, c is given by [11]:

$$P(a,b,c) = P(a|b,c) \cdot P(b,c) = P(a|b) \cdot P(b,c) = \\ = P(a|b) \cdot P(b|c) \cdot P(c) = \frac{P(a,b) \cdot P(b,c)}{P(b)}, \quad (5)$$

where use has been made of the fact that the process is memoryless, and of the Bayse rule: $P(x,y) = P(x|y) \cdot P(y)$. Now, given the a-posteriori probabilities for the first and last symbols in the triplet: $Q_{i-1}^a = Pr(s_{i-1} = a)$ and $Q_{i+1}^c = Pr(s_{i+1} = c)$, one can calculate a prior for the probability that $s_i = b$ [11]:

$$Pr(s_i = b) = \sum_{a,c=0}^{q-1} P(a,b,c) \cdot P(a) \cdot P(c) = \\ = \frac{\sum_{a=0}^{q-1} (P(a,b) \cdot Q_{i-1}^a) \cdot \sum_{c=0}^{q-1} (P(b,c) \cdot Q_{i+1}^c)}{P(b)}. \quad (6)$$

The extension of the MN algorithm to the joint source-channel case consists of the following steps:

1. A binary sequence of $k \cdot log_2(q)$ bits is converted to k $GF(q)$ symbols.

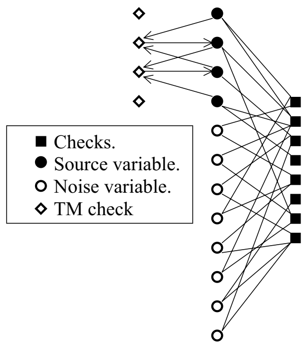

Figure 1: The Dynamic Block Prior visualized as a third layer attached to the bipartite graph of the LDPC code.

2. The encoder measures $P(a,b)$ and $P(a)$ for all the q symbols over the source, and transmits reliably this side information to the decoder.

3. The source is encoded according to (3), then reconverted to binary representation and transmitted over the BSC.

4. The decoder maps the received signal back to GF(q), and performs the regular decoding (4), but after every iteration of the BP, the prior for each source symbol is recalculated according to (6), and normalized such that: $\sum_{a=0}^{q-1} Pr(s_i = a) = 1$.

The complexity of calculating the q priors for a single symbol according to the posteriors of its neighbors is reduced from q^3 to q^2 by eq. (6), hence the decoder's complexity remains linear, with total complexity of $O(kqu + kq^2)$ per iteration. The above-mentioned procedure may be thought of as adding a layer to the bipartite random graph represented by the matrix $[AB]$: while the original BP messages are exchanged between check and variable nodes (z and x elements respectively), eq. (6) represents messages passed among *source* elements of x only, which are spatially related. In Fig. 1, the diamonds represent this new (directional) layer, which connects neighboring source nodes (filled circles). We note that the possibility of extending this scheme to Gallager codes is an open question, since the source is not explicitly represented in the graph.

4. SIMULATION RESULTS

We report here results for rate $R = 1/3$, using the corresponding KS for A and B devised in [8]. Other rates, constructions and block length were also checked. Random vectors of length $L = 10^4$ bits were generated by the Markov process, then mapped to a vector in $GF(q)$ with length $k = L/log_2[q]$, and were encoded and decoded as

q	k	H	$f_{Sh.}$	f_c	f_∞
4	5000	0.49	0.266	0.215	0.244
8	3333	0.471	0.271	0.223	0.243
16	2500	0.49	0.266	0.21	0.236

Table 1: some selected results of simulations

q	%AC	R_{AC}	f_{AC}	%PPM	R_{PPM}	f_{PPM}
4	58.4%	0.195	0.247	56.6%	0.189	0.25
8	58.1%	0.194	0.248	55.5%	0.185	0.253
16	60.5%	0.201	0.243	59.4%	0.198	0.245

Table 2: critical noise level for separate scheme using Arithmetic Coder and Prediction by Partial Match compression algorithms

described in the previous section. For each reported result, at least 1000 sample vectors were generated and transmitted. Some selected results are presented in Table 1. The columns correspond to: the field size q; the source length k; the entropy (per bit) of the source H [13]; and the corresponding maximal noise $f_{Sh.}$ (eq. (1)); the critical noise, up to which the bit error rate $P_b \leq 10^{-5}$; and the critical noise for infinitely large blocks ($k \to \infty$) calculated using the scaling property for the convergence time t (in iterations): $t \propto \frac{1}{f_\infty - f(t)}$ (see the inset of Fig. 2, and [11] sec. 3, for a detailed explanation).

In order to compare the joint and the separate schemes, the generated samples were concatenated to strings of size $L = 10^5 - 10^6$ bits, and compressed using two advanced compression algorithms: Prediction by Partial Match (PPM) [14], and Arithmetic Coder (AC) [15]. In Table 2, we preset the compression ratio of the source vectors using each method, the new rate after compression, R_{AC}, R_{PPM} (for transmission of blocks of size m as in the joint scheme) and the maximal noise level for the new rate for i.i.d. source, f_{AC}, f_{PPM}. Since we assume an *optimal* decoder, these noise levels should be compared to f_∞ for the joint scheme. In all cases, the threshold of the proposed joint scheme for $k \to \infty$ is comparable with the separate scheme. One should recall that the construction of A and B can be optimized for improved results.

In Fig. 2, P_b is plotted against the noise level of the channel, f; for the examples of Table 1, filled triangles represent $q = 8$; filled squares represent $q = 16$; the empty symbols in this figure refer to an approximation that will be described in the following section. The dashed (full) arrow points to $f_{Sh.}$ for $q = 8(16)$. The inset of Fig. 2 demonstrates the extrapolation of f_∞ from the convergence time: for $q = 8$, f is plotted against $1/t_{med}$, the median of the convergence time for all the samples. f_∞ is then recovered by fitting the points to the equation (dashed line):

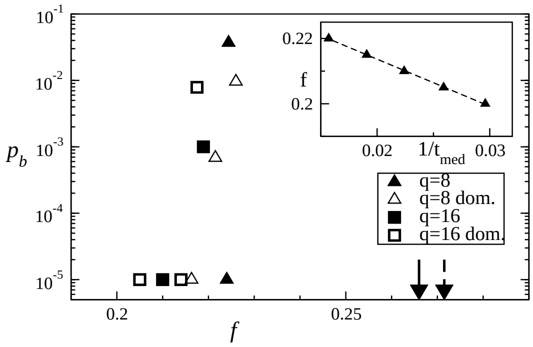

Figure 2: Bit error rate p_b vs. noise level f. triangles for $q = 8$, squares for $q = 16$, empty symbols: use of only q dominant elements in T, see sec. V. Inset: scaling behavior for median convergence time t_{med}

$f = -A \cdot 1/t_{med} + f_\infty$ [8].

5. REDUCING THE AMOUNT OF SIDE INFORMATION

The main drawback of our scheme is the request for side information, or a header, for each transmitted block, namely the transition matrix T of q^2 elements. Even for moderate sized symbols, such as 1 byte ($q = 256$) the header becomes very large relative to the transmitted block size (say 10^4 bits). This point may be tackled by observing that for a process with low entropy, characterized by enhanced repetitions and correlations, T is dominated by a small number of elements, while the rest of the elements are negligible. We therefore repeated our simulations, using only the q largest elements in T as side information. The decoder would then set all other elements in each row of T equally, to obey the normalization condition $\sum_i T_{ij} = 1$. In Fig. 2 the empty squares/triangles represent working points for the algorithm with $q = 8/16$. In both cases, the critical noise level f_c is only slightly decreased, but the size of the side information becomes considerably smaller. Treating T as a sparse matrix also reduces the complexity of calculating the Dynamical Block Priors (eq. (6) [11]).

6. EXTENDING TO 2D (AND HIGHER) ARRAYS

The proposed scheme is not limited to 1D sequences. We shell demonstrate this point by considering a 2D array of size $k \times k$ of $GF(q)$ symbols. Each symbol depends only on its nearest neighbors. The probability of finding the quartet a, c, d, e (left and right, above and below, respectively) surrounding the symbol b is (following the Markovian assumption, (5)):

$$P(a, c, d, e) \quad = \quad P(a|b) \cdot P(c|b) \cdot P(d|b) \cdot P(e|b) \cdot P(b) =$$

$$= \frac{P(a,b)P(c,b)P(d,b)P(e,b)}{P(b)^3} \quad (7)$$

Having the posterior probabilities for all the neighbors of a given symbol $s_{i,j}$, the dynamic block prior can be calculated by:

$$Pr(s_{i,j} = b) = \frac{\prod_{n=1}^{4}[\sum_{a=0}^{q-1}(P(a,b) \cdot Q_n^a)]}{P(b)^3}, \quad (8)$$

where the index n runs over the nearest neighbors. The complexity of calculating all q dynamical priors for a single symbol with N neighbors is $O(Nq^2)$, and reducing the size of the side information, as described in the previous section, may also be applied to the 2D case.

7. JOINT SOURCE-CHANNEL CODING WITH THE LACK OF SIDE INFORMATION

As a concluding remark, we would like to mention the possibility of use of this scheme with no side information. The key points are the special properties of the KS construction [8]: the first k rows of A are characterized by one non-zero element per row and column, where the first k rows of B are characterized by 2 non-zero elements. Furthermore, due to the systematic form of B, each row cannot be written as a linear combination of the other rows. Hence, the first k bits of the syndrome vector z, are equal (up to a simple permutation) to the source, with an effective flip rate, f_{eff}. For $GF(2)$ for instance, $z_j = s_i + n_j + n_{j+1}$ (i marks the position of the nonzero element in the j^{th} row of A), and $f_{eff} = 2f(1-f)$. The first k symbols of z are therefore a result of a *hidden* Markov Model (HMM). The underlaying transition matrix, T, generating the source sequence, (as well as f_{eff}) can be estimated by means of the EM algorithm [16] , which is a standard tool for solving such *Parametric Estimation* problems, with linear complexity. Having T (approximately) revealed, the DBP's can be calculated as described in Eq. (6).

For the general construction of the MN algorithm one adds/subtracts rows of the concatenated matrix $[AB]$ and the corresponding symbols in z, such that a situation is finally reached as follows: The first k rows of A are the identity matrix, regardless of the construction of the first k rows of B. From the knowledge of the noise level f and the structure of i^{th} row of B one can now calculate the effective noise, $f_{i,eff}$, of the i^{th} received source symbol. Since all $\{f_{i,eff}\}$ are functions of a unique noise level f, one can estimate the parameters of the Markovian process using some variants of the EM algorithm. Note, that in the general case the first k rows of B contain loops, hence $\{f_{i,eff}\}$, are correlated. However, these correlations are assumed to be small as the typical loop size is of $O(\log(k))$[17].

8. REFERENCES

[1] C.E. Shannon , "A mathematical theory of communication", *Bell System Technical J.*, **27**, 379-423, 623-656, 1948.

[2] T.M. Cover, J.A. Thomas, *"Elements of information theory"*, Wiley. 1991, UK.

[3] J. Garcia-Frias, Y. Zhao, "Compression of binary memoryless sources using punctured turbo codes", *IEEE Communication Letters, vol.6, no.9, pp. 394-396, 2002*

[4] R.G. Gallager, *Low density parity check codes*, Research monograph series **21** (MIT press), 1963.

[5] D.J.C. MacKay, "Good error-correcting codes based on very sparse matrices", *IEEE Transactions on Information Theory, vol.45, no.2, March 1999, pp.399-431.*

[6] M. C. Davey, D. J. C. MacKay, "Low-density parity check codes over GF(q)", *Communications Letters, vol.2, no.6, June 1998, pp.165-7*.

[7] I. Kanter, H. Rosemarin, "communication near the channel capacity with the abseance of compression: statistical mecanical approach", cond-mat-0301005.

[8] I. Kanter, D. Saad, "Error-correcting codes that nearly saturate Shannon's bound", *Physical Review Letters, vol.83, no.13, 27 Sept. 1999, pp.2660-3.*

[9] D. J. C. MacKay, M. C. Davey, "Gallager Codes for Short Block Length and High Rate Applications", Codes, Systems and Graphical Models, IMA Volumes in Mathematics and its Applications, Springer-Verlag (2000).

[10] T. Richardson, R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding", *IEEE Transactions on Information Theory*, **47**, 599 (2001).

[11] I. Kanter, Kfir H, "An efficient joint source-channel coding for a D-dimensional array" *cond-mat/0308308* (submitted to IEEE Trans-IT).

[12] I. Kanter, H. Kfir, "Statistical Mechanical Approach to Joint source-channel coding." *Europhys. lett.* ***63****, 310 (2003)*

[13] For the sake of brevity we report T only for q=4:

$$T = \begin{pmatrix} 0.85 & 0.07 & 0.04 & 0.04 \\ 0.23 & 0.25 & 0.05 & 0.47 \\ 0.47 & 0.05 & 0.25 & 0.23 \\ 0.04 & 0.04 & 0.07 & 0.85 \end{pmatrix} \quad (9)$$

[14] PPMZ downloaded from www.cbloom.com/src/ppmz.html

[15] AC downloaded from www.cs.mu.oz.au/ alistair/arith_coder

[16] McLachlan GJ and Krishan T, The EM Algorithm and Extension. *Wiley Sons, New York, 1997*.

[17] P. Erdos and A. Renyi, "On the evolution of random graphs", reprinted in J. Spencer (ed) "The art of counting: selected writings", (MIT Press, Cambridge) 1973.