JOINT SOURCE-CHANNEL DECODING OF ENTROPY CODED MARKOV SOURCES OVER BINARY SYMMETRIC CHANNELS

K. P. Subbalakshmi and Jacques Vaisey

School of Engineering Science, Simon Fraser University
Burnaby, British Columbia, V5A 1S6
e-mail : {subbalak, vaisey}@cs.sfu.ca
Tel : (604) 291 3818, Fax : (604) 291 4951

ABSTRACT

This paper proposes an optimal joint source-channel, *maximum a posteriori*, decoder for entropy coded Markov sources transmitted over noisy channels. We introduce the concept of *incomplete* and *complete* states to deal with the problem of variable length source codes in the decoder. The proposed decoder is sequential, there by making the expected delay finite. When compared to the standard Huffman decoder, the proposed decoder shows a maximum improvement, of about 4dB in a modified signal to noise ratio and an improvement of 21.95% in percentage of bits that are received in an out of synchronization condition, for a simple test source.

1. INTRODUCTION

Shannon's source-channel separation theorem [1] holds only under asymptotic conditions where both codes are allowed infinite length and complexity, which is not possible in practice. Hence, joint source-channel encoding and decoding have been gaining considerable attention as viable alternatives for achieving reliable communication of signals across noisy channels. Joint source-channel encoders (JSCE) aim at designing a single code that combines both the source and the channel codes so as to minimize the average distortion between the source and its reproduction, while joint source-channel decoders (JSCD) use the knowledge of the channel characteristics in the optimization of the decoder.

JSCD schemes have been well studied for sources which are *not* entropy coded [2] [3]. Until recently, JSCDs for entropy coded sources were not developed mainly because handling error propagation in entropy codes is difficult. In this paper, we propose the optimal decoder based on minimizing the probability of error in the decoded sequence, or in other words the *maximum a posteriori* (MAP) decoder, for entropy coded Markov sources transmitted over *binary symmetric channels* (BSC). This is a generalization of the work done by the authors [4] for memoryless sources. The proposed decoder does not assume the knowledge of the number of samples transmitted (as opposed to [5]) and we circumvent the problems of complexity faced in the JSCD developed in [6] by the use of the proposed state space structure. The following section states the MAP decoding problem and describes the proposed state space to tackle the decoding complexity and the variable length nature of the code.

2. THE STATE SPACE AND THE MAP PROBLEM

Let $\mathbf{C}$ denote the set of all possible B-bit sequences of Huffman codewords. The α^{th} such sequence is then denoted by $\mathbf{c}_\alpha^T = \{c_{\alpha,i}\}_{i=1}^T$, where $c_{\alpha,i}$ is the codeword corresponding to the i^{th} symbol in the transmitted stream and T is the total number of codewords in the stream. Since there are many ways in which to partition the received bit stream to yield different index sequences, the problem is to find the "best" possible index sequence, given the source and channel statistics. We note that the different partitions of the received bit stream will, in general, lead to different number of codewords in the index sequence. Let $\mathbf{R}$ denote the set of all B-bit received streams and let the sequence $\mathbf{r}_j^{n(j)}$ represent the j^{th} stream, with $n(j)$ being the number of codewords in the stream. We also define the probability of the symbol s_i to be $\Pr(s_i)$ with $\Pr(s_i|s_k)$ representing the probability that s_i was transmitted immediately after s_k. If $\hat{j}$ represents the index of the most probable transmitted sequence, then our problem is to determine

$$\mathbf{c}_{\hat{j}}^{n(\hat{j})} = \arg\max_{\mathbf{c}_j^{n(j)}} \Big\{ \Pr(c_{j,1}) \epsilon^{d_H[c_{j,1},r_{j,1}]} (1-\epsilon)^{(l_1 - d_H[c_{j,1},r_{j,1}])} \prod_{k=2}^{n(j)} [\Pr(c_{j,k}|c_{j,(k-1)}) \epsilon^{d_H[c_{j,k},r_{j,k}]} (1-\epsilon)^{(l_k - d_H[c_{j,k},r_{j,k}])}] \Big\}$$

where $d_H[c_{j,k}, r_{j,k}]$ is the Hamming distance between the k^{th} transmitted codeword of a specific sequence $\mathbf{c}_j^{n(j)}$ and the k^{th} word of the received sequence $\mathbf{r}_j^{n(j)}$ and l_k is the length of these words.

We propose a novel state space for the MAP decoder (MAPD) consisting of two classes of states: the *complete* and the *incomplete* states. We describe these states using an example Huffman codebook: $\{A : 0, B : 10, C : 110, D : 111\}$. Let the first three bits of the transmitted bit stream be $\{0, 1, 0\}$ corresponding to the sequence A,B and the received bit stream be $\{0, 1, 1\}$. We see that the bit stream can be partitioned in seven ways as shown in Figure 1. The dots connected by solid lines represent a single completely received codeword, those connected by dotted lines denote bits of a codeword that is only partially received and the dark dots represent the one bit codeword. The first four cases (cases 1, 2, 3a and 3b) represent the situations when the received bits are completely partitioned into full codewords and hence these cases correspond to *complete states*. Cases 4, 5 and 6 correspond to situations when the received bits cannot be completely partitioned into full codewords, and hence they represent the *incomplete states*. The state space thus consists of six states: 4 degree 0 states (one for each codeword), a degree 1 and a degree 2 state.

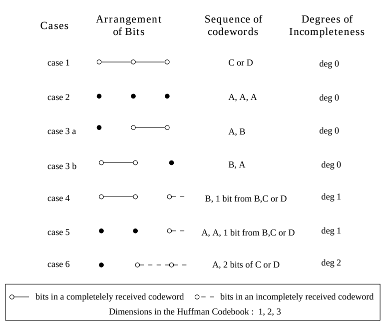

Figure 1: **Degrees of Incompleteness in the State Space**

We note that the number of degrees of incompleteness and the number of transitions between the states are determined by the lengths of the codewords in the Huffman codebook. For a Huffman codebook containing N codewords of varying lengths belonging to the set, $\mathcal{L} = \{l_{min}, \cdots, l_{max}\}$, the maximum degree of incompleteness is $l_{\max} - 1$. State transitions from the degree $k - 1$ state to the degree k state ($k > 0$) are possible at every stage of the algorithm, however a transition from the degree k ($k > 0$) to one of the degree 0 states is possible only if $k = l_i - 1$, where l_i is the length of the codeword represented by that degree 0 state. A trellis is used to show the evolution of the state space with time (as bits are received). At each stage of the trellis there are as many nodes as there are states; the state space and the transitions from state $i - 1$ to i are depicted in Figure 2.

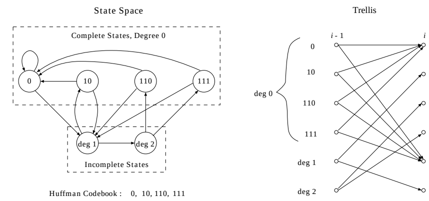

Figure 2: **MAPD State-Space and Trellis Diagram for Huffman Codebook :** $0, 10, 110, 111$**.**

3. THE MAP DECODING ALGORITHM

The key feature of our algorithm is the use of complete and incomplete states. Two operations take place at each stage of the decoding algorithm. One consists of examining the metrics of all the paths, entering each node (state-stage pair) in the trellis and the other consists of looking for a merger of the paths and the actual declaration of decoded codewords (if there is a merge). For complete states, the path update step involves finding the best metric path to the state and retaining it, whereas, for the incomplete states, we must retain the metric values of all the paths back to the last complete state.

Let $\mathbf{v} = (v_x, v_y)$ represent the node corresponding to state v_x at stage v_y. The complete states are numbered from 1 to N corresponding to the N codewords in the codebook and incomplete states are numbered from $N + 1$ to $N + l_{\max} - 1$. Let c_k denote the length d_k Huffman codeword corresponding to the complete state k and let $\mathbf{b}_{k,i}$ be the d_k most recent received bits at node (k, i). $\mathcal{M}[k, i, c_j]$ gives the metric increment associated with the path segment that begins at node $(j, (i - d_k))$ and terminates in the node (k, i) (where k and j are complete states). Thus, when $i > d_k$

$$\mathcal{M}[k, i, c_j] = \quad \log_{10}(\Pr\{c_k|c_j\}) + d_H[c_k, \mathbf{b}_{k,i}] \quad \log_{10}(\epsilon) + (d_k - d_H[c_k,$$

Now, $M_{k,i}$ is defined to be the metric value associated with the maximum metric path terminating in (k, i), where $k \in \{1, 2, \cdots, N\}$ and for the incomplete states we define $M_{k,i}(u)$ to be the u^{th} element in the vector of metric values that need to be remembered at (k, i), where $k \in \{N + 1, N + 2, \cdots, N + l_{\max} - 1\}$ and $u \in \{1, 2, \cdots, N\}$. Finally, let $\mathbf{v}^p$ be the parent of the node $\mathbf{v}$, which is defined as the penultimate node in the maximum metric path terminating at node $\mathbf{v}$ and let $d_{v_x^p}$ be the length of the codeword corresponding to the parent node (which is complete).
We now state the algorithm as:

Initialize:
Input first $l_{\min} - 1$ bits

For $i = 1, \cdots, l_{\min} - 1$,
  $M_{k,i} \leftarrow 0 \; \forall k \in \{1, 2, \cdots, N\}$.
  $M_{k,i}(u) \leftarrow 0 \; \forall \; u \in \{1, 2, \cdots, N\}; \; \forall k \in \{N+1, \cdots, N + l_{\max} - 1\}$.
$i \leftarrow l_{\min}$
For $k \in \{1, 2, \cdots, N\}$
  $\mathbf{v} = (k, i)$
  $\mathbf{v}^p = \Phi$ (no parents)

Input:
  Input bit at the i^{th} stage

Update path metrics:
  For $\{k = 1, 2, \cdots, N\}$

$$M_{k,i} \leftarrow \begin{cases} \max_j \{ M_{(N+d_k-1),(i-1)}(j) + \mathcal{M}[k, i, c_j] \}, \; i > d_k \\ \log_{10}(\Pr\{c_k\}) + d_H[c_k, \mathbf{b}_{k,i}] \log_{10}(\epsilon) + (d_k - d_H[c_k, \mathbf{b}_{k,i}]) \log_{10}(1-\epsilon), \; i = d_k \\ 0, \;\; i < d_k \end{cases}$$

$$M_{k,i}(u) \leftarrow \begin{cases} M_{u,(i-1)} & \forall \; u \in \{1, 2, \cdots, N\}, \; k = N+1. \\ M_{(k-1),(i-1)}(u) & \forall \; u \in \{1, 2, \cdots, N\}, \; \forall k \in \{N+2, \cdots, N + l_{\max} - 1\} \end{cases}$$

$$j_k^* \leftarrow \arg \max_j \{ M_{(N+d_k-1),(i-1)}(j) + \mathcal{M}[k, i, c_j] \}, \; \forall k \in \{1, 2, \cdots, N\}.$$

We note that updating $M_{k,i}(u)$ involves only a copy operation whereas updating $M_{k,i}$ involves some calculations.

Update paths:
  For $k \in \{1, 2, \cdots, N\}$
  $\mathbf{v} = (k, i)$
  $v_x^p = j_k^*; \; v_y^p = i - d_{j_k^*};$

Merge Check and Output:
  A merge is declared when all nodes at stage i arise from a common ancestor. For degree zero nodes, we trace back from the respective nodes at stage i; however, for any incomplete state k, we trace back from *all* of the complete states of stage $i - k + N$. This is so because, at incomplete states no parent node can be discarded as it may be the parent of some state at a later stage. In order to determine if there are merges we define a set of nodes $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{Nl_{\max}}\}$ and initialize them to the nodes from which we need to trace back in order to determine a merge. We need $Nl_{\max}$ such nodes since each degree of incompleteness contributes to N nodes and the N complete nodes contribute to one parent each.
  $\mathbf{V}$ is formed according to
  $\mathbf{v}_m = (m - N\lfloor \frac{m}{N} \rfloor, \; i - \lfloor \frac{m}{N} \rfloor), \; \forall m \in \{1, 2, \cdots, Nl_{\max}\}$.

  If $(i \geq l_{\max} + 1)$
  a: *If* (the parents of all the nodes in $\mathbf{V}$ are equal)
    decide that the path terminating in $\mathbf{v}_1$ was indeed transmitted and set the parent of all the nodes at this stage to be Φ.
   Else
    $\mathbf{v}_m \leftarrow \mathbf{v}_m^p, \; \forall \, m \in \{1, 2, \cdots, Nl_{\max}\}$
  If $(\mathbf{v}_m^p \neq \Phi \; \forall \, m \; \{1, 2, \cdots, Nl_{\max}\})$
    go to **a**.
  If (end of sequence)
  $k^* \leftarrow \arg \max_{k \in \{1,2,\cdots,N\}} M_{k,i}$
  Declare the path terminating at (k^*, i) as the optimal path.
  Else
  $i \leftarrow i + 1$
  go to **Input**.

4. EXPERIMENTAL RESULTS AND CONCLUSIONS

We performed experiments on 5000 samples of a zero mean, first order, Gauss-Markov source with correlation coefficient $\rho = 0.97$, quantized with a 50 level, uniform quantizer with step size 0.55. We pick a source with high correlation coefficient, in order to allow the MAP decoder to exploit the memory in the source. The symbols were then Huffman encoded and corrupted by random bit error patterns representing the BSC. The results presented are an average of six channel realizations. The performance of the MAPD is compared to that of the Huffman decoder (HD) using two parameters: the percentage of bits out of synchronization and the modified signal to noise ratio (MSNR) of the decoded stream. The MSNR is defined so as to capture the performance of the decoders in the portions where the decoded signal is in synchronization with the original signal and the percentage of bits that are out of synchronization demonstrates the efficacy of the decoder in controlling synchronization losses. The MSNR is calculated by aligning the synchronous segments of the transmitted and decoded signals. The decoded segments are padded with zeros when the number of decoded words are less than the number of transmitted words and are truncated when there are more words in the decoded segment than in the transmitted one.

Figure 3 shows the comparison of the decoded MSNR for both the decoders. We note that at very low error rates, the MSNRs of both the decoders are nearly the same; however, as the error rate increases the MAPD performs better than the HD, reaching a maximum improvement of 4dB. As the error rates become very high, the HD shows better MSNRs; however, the MSNR is negative and hence this region is not of much practical significance. We note here that when the ordinary SNR is used as a performance measure, the maximum improvement is about 6dB and there is no qualitative difference in the results.

Figure 4 shows the average percentage of bits that are out of synchronization in the decoded sequence, for both the HD and MAPD. The trends in the relative performance of the MAPD versus the HD is just as in the MSNR profile, with the maximum improvement in percentage loss of synchronization being 21.95%.

Finally, Table 1 gives the statistics of synchronization loss for selected error rates. Since we need a significant number of synchronization loss events in order to calculate the statistic, we used $20,000$ source symbols in this case. The results indicate that both the number and length of synchronization losses are significantly lower in the MAP de-

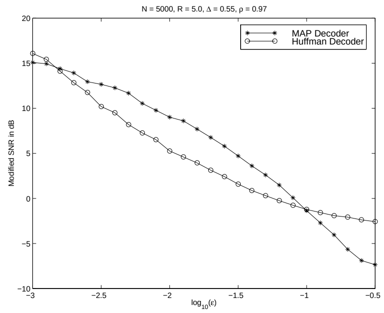

Figure 3: Comparison of the modified SNR of the MAPD with the HD for an AR(1) source with $\rho = 0.97$, quantized to $N = 50$.

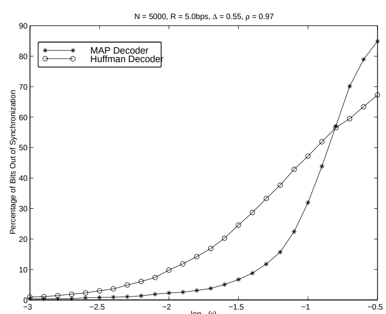

Figure 4: Comparison of the percentage loss of synchronization of MAP decoded sequence with that of the Huffman decoded sequence for an AR(1) source with $\rho = 0.97$, quantized to $N = 50$.

coded sequence than in the Huffman decoded sequence.

Error Rate	**MAP Decoded Sequence**				**Huffman Decoded Sequence**			
$log_{10}(\epsilon)$	Number		Length		Number		Length	
	Avg.	Var.	Avg.	Var.	Avg.	Var.	Avg.	V
-0.5	2278.00	1437.67	38.83	1337.95	2187.50	37.58	31.27	701
-1.0	1237.00	249.00	27.04	444.62	1393.17	329.14	34.22	885
-1.5	304.83	84.81	22.44	254.85	685.00	90.00	35.98	105
-2.0	99.67	82.89	21.47	204.45	261.83	335.14	38.87	133
-2.5	41.50	2.58	20.21	166.21	87.17	95.81	37.86	108
-3.0	20.00	2.00	19.17	137.63	28.50	7.92	36.71	147

Table 1: Synchronization Loss Statistics

In summary, we have presented a new joint source-channel MAP decoder for entropy coded sources using a new state space structure for the decoder. Simulation results have demonstrated that this decoder does better than the conventional decoder significantly at low to medium channel error rates and restricts both the number and the length of synchronization losses in the decoded sequence.

5. REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, vol. 27, pp. 379–423 and 623–656, 1948.

[2] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source channel coders," *IEEE Transactions on Communications*, vol. 39, pp. 839–846, June 1991.

[3] N. Phamdo and N. Farvardin, "Optimal detection of discrete markov sources over discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 40, pp. 186–193, January 1994.

[4] K. P. Subbalakshmi and J. Vaisey, "Optimal decoding of memoryless sources over binary symmetric channels," in *IEEE Data Compression Conference*, p. 573, 1998.

[5] N. Demir and K. Sayood, "Joint source/channel coding for variable length codes," in *IEEE Data Compression Conference*, pp. 139–148, 1998.

[6] M. Park and D. J. Miller, "Decoding entropy-coded symbols over noisy channels by MAP sequence estimation for asynchronous HMMs," *Conference on Information Science and Systems*, 1998.