DISTRIBUTED JOINT SOURCE-CHANNEL ARITHMETIC CODING

Marco Grangetto

Dipartimento di Informatica Università degli Studi di Torino (Italy)

ABSTRACT

We deal with the problem of distributed source coding with decoder side information, when the decoder observes the source through a noisy channel. Existing approaches employ syndrome- or parity-based channel codes. We propose a new approach based on distributed arithmetic coding (DAC). We introduce a DAC with forbidden symbol, which allows to tune the redundancy according to the amount of channel noise. We propose a novel sequential decoder that employs the known side information to decode the corrupted codeword. Experimental results show that the proposed scheme is better than parity-based turbo codes at relatively short block lengths.

1. INTRODUCTION

In the last few years, a lot of interest has been devoted to the problem of encoding correlated information sources. In one such problem (coding with side information), the encoder generates a codeword for sequence X at rate less than its entropy, knowing that the decoder will be aided by knowledge of a correlated side information Y. The prevalent approach to this problem is to use channel codes designed for the "virtual channel" between X and Y, transmitting the syndrome or the parity bits of a channel code of suitable rate, e.g. a low-density parity-check (LDPC) or turbo code. Alternatively, distributed arithmetic coding (DAC) has also been proposed [1]. DAC uses interval overlap during the arithmetic coding (AC) process to yield and ambiguous compressed description of X; sequential decoding of X with branch metric depending on Y yields results better than turbo and LDPC codes at relatively short block lengths.

Recently, the problem of distributed joint source-channel coding has also been explored. In this setting, the codeword representing \mathbf{X} is not available exactly at the decoder, but is received through a noisy communications channel. The decoder attempts to estimate \mathbf{X} given the received sequence \mathbf{R} and the known side information \mathbf{Y} . In [2] irregular repeataccumulate codes are employed, while LDPC codes are used in [3]. In [4] the syndrome and parity approaches are compared, and the parity approach is found to be more error-resilient under noisy transmission.

In this paper, we extend the DAC approach to the scenario

Enrico Magli, Gabriella Olmo

Dipartimento di Elettronica Politecnico di Torino (Italy)

of noisy communication. The proposed encoder is a combination of DAC and arithmetic coding with forbidden symbol [5]. The decoding process for the noisy case poses significant challenges. Sequential decoders are typically either bitor symbol-synchronous, i.e. they use one codeword at each step or they attempt to estimate one input bit at each step. The joint problem is such that the side information is available in the symbol domain, but the channel model is available in the codeword domain. As a consequence, a novel decoder is required to successfully decode the received information in the joint scenario.

2. PROPOSED TECHNIQUE

Let us consider a binary sequence $\mathbf{X} = [X_0, X_1, \dots, X_{L-1}]$ with *L* binary symbols drawn with probabilities $p_0 = P(X_i = 0)$ and $p_1 = P(X_i = 1)$, and the correlated side information $\mathbf{Y} = [Y_0, Y_1, \dots, Y_i, \dots, Y_{L-1}]$. The goal of joint source channel distributed encoding is to encode and reliably transmit \mathbf{X} across a communication channel by exploiting the fact that the correlated side information is available at the receiver.

In this paper we address the scenario in which \mathbf{X} is mapped onto the codeword $\mathbf{C} = [C_0, C_1, \dots, C_{N-1}]$ by means of the proposed modified AC. On the receiver side, \mathbf{C} is observed across a channel characterized by its transition probability $P(\mathbf{R}|\mathbf{C})$. Moreover, the distributed joint sourcechannel decoder (DJSCD) can exploit the knowledge of the correlated side information \mathbf{Y} . As usual in the related literature, we model the correlation by introducing a virtual correlation channel of known transition probability $P(\mathbf{Y}|\mathbf{X})$. The objective of the DJSCD is to select the most likely transmitted sequence according to the maximum a posteriori criterion (MAP) $\hat{\mathbf{X}} = \arg \max_{\mathbf{X}} P(\mathbf{X}|\mathbf{Y}, \mathbf{R})$.

2.1. The encoder

Standard arithmetic coding operates by mapping the source symbols onto sub-intervals of the [0, 1) probability space. The coding procedure is based on a recursive interval selection: for each input symbol X_i , the coding interval is partitioned into two adjacent sub-intervals whose lengths are proportional to p_0 and p_1 , respectively. The sub-interval representing X_i is



Fig. 1. Joint source channel distributed coding intervals.

selected for the following iteration. When all the N symbols have been considered, **X** is mapped onto a binary codeword **C** that can consist in any binary representation of a number lying in *I*; this mapping requires approximately $-\log_2 |I|$ bits, where |I| is the length of *I*. For large N the coding rate r(X) approaches the source entropy H(X).

In the past years two modifications have been proposed to achieve error resilience and distributed coding separately. The first approach amounts to reserve a given portion ϵ of the probability interval to the so called *forbidden symbol* μ . μ is never encoded and serves as a continuous error detection mechanism on the decoder side [5, 6]. The coding rate increases to $r(X; \epsilon) = H(X) - \log_2(1-\epsilon)$ and the presence of μ allows one to drive the MAP decoding algorithm. The second AC modification is based on the insertion of an ambiguity in the encoding process [1] and leads to the definition of the distributed arithmetic coding (DAC). The DAC is obtained using larger probability intervals, i.e., proportional to the modified probabilities $\widetilde{p}_j = \alpha_j p_j$, j = 0, 1 and $\alpha_j \ge 1$. In order to fit the enlarged sub-intervals \tilde{p}_j into the (0, 1] interval, the sub-intervals are allowed to partially overlap. The intuition is that, since the intervals are not disjoint, the decoder will typically be unable to decode the source unambiguously without knowledge of the side information. If one sets $\alpha_j = p_j^{-k}$ the coding rates turns to be r(X; k) = (1-k)H(X) and any rate r(X) < H(X) can be achieved selecting the proper value of k.

In this work we introduce both μ and symbol overlap at the same time so as to obtain a joint source channel distributed AC. We note that, while the joint encoder is a relatively simple combination of [5] and [1], the joint decoder requires a new design and will be described in Sect. 2.2. The coding intervals are modified as shown in Fig. 1. First of all, the forbidden symbol μ is allocated the probability ϵ on the right end of the unit interval. The source symbols probabilities are then reduced to $p'_j = (1 - \epsilon)p_j$ to compensate for the reduced probability space. Finally, the source symbol intervals are allowed to overlap by setting $\widetilde{p}_j = \alpha_j p'_j = \alpha_j (1 - \epsilon) p_j$. It is intuitive that the parameters k and α_i have opposite effects: the first one shrinks the intervals and increase the coding rate, the others reduce the coding rate by inserting ambiguity in the decoding process. In practice, to make the decoder more reliable, the last f symbols are encoded without overlap, i.e. k = 0. By setting $\alpha_j = (p'_j)^{-k}$, the coding rate in presence of both μ and symbol overlap can be written as

$$r(X;\epsilon,k,f) = \frac{(L-f)(1-k) + f}{L} \left(H(X) - \log_2(1-\epsilon) \right)$$
(1)

If one sets a target rate r^* , according to (1) the following relation between ϵ and k must hold:

$$k = \frac{L}{L - f} \frac{H(X) - \log_2(1 - \epsilon) - r^*}{H(X) - \log_2(1 - \epsilon)}$$
(2)

Therefore, given r^* and ϵ , (2) can be used to select the required value for k.

2.2. The decoder

Combining the error resilient and distribute coding capabilities at the encoder side is not particularly difficult. The most challenging issues are raised by the design of the decoder. In principle, as for the DAC and error resilient AC, decoding can be achieved by maximizing the MAP metric of the decoded sequence using a sequential search approach. To this end we have to employ a sequential decoder and introduce an additive MAP metric to rank the various decoding attempts.

Arithmetic decoding is an iterative task, where at each step the value of the coded sequence \mathbf{C} is used to select the interval corresponding to a decoded symbol. The selected interval is subdivided according to the known probabilities; the decoding process is repeated up to the detection of the *L* source symbols. All the results reported in this paper are obtained by representing the intervals in 32 bit fixed point format and using the typical normalization strategies to avoid underflow in the numerical representation of both \mathbf{C} and all the probability values.

Sequential arithmetic decoding can be designed according to a *bit-driven* or *symbol-driven* approach.

In the first case one constrains the decoder to consume only j out of the N codeword bits. Truncating the binary representation of C, means that we do not have enough numerical precision to select the intervals corresponding to all the source symbols. As soon as the numerical precision corresponding to the first j bits is not enough to discriminate between the sub-intervals, the sequential decoder is stopped yielding a partial decoding attempt. The sequential decoder state can be stored for subsequent use in future decoding attempts that consider more codeword bits. This strategy is used in [5] to represent the search space for MAP arithmetic decoding in presence of transmission errors as a binary tree, where branches are represented by alternative binary selections of the codeword bits $C_i = 0, 1$.

In the symbol driven case the whole C is available but the decoder is stopped as soon as i out of L source symbols are detected. This complementary approach has led to the design of the DAC decoder [1] evaluating conditional probabilities along a binary search tree where branches are triggered when

the numerical value of C falls in the overlapped, i.e. ambiguous, sub-interval. In case of ambiguity both the alternatives $X_i = 0, 1$ are tested and ranked in terms of the MAP metric.

The design of the DJSCD requires the conciliation of both strategies: on the one hand only the noisy observations **R** are available and therefore we need to consider a tree of decoding attempts triggered by opposite decisions for each C_i . On the other hand the presence of interval overlap introduces further branching when ambiguity is revealed. In conclusion, the DJSCD explores a set of sequential decoding attempts, each one represented by a different state of the sequential arithmetic decoder. Given a root decoder state $\chi(\mathbf{C}_0^{j-1}, \mathbf{X}_0^{i-1})$, obtained by assuming the first jcodeword bits $\mathbf{C}_0^{j-1} = [C_0, \ldots, C_{j-1}]$ and corresponding to $\mathbf{X}_0^{i-1} = [X_0, \ldots, X_{i-1}]$ decoded source symbols, two types of branching may occur.

- 1. Bit-driven branching: two alternative attempts corresponding to the states $\chi([\mathbf{C}_0^{j-1}, C_j = 0], \mathbf{X}_0^{i'-1})$ and $\chi([\mathbf{C}_0^{j-1}, C_j = 1], \mathbf{X}_0^{i''-1})$ can be performed by extending the decoding to the next codeword bit; this branching occurs only if the root state represents a decoder whose sequential decoding was interrupted as a consequence of the limited numerical precision carried by the first j bits of the codeword.
- 2. Symbol-driven branching: two alternative attempts corresponding to the states $\chi(\mathbf{C}_0^{j-1}, [\mathbf{X}_0^{i-1}, X_i = 0])$ and $\chi(\mathbf{C}_0^{j-1}, [\mathbf{X}_0^{i-1}, X_i = 1])$ are obtained by making a decision on the next source symbol X_i ; such branching is admissible only if the root state has terminated with an ambiguous decoding for the *i*-th source symbol.

Finally, by using an additive MAP metric and a sequential search algorithm the most likely decoding path, which consumes all the codeword bits and corresponds to *L* source symbols, is selected. The MAP metric is evaluated as follows:

$$P(\mathbf{X}|\mathbf{Y}, \mathbf{R}) = P(\mathbf{X}|\mathbf{Y}, \mathbf{C})P(\mathbf{C}|\mathbf{R}) \approx P(\mathbf{X}|\mathbf{Y}, \mathbf{C})\frac{P(\mathbf{R}|\mathbf{C})}{P(\mathbf{R})}$$

where in the last equality we neglect the term $P(\mathbf{C})$, i.e. we assume that all the codewords are equally likely. This hypothesis follows from the observation that arithmetic coding yields a codeword with $P(C_i) \approx 0.5$. The metric (3) can be recast into an additive metric in the logarithmic domain. The term $P(\mathbf{X}|\mathbf{Y}, \mathbf{C})$ depends on the correlation with the side information and it is the MAP metric used in [1] for the DAC. The term $P(\mathbf{R}|\mathbf{C})/P(\mathbf{R})$ depends on the channel transition probability and can be evaluated as in [5]. The proposed MAP metric can be updated sequentially for every decoding attempt adding a correlation term each time a decision on a source symbol X_i is taken and the corresponding channel term each time a new coded bit C_j is tested.

As for the sequential search algorithm, the most viable approach turns out to be the *Stack Algorithm* (SA), which is

a depth first technique that starts from the root node and always extends the path with the best accumulated MAP metric one step forward. SA allows one to move in depth along the tree of the decoding attempts by performing either a bitdriven or symbol-driven branching. It is worth pointing out that this sequential search is very innovative in the context of AC; in fact, in all previous distributed or error resilient AC decoder implementations either the symbol-driven or bit-driven approach have been used separately. A maximum memory M is used to store previous decoding attempts for backtracking. As soon as the forbidden symbol is revealed, the corresponding path is dropped. For more details on SA see [5] and reference therein.

3. EXPERIMENTAL RESULTS

The proposed distributed joint source-channel system has been implemented and its performance evaluated in the following scenario. The correlated side information is obtained by means of a binary symmetric channel with a given transition probability, corresponding to a certain value of the conditional entropy $H(\mathbf{X}|\mathbf{Y})$. The received codeword **R** is observed across the additive white Gaussian channel using binary phase-shift keying modulation with known signal to noise ratio (SNR) E_b/N_0 , being E_b the energy per transmitted bit and $N_0/2$ the noise variance. The DJSCD determines the MAP metric using the demodulated soft values. On the decoder side we estimate the Frame Error Rate (FER), i.e., the probability $P(\mathbf{X} \neq \mathbf{X})$ and the Bit Error Rate (BER) achieved by DJSCD. A minimum of 10^4 decoding trials are performed for each estimate. Due to space limitation, in the following only the BER will be reported. The simulations are performed by fixing the coding rate $r(X; \epsilon, k, f) = r^*$ and using (2) to select the corresponding values of ϵ and k. In all the reported experiments we used f = 20 non overlapped symbols to terminate the arithmetic encoder. The memory used by SA has been set to M = 2048.

⁽³⁾ The performance of DJSCD is compared with that obtained by a system based on turbo codes, where distributed joint source channel coding is achieved by puncturing parity bits (which is the optimal strategy according to [4]) down to the desired coding rate r^* . We use turbo codes with rate- $\frac{1}{2}$ generator (31,27) octal (16 states), and employ S-random interleavers. The BCJR algorithm, based on a modified MAP metric taking into account the side information correlation, is used at the receiver with 15 iterations.

In Fig. 2 the BER achieved by DJSCD and turbo codes in the case L = 200, $p_0 = 0.5$, $H(\mathbf{X}|\mathbf{Y}) = 0.25$ bits per symbol (bps) when setting $r^* = 0.6$ bps is shown versus the channel SNR. 5 choices of ϵ and k, equivalent in terms of the coding rate, are compared. It can be noted that the proposed solution is more reliable than turbo codes over a wide range of SNR values. As an example, when $E_b/N_0 = 7dB$ the DJCSD BER is $6 \cdot 10^{-5}$ whereas TC yields $2 \cdot 10^{-4}$. In the same



Fig. 2. BER in the case $L = 200, p_0 = 0.5, H(\mathbf{X}|\mathbf{Y}) = 0.25$ bps, $r^* = 0.6$ bps.



Fig. 3. BER in the case L = 200, $p_0 = 0.9$, $H(\mathbf{X}|\mathbf{Y}) = 0.2$ bps, $r^* = 0.4$ bps.

figure we observe that the DJSCD performance depends on the probability allocation between the forbidden symbol and the interval overlap. It can be noted that, even in the case $E_b/N_0 = 25$ dB, which refers to the error free transmission of C, the forbidden symbol ($\epsilon \neq 0$) is useful. This behavior can be explained by the fact that μ allows to prune more decoding paths also in absence of transmission noise, thus helping to remove the ambiguity introduced with the overlap. This observation paves the way to further research in the area of DAC [1], where the use of μ has never been taken into account.

Fig. 3 and Fig. 4 show the BER in the case of a compressible **X** with $p_0 = 0.9$, $H(\mathbf{X}|\mathbf{Y}) = 0.2$ bps and $r^* = 0.4$ bps, for L = 200 and L = 1000. Comparing Fig. 2 and Fig. 3 one can notice that the gain of DJSCD with respect to turbo codes is more evident in the case $p_0 = 0.9$. Finally, in Fig. 4 the gain of DJSCD turns out to be much more limited because the error correction performance of the turbo codes improves for larger block lengths.



Fig. 4. BER in the case L = 1000, $p_0 = 0.9$, $H(\mathbf{X}|\mathbf{Y}) = 0.2$ bps, $r^* = 0.4$ bps.

4. CONCLUSIONS

We have presented a distributed joint source-channel coding scheme based an AC with sequential decoding, and a sequential decoder that merges the bit-driven and symboldriven approaches. Experimental results show that the proposed scheme outperforms turbo codes at short block lengths, and is even at medium block length. Moreover, the decoder design gives insight on how to improve the DAC performance in the case of error-free transmission exploiting the forbidden symbol.

5. REFERENCES

- M. Grangetto, E. Magli, and G. Olmo, "Distributed arithmetic coding for the Slepian-Wolf problem," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2245–2257, June 2009.
- [2] A. Liveris, Z. Xiong, and C. Georghiades, "Joint source-channel coding of binary sources with side information at the decoder using IRA codes," in *Proc. of IEEE MMSP*, 2002.
- [3] A. Heidarzadeh and F. Lahouti, "On robust syndrome-based distributed source coding over noisy channels using LDPC codes," in *Proc. of IEEE ICSPC*, 2007.
- [4] P. Tan, K. Xie, and J. Li, "Slepian-Wolf problem using parity approach and syndrome approach," in *Proc. of CISS*, 2007.
- [5] M. Grangetto, P. Cosman, and G. Olmo, "Joint source/channel coding and MAP decoding of arithmetic codes," *IEEE Transactions on Communications*, vol. 53, no. 6, pp. 1007–1016, June 2005.
- [6] J. Chou and K. Ramchandran, "Arithmetic coding-based continuous error detection for efficient ARQ-based image transmission," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 861–867, June 2000.