Joint Source-Channel Turbo Decoding of Entropy Coded Sources

Karim Ali Fabrice Labeau

Center for Advanced Systems and Technologies in Communications (SYTACom)
Department of Electrical Engineering
McGill University Montreal, Quebec H3A 2A7
Email:{karim, flabeau}@tsp.ece.mcgill.ca

Abstract

A new turbo joint source-channel decoding algorithm is presented. The proposed scheme, derived from a Bayesian network representation of the coding chain, incorporates three types of information: the source memory; the residual redundancy of the source coder; and finally the redundancy introduced by the channel coder. Specifically, we modify an existing algorithm by introducing an equivalent graph, that is shown to hold the same state-space while exhibiting far less undirected cycles. A fully consistent solution for joint turbo decoding within the Bayesian networks framework follows. The proposed algorithm is demonstrated to yield considerably better results along with a drastic reduction in computational complexity when compared to the existing one.

1 Introduction

Communication systems that employ sequential decoding, namely channel decoding followed by source decoding, ignore two types of information: the residual redundancy of the source coder and the source memory (inter-symbol correlation). If a block length 1 optimal source coder is used — a common component — as much as 1 bit of residual redundancy per symbol may remain in the data. In addition, such a compression scheme leaves all source memory intact. These two types of redundancies, present at the output of the source coder, are also necessarily present in the received data stream. One can therefore consider designing a joint decoder that would incorporate the two former sources of *natural* redundancy along with that *artificial* redundancy introduced by the channel coder; a possibility mentioned as early as Shannon's seminal paper [1]. Such a design strategy is motivated further by the fact that optimal source coders of the variable length code (VLC) variety have corresponding source decoders that are extremely sensitive to noise: the lack of set symbol boundaries resulting in a vulnerability to synchronization errors. Joint decoders have been shown to reduce the effects of de-synchronization and generally improve the overall decoding performance [2]-[6].

The authors in [3] developed a generic solution to the joint decoding problem by deriving the product finite state machine (FSM) model of the source, the source coder and the channel coder. Various algorithms such as Hard Viterbi, Soft Viterbi and BCJR (Kalman smoothing) are then readily applicable yielding the *optimal* solution with respect to the algorithms' criteria. Unfortunately, this solution has intractable complexity: the cardinality of the state-space of the product model is equal to the product of the cardinalities of the three constituent models. This unaffordable phenomenon, leads to the need for less complex and therefore sub-optimal joint decoders. In this context, the authors in [4]-[5] provided a sub-optimal joint decoding solution under the additional assumption of a memoryless source. Specifically, their proposed algorithm uses the principle of turbo-decoding and alternates the use of a soft VLC decoder with a soft channel decoder. This approach was recently extended by Guyader *et al.* in [6] to include sources with memory. The algorithm, which also relies on the principles of turbo-decoding and was derived in the context of Bayesian networks, has the advantage of isolating the three constituent components and therefore has limited complexity.

In this paper, we present an algorithm largely inspired from [6]. In particular we consider an equivalent Bayesian network representation of the joint decoding problem. The resulting graph has the same state-space, contains fewer loops (undirected cycles) and is shown to yield considerably better result with a significant reduction in computational complexity. We begin this paper with a brief review of the algorithm as presented in [6]. Next, the proposed algorithm is introduced and studied. Finally, experimental results are shown.

2 Turbo Joint Decoding via the Bayesian Networks Framework

A good introduction to Bayesian networks may be found in [7]. Essentially Bayesian networks provide a graphical representation of statistical problems based on the *factoring* of their joint distribution into conditional distributions. The resulting graph may then be used to incorporate new knowledge as particular nodes (random variables) are instantiated. Belief Propagation (BP), which we henceforth refer to, essentially performs maximum-a-posteriori (MAP) estimation of each random variable in the graph. Belief Propagation may be either *blind* (or rather locally triggered by each node) or *organized*, in two passes equivalent to the BCJR algorithm.

This research was supported in part by a grant from FQRNT

Here we briefly summarize the algorithm developed by Guyader et al. [6]. Let an order one, finite alphabet, stationary Markov source (MS) generate symbols $S = S_1 \dots S_N$. These symbols are subsequently mapped via a block length one, binary source coder[1] (SC) into a sequence of information bits $U = U_1 \dots U_K$. The source coder mapping may be represented by a binary tree τ. Determining the Bayesian network corresponding to the MS+SC pair is not a difficult exercise and essentially requires deriving the product state-space representation of the MS and SC. This can be done by defining the general state variable X as the pair[2] (Γ, V) where Γ with instance $\gamma^{(i)}$, is a variable representing the last completed symbol and V, with instance $v^{(j)}$, is a variable representing the current vertex of τ. The transitions of $(\Gamma, V)_k$ to $(\Gamma, V)_{k+1}$, yielding the information bits, are then completely determined by the source transitions probabilities $P(S_{n+1}|S_n)$ and the topology of τ.

Supposing that the information bits are now sent to a channel coder (CC), the Bayesian network corresponding to the CC is equally easy to derive. We may with no loss in generality assume that the CC takes information bits one at a time and yields a number of redundant bits, possibly none. We denote by R_k, the *set* of redundant bits generated at time k: thus for a rate $1/3$ systematic channel coder each R_k is in fact 2 bits. The Bayesian network corresponding to all three elements (MS, SC and CC) is shown in figure 1 where we have used X_k to denote $(\Gamma, V)_k$ and X'_k to denote the state variable of the CC. The constraint on symbol termination is used to ensure that the last state X_K indeed corresponds to the end of a symbol: an important constraint in the VLC case. Such a graph, as previously mentioned may be used to provide MAP estimates of all nodes, given observations on R_k's and possibly U_k's (for a systematic channel coder). Unfortunately, the graph is not singly connected due to the presence of loops, and hence estimation algorithms (BP) do not apply directly [7]. One solution to render the graph singly connected is to compound the state variables X and X' into a single variable (X, X'). However this solution is essentially equivalent to the optimal joint decoding proposed by [3] and therefore suffers from the same intractable complexity issue. It was noted instead that the introduction of an interleaver between the MS+SC and the CC increases the average length of the loops. Graphs with long cycles can locally be approximated by a singly connected graph and hence BP gives good approximations. This is shown conceptually in figure 2. In actuality the graph is separated into two subgraphs with the information bit sequence U reproduced. BP, in two passes, is then applied separately on each constituent graph with information being passed via standard extrinsic information calculations. This is shown in figure 3 where we have included the pointwise observations on U_k and R_k as well as the extrinsic information measurements. It is further possible to organize computations in the MS+SC pair in such a way as to entirely isolate the three components: MS, SC and CC. If the receiver is aware of the total number of symbols N sent, one can incorporate this information with relative ease. Simply define the a new state variable (X, C) that is to replace X. With C representing a symbol counter denoting the number of symbols currently achieved, the state transitions of $(X, C)_k$ to $(X, C)_{k+1}$ immediately follow. The constraint on symbol termination essentially then ensures that $(X, C)_K$ corresponds to the end of the N^{th} symbol so $C_K = N$ and X_K is as before. The information on the total number of symbols is extremely relevant when a VLC is used as it permits the synchronization both at the beginning of the data stream and at the end. It does however come at the cost of multiplying the cardinality of the state-space of X by N and hence, the decoding complexity is dramatically increased.

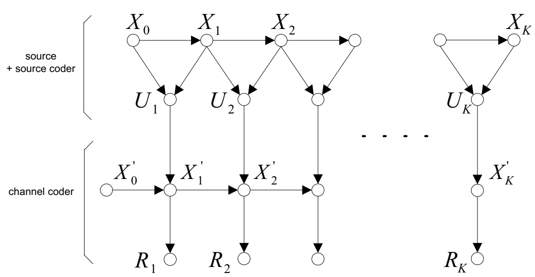


Fig. 1 Bayesian network corresponding to the MS+SC+CC.

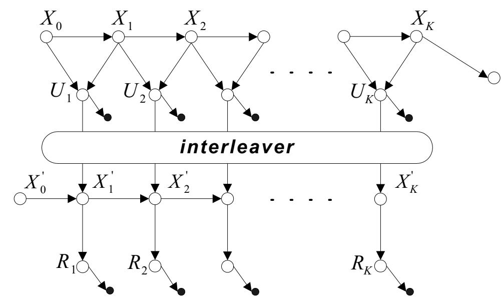
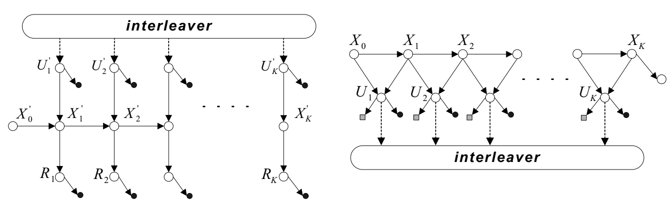


Fig. 2 Introducing an interleaver.



(a) CC subgraph. (b) MS+SC subgraph.

Fig. 3 Iterative scheme

[1] The assumptions on the order of the Markov source and the block length of the SC may be relaxed and similar networks derived.

[2] some impossible $(\gamma^{(i)}, v^{(j)})$ pairs exist namely for the case that v is a leaf node and hence $\gamma^{(i)}$ must necessarily be the corresponding symbol.

3 Proposed Algorithm

3.1 Refactoring the Graph

The proposed equivalent graph stems from different considerations when defining the Bayesian network corresponding to the Markov source and source coder pair. One can consider the information bits U_k to depend solely on the *current* state $(\Gamma, V)_k$ and not on the *transition* of states. Since the states $(\gamma^{(i)}, v^{(j)})$ lie on a tree, the two representations turn out to be equivalent. Indeed, it can be shown that the dependence on the transition of states is merely formal and may be removed. The equivalent graph is shown in figure 4. In essence, we may re-factor the joint distribution of the MS+SC from conditionals of the form $P(U_k|X_{k-1}, X_k)$ to the conditionals $P(U_k|X_k)$. Generally speaking, it is always possible to transform a state-

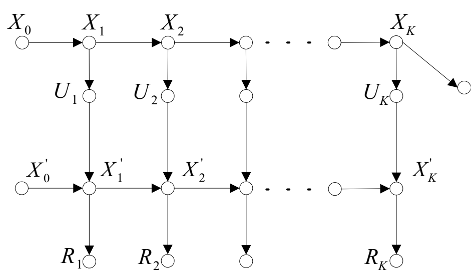


Fig. 4 Proposed equivalent graph to figure1.

space representation of a domain with outputs depending on the transitions of states into a state-space representation with outputs depending on the current state. This however generally comes at a cost. With q denoting the number of possible transitions from one state to a next and a state-space of cardinality $|S_p|$, such a transformation may result in a state-space of cardinality as large as $q|S_p|$. When the states lie on a q-ary tree, one can change from one model to the next with no cost.

3.2 Theoretical Advantages

3.2.1 Relaxing a stringent assumption

Our equivalent graph relaxes a relatively stringent assumption. Indeed Guyader *et al.*'s MS+SC graph is not singly connected. Node U_k assumes information in node X_{k-1} to be statistically independent from the information in node X_k as dictated by BP. This will generally yield the posterior marginal of U_k to be inconsistent with the posterior marginals of X_{k-1} and X_k. This problem is still present after the insertion of the interleaver and the implementation of turbo decoding. On the other hand, our equivalent MS+SC graph is a tree, implying that belief propagation algorithms will yield correct posterior distributions for U_k, consistent with all other nodes and hence, it is also expected to yield a better performance.

3.2.2 Computational complexity reduction

The complexity of the entire decoder is dictated by the cardinality $|\mathcal{X}|$ of the state-space of X and the cardinality $|\mathcal{X}'|$ of the state-space of X'. Our equivalent graph results in a significant reduction in complexity as node U_k now performs a number of operations proportional to $|\mathcal{X}|$ instead of operations proportional to $|\mathcal{X}|^2$. Hence we have that the number of multiplications performed by our algorithm as a fraction of the number of multiplications performed by [6] is,

$$\frac{2|\mathcal{X}|^2 + 10|\mathcal{X}'|^2}{10|\mathcal{X}|^2 + 10|\mathcal{X}'|^2} \tag{1}$$

whereas the fraction of additions performed by our algorithm is,

$$\frac{2|\mathcal{X}|^2 + 6|\mathcal{X}'|^2}{8|\mathcal{X}|^2 + 6|\mathcal{X}'|^2} \tag{2}$$

Figure 5 shows the percentage of both additions and multiplications with respect to $|\mathcal{X}|$ for the entire decoding scheme. It was assumed that the channel coder has 5 bits of memory and hence $|\mathcal{X}'| = 2^5$. The rate of decrease of the two curves is due

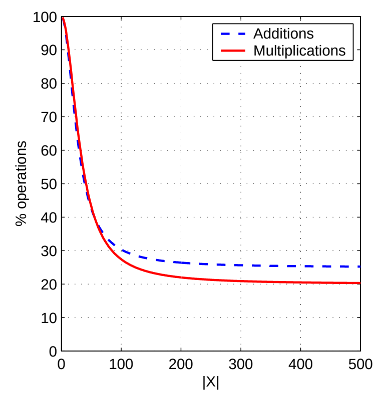


Fig. 5 Number of computations in our equivalent graph as a percentage of the number of computations in the original graph [6] versus the MS+SC state-space cardinality.

to the fact that the channel decoder complexity is becoming more and more negligible with respect to the complexity of the MS+SC. When the channel decoder complexity is negligible the proposed decoder performs 20% of the multiplication and 25% of the addition operations of the existing algorithm.

3.2.3 Reduction in number of loops

The proposed algorithm comes with the added advantage that our equivalent representation of the entire coding chain contains significantly fewer undirected cycles (loops). Assuming a bit sequence of length K, the proposed graph exhibits $\frac{1}{2}K(K-1)$ loops. This stands in sharp contrast with the $2^{K+2} - 3K - 4$ loops exhibited by Guyader *et al.* Figure 6 shows the percentage in the total number of loops present in our equivalent graph versus the graph length K (number of bits). This drastic reduction in the number of loops necessarily

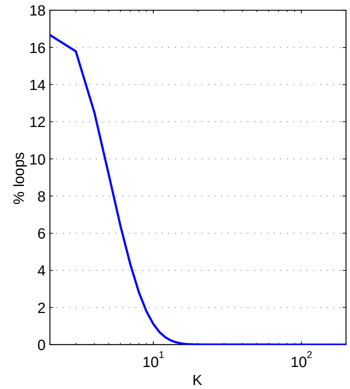


Fig. 6 Number of loops in our equivalent graph as a percentage of the number of loops in the original graph [6] for different lengths of source sequences in bits.

implies that our graph may be better locally approximated by a singly connected graph. This is in turn implies a better approximation of the posterior probabilities of each node and thus a better performance in the context of decoding.

3.3 Iterative Decoding

Our equivalent graph may be used in much the same way as the original one to achieve turbo decoding. That is, an interleaver is inserted between the MS+SC model and the CC model. However, we propose a different approach to extrinsic information calculations. Specifically, we leave the entire graph connected, yet interleaved, and perform the iterative decoding through a specific ordering of node activation. This consists in performing MAP BP in two passes on the CC component, allowing the information bits to update their posterior distributions *within* the parameters of MAP BP, and subsequently perform two passes on the MS+SC component. This closes the loop of the first iteration and the process is repeated for the subsequent iterations. The 1^{st} iteration is shown in Table 1 below. We have used y_1^k to denote the vector $y_1, \ldots, y_k$.

Table 1 Iterative scheme as a particular ordering of node activation

Node Act. (CC)	$BEL(\cdot)$	Node Act. (MS& SC)	$BEL(\cdot)$
U_1 $\vdots$ U_K	$P^0(u_k\|y_k)$	U_1 $\vdots$ U_K	$P^0(u_k\|y,z)$
X'_0 $\vdots$ X'_K	$P^0(x'_k\|y_1^k,z_1^k)$	X_0 $\vdots$ X_K	$P^0(x_k\|y_1^k,z_1^k)$
$\vdots$ X'_0	$P^0(x'_k\|y,z)$	$\vdots$ X_0	$P^0(X_k\|y,z)$

As the second iteration begins, the activation of nodes U_1 to U_K yields $P^1(u_k|y,z)$ and we continue the process. It can be shown that with the joint distribution of the MS+SC re-factored as previously described, this approach is entirely equivalent to standard extrinsic information computations. The intuition behind our scheme reduces to the fact that the messages on the link $U_k \rightarrow X'_l$ contain *disjoint* information and as such, should be used as the appropriate extrinsic information quantities. This method has the added advantage of forgoing the additional overhead required in separating the graph and computing the extrinsic information quantities externally to the BP process. It therefore also provides a fully consistent solution for joint turbo decoding within the Bayesian networks framework in agreement with [8].

4 Results

To evaluate the performance of the joint decoding scheme, experiments were performed on an order one stationary Gauss-Markov source with zero mean, unit variance and correlation coefficient 0.9. The source is quantized with an 3-bit uniform quantizer on the interval $[-3, 3]$. Blocks of $N = 200$ symbols were decoded at a time. The source coder is a Huffman code designed for the source statistics and yields an average length of 2.54 bits per source symbol. The channel coder is a rate $3/4$ recursive systematic convolutional coder with 5 bits of memory. An additive white gaussian noise (AWGN) channel was assumed along with a BPSK modulation.

Figure 7 shows the symbol error rates (SER) for different channel E_b/N_0 under the assumption that the decoder has no knowledge of the total number $N = 200$ of transmitted symbols. The top curve shows the case when Hard Huffman decoding is used directly on the received bit stream. The second curve shows the performance of sequential decoding, namely soft-input MAP channel decoding followed by hard Huffman decoding. The third shows the performance of Guyader *et al.*'s joint decoding scheme at the fourth iteration while the last curve shows the proposed procedure also at the fourth iteration.

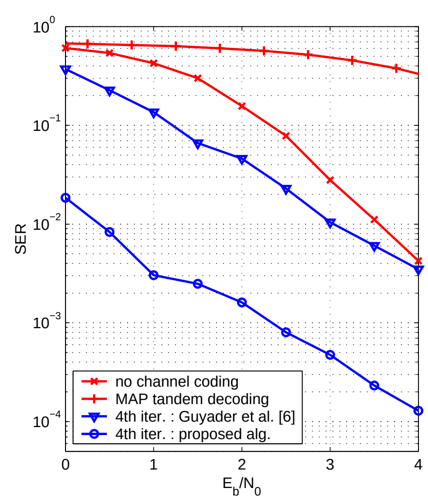


Fig. 7 SER vs. E_b/N_0 with no knowledge of N.

We note an additional gain of approximately 3dB. Clearly the proposed algorithm has much higher synchronization power in the case when N is unknown to the decoder. In the specific case where the decoder has no knowledge of N, the cardinality $|\mathcal{X}|$ is in fact a monotonic increasing function of the number of possible output symbols of the source. Figure 8 shows the percentage complexity versus the source alphabet cardinality. For the simulated 8 symbol source, the aforementioned gain of 3dB was obtained while performing 60% of the addition and 40% of the multiplication operations. A 16 symbol source already yields the asymptotic reduction in computations and similar gains are expected.

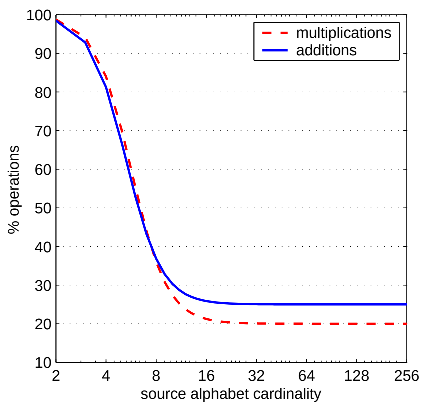


Fig. 8 Number of computations in our equivalent graph as a percentage of the number of computations in the original graph [6] versus the source alphabet cardinality.

Figure 9 shows the symbol error rates (SER) for different channel E_b/N_0 under the assumption that the decoder *has* knowledge of the total number $N = 200$ of transmitted symbols. The order of the curves is the same as in figure 7.

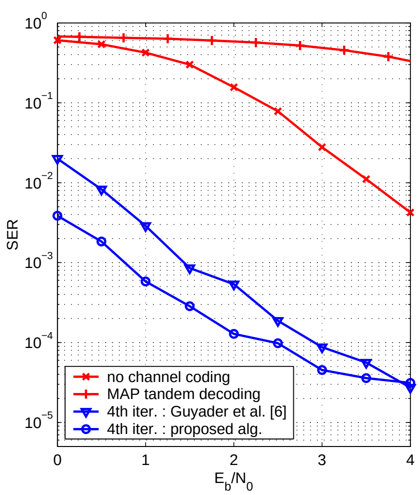


Fig. 9 SER vs. E_b/N_0 with knowledge of N.

We note a gain of approximately 1dB with the two methods converging for higher E_b/N_0. In the case where knowledge of N is incorporated, the cardinality $|\mathcal{X}|$ will depend both on the source alphabet cardinality and N as previously mentioned. With $N = 200$ the channel decoder's complexity is negligible for any number of possible source symbols and hence our algorithm always performs 50% addition and 20% multiplication operations.

5 Conclusion and Future works

An improved turbo joint decoding algorithm was presented. The proposed scheme, used for decoding of entropy coded sources, is significantly reduced in computational complexity and is shown to yield better results. The increase in performance is particularly noticeable in the case where the decoder has no knowledge of the total number of sent symbols. The synchronization power of the algorithm is great enough to consider the possibility of only using it in that case and not to incorporate the knowledge of the total number of sent symbols. Indeed the number of operations required by the decoder when knowledge of the total number of sent symbols is incorporated is several orders of magnitude greater. Hence such a scheme would perform less then 0.002% of the operations and shows a performance loss of only 1dB.

References

[1] C. E. Shannon, "A mathematical theory of communication", *Bell Systems Technical Journal*, vol. 27, pp. 379-423, July-Oct. 1948.

[2] D. J. Miller, M. Park, "A sequence-based approximate MMSE decoder for source coding over noisy channels using discrete hidden Markov models", *IEEE Journal Transactions on Communications*, vol. 46, pp. 222-231, Feb. 1998.

[3] A. H. Murad, T. E. Fuja , "Joint source-channel decoding of variable length encoded sources", *in Proc. Inform. Theory Workshop, ITW*, June 1998, pp. 94-95.

[4] R. Bauer, J. Hagenauer, "Symbol-by-symbol map decoding of variable length codes", *in Proc. 3rd ITG Conf. Source and Channel Coding*, January 2000, pp. 111-116.

[5] R. Bauer, J. Hagenauer , "Iterative source/channel decoding based on a trellis representation for variable length codes", *in Proc. Int. Symp. Information Theory, ISIT* , June 2000, p. 238.

[6] A. Guyader, E. Fabre, C. Guillemot, M. Robert, "Joint source-channel turbo decoding of entropy-coded sources", *IEEE Journal on Selected Areas in Communications*, Vol. 19, 9, Sept. 2001 pp. 1680-1696.

[7] J. Pearl,"Fusion, propagation, and structuring in belief networks", Artificial Intelligence, vol. 29, pp. 241-288, 1986.

[8] R. J. McEliece, D. J. C. MacKay, J.-F. Cheng,"Turbo decoding as an instance of Pearls belief propagation algorithm", *IEEE Journal on Selected Areas in Communincations*, vol. 16, pp. 140-152, Feb. 1998.