Length-Constrained MAP Decoding Revisited

Zhe Wang, Xiaolin Wu and Sorina Dumitrescu
Department of Electrical and Computer Engineering
McMaster University, Hamilton, ON, Canada, L8S 4K1

Abstract—In this paper, we consider the problem of length-constrained maximum a posterior (MAP) decoding of a Markov sequence that is variable length encoded and transmitted over a binary symmetric channel. We convert this problem into one of maximum-weight $k$-link path in a weighted directed acyclic graph. The induced graph optimization problem can be solved by a fast parameterized search algorithm that finds either the optimal solution with high probability or a good approximate solution otherwise. The proposed algorithm has lower complexity and superior performance than the previous heuristic algorithms.

I. INTRODUCTION

In practice, due to the constraint of system complexity, the source encoder is almost always suboptimal in the sense that it fails to remove all the redundancy from the source. This residue redundancy makes it possible for the decoder to detect and correct channel errors, even in the absence of channel code [3]. Consider the situation that a scalar-quantized Gaussian Markov source sequence \{\(X_i\)\} is compressed by Huffman code that only approaches the self-entropy \(H(X_i)\) of the source. The residue redundancy \(H(X_{i+1}|X_i) = H(X_{i+1}) - H(X_i)\) can be used by a MAP decoder that exploits the source memory to combat channel noises.

The number of source symbols in the transmitted sequence, if known to the decoder, can help to improve the MAP decoding performance, because it puts a necessary condition on the correctness of decoded sequence. The objective of length-constrained MAP decoding is to examine all sequences with a specified number of symbols and find the one with the maximum a posterior probability. Park and Miller solved the problem with a specified number of symbols and find the one with the maximum-length MAP decoding is to examine all sequences \(G\) in \(G\) with the induced graph optimization problem can be solved of maximum-weight \(k\)-link path in a weighted directed acyclic graph (DAG) that generates \(K\) edges in the maximum-weight path of \(G\), where \(K\) is high in practice. In case the proposed algorithm fails to find such a \(K\)-path, it still offers an approximate solution by finding a maximum-weight path whose edge count is the closest to \(K\). Furthermore, this approximate solution can be refined by a local adjustment such that the constraint of edge count can be met. This refined approximation outperforms the approximate algorithm of [4] at a comparable cost. Complexity analysis of various MAP decoding algorithms is offered in Section III. Finally, the experimental results are presented in Section V.

II. PROBLEM FORMULATION

In this paper, we consider the discrete memoryless binary symmetric channel (BSC). We assume the source to be a scalar-quantized first-order Markov sequence, which is a first-order discrete Markov process with alphabet \(I = \{0, 1, \cdots, N - 1\}\). The process \(X_t\) can be completely characterized by transition probabilities \(p(i|j) = Pr(X_t = i|X_{t-1} = j)\), for all \(i, j \in I\), and the prior probability distribution, \(p(i_0) = Pr(X_0 = i_0), i_0 \in I\). The scalar quantization indexes are compressed by a VLC source encoder, such as Huffman code, whose codebook is \(C = \{c_0, c_1, \cdots, c_{N-1}\}\). If the input of the VLC encoder is \(i\), we denote the output by \(c_i = C(i)\).

A sequence of \(K\) symbols generates a sequence of indexes \(I = i_0i_1\cdots i_{K-1}\) after quantization. The input to the BSC is \(C(I) = C(i_0)C(i_1)\cdots C(i_{K-1})\), which is a binary sequence of length \(M = \sum_{i=0}^{K-1}|C(i_i)|\), where \(|\cdot|\) is the length of a binary sequence in bits. Since the channel has no insertion/deletion errors, the decoder will receive a binary sequence of the same length, denoted by \(y = y_0y_1\cdots y_{M-1}\). Given \(y\) and \(K\), the channel input must be a sequence of \(K\) symbols and have length of \(|y|\) in bits. The decoder should find the \(\hat{I}\) that maximizes \(P(I|y)\) over all permissible \(I\). Using the Bayesian rule, this is to maximize \(P(y|I)P(I)/P(y)\), which is equivalent to maximizing \(P(y|I)P(I)\).

For a particular \(I = i_0i_1\cdots i_{K-1}\), we can calculate the...
probability of \( I \) as
\[
P(I) = p(i_0) \cdot \prod_{t=1}^{K-1} p(i_t|i_{t-1}).
\] (1)

Consider a parsing of \( y \) with respect to \( I \), denoted by \( y(m_0, m_1, \cdots, m_K-1, m_K : I) \), where \( m_0 = 0 \) and \( m_K = M \), such that the substring of \( y \) from position \( m_t \) to \( m_{t+1}-1 \) is the channel output of codeword \( C(i_t) \). For a cleaner notation, we write the substring \( y_{m_t} \cdots y_{m_{t+1}-1} \) as \( y|m_t, m_{t+1}\). Then we have
\[
P(y|I) = \prod_{t=0}^{K-1} P_c(C(i_t), y|m_t, m_{t+1}).
\] (2)

\( P_c(s, t) \) is the probability of having \( s \) as the channel output of \( t \), where \( s \) and \( t \) are two binary strings of the same length. For BSC, it can be calculated as
\[
P_c(s, t) = p_c H_d(s, t) \cdot (1 - p_c)^{|s| - H_d(s, t)},
\]
where \( p_c \) is the crossover probability, and \( H_d(s, t) \) is the Hamming distance between \( s \) and \( t \).

Then the objective of length-constrained MAP decoding is to find
\[
\hat{I} = \arg \max_{I \in S_K(y)} P(y|I) \cdot P(I),
\] (3)
where
\[
S_K(y) = \{ I = i_0i_1 \cdots i_{K-1} \mid |C(I)| = |y| \}
\]
is the set of all permissible input sequences consisting of \( K \) source symbols and having \( |y| \) bits in length.

III. GRAPH REPRESENTATION AND ALGORITHMS

A. Graph Construction

To facilitate the development of efficient algorithm for the above optimization problem, let us construct a weighted directed acyclic graph (DAG) \( G \). This weighted DAG \( G \) has \( NM + 1 \) nodes, where \( M = |y| \) is the length of received bit sequence, and \( N \) is the size of the VLC codecbook. The graph has a unique starting node \( s \), and all other nodes are grouped into \( M \) stages. Each stage has \( N \) nodes, indexed from 0 to \( N-1 \), as shown in Fig. 1. Each stage corresponds to a bit location in the received sequence \( y \). Node \( s \) is at the 0-th stage. The nodes at the \( M \)-th (last) stage are so-called final nodes. Denote by \( F \) the set of all final nodes, marked by double circles in Fig. 1. We use \( n_i^m \) to label the \( i \)-th node at stage \( m \). Node \( n_i^m \) corresponds to codeword \( c_i \) that is parsed out of \( y|m - |c_i|, m) \). At the bit location \( m \) in \( y \). From \( n_i^m \) to \( n_i^{m+|c_i|} \), there is an edge corresponding to the probability event that the substring \( y|m, m + |c_i| \) is decoded as \( c_i \), given that the previously decoded codeword is \( c_j \). From the starting node \( s \) there is an edge to each node \( n_i^{|c_i|} \), \( 0 \leq i \leq N - 1 \), corresponding to the probability event that the substring \( y[0, |c_i|) \) is decoded as \( c_i \). Generally, for an \( n_i^m \), there are \( N \) incoming edges, one from each of the nodes on stage \( m - |c_i| \); there are \( N \) outgoing edges emitted from \( n_i^m \), one to each node \( n_j^{m+|c_i|} \), \( 0 \leq j \leq N - 1 \). The weight of outgoing edge emitted from \( s \) to \( n_i^{|c_i|} \) is \( \log p(i) + \log P_c(C(i), y[0, |c_i|]) \), \( 0 \leq i \leq N - 1 \); the weight of the edge from \( n_j^m \) to \( n_i^{m+|c_i|} \) is assigned to be \( \log p(i) + \log P_c(C(i), y[m, m + |c_i|]) \).

Fig. 1. Weighted directed acyclic graph for MAP decoding of a variable length code of a Markov sequence. In this example, the codebook has one 1-bit codeword (\( c_1 \)), and two 2-bit codewords (\( c_2 \) and \( c_3 \)). For clarity, only directed edges to and from the nodes on stage \( m \) are drawn.

B. Solution

With the definition of \( G \), we can convert the optimization problem in (3) to finding the maximum weighted \( K \)-link path in \( G \). The objective function in (3) is equivalent to
\[
\hat{I} = \arg \max_{I \in S_K(y)} \log(P(y|I) \cdot P(I)),
\] (4)
since logarithm is a monotonic function. By expanding the product terms inside the logarithm operator in (4), we have
\[
\hat{I} = \arg \max_{I \in S_K(y)} \left\{ \log \left( \sum_{t=1}^{K-1} p(i_t|i_{t-1}) + \log P_c(C(i_t), y[m, m_{t+1}] \right) \right\}.
\] (5)

Then it follows from (1), (2) and (5), that \( \hat{I} \) is determined by the maximum-weight \( K \)-link path (i.e., the path of maximal weight over all paths with \( K \) edges) from \( s \) to \( F \).

To this end, we can use a parameterized search technique of Aggarwal et al [1] to find the required path. Let us restate below two lemmas proved in [1]. For any real number \( \tau \), define a new weighted DAG \( G(\tau) \) that is derived from the same sets of nodes and edges as \( G \). The weight of an edge \( e \) in \( G(\tau) \) is the sum of the weight of \( e \) in \( G \) and \( \tau \). Then we have:

Lemma 1: If for some real \( \tau \), the maximum-weight path from \( s \) to \( F \) in \( G(\tau) \) has \( K \) links, then this path is the maximum-weight \( K \)-link path from \( s \) to \( F \) in \( G \).

Thus, if there exists a real number \( \tau \) such that the maximum-weight path from \( s \) to \( F \) in \( G(\tau) \) has exactly \( K \) edges, then this path is the maximum-weight \( K \)-link path, which in return solves the length-constrained MAP decoding problem exactly.

Lemma 2: Suppose the maximum-weight path from \( s \) to \( F \) in \( G(\tau) \) has \( K \) links. Then for every \( \beta < \tau \), the maximum-weight path from \( s \) to \( F \) in \( G(\beta) \) has at most \( K \) links.

Lemma 2 implies that bisection search can be used to search for the optimal parameter \( \tau \). Given a \( \tau \) algorithm in [6] can be directly applied to find the maximum-weight path in \( G(\tau) \) in \( O(N^2M) \) time. For a general weighted directed acyclic graph induced by MAP decoding, there is no
guarantee for a real value \( \tau \) to exist such that there is a path of \( K \) edges in \( G(\tau) \). Fortunately, in practice, the probability that such a \( \tau \) exists is very high. This success probability is tabulated in Table 1 for different sequence lengths and crossover probabilities. We observe in experiments that the proposed MAP decoding algorithm has higher than 0.9 chance to obtain the globally optimal solution when \( p_c \approx 10^{-2} \) and the sequence length \( K \leq 100 \). Furthermore, if such a \( \tau \) exists, it can be found by bisection in few steps (see Table 2). Since it takes a constant number of iterations to find the required \( \tau \), the overall complexity of the proposed algorithm is \( O(N^2M) \). Comparing with Park and Miller’s \( O(N^2M^2) \) algorithm, our algorithm is faster by a factor of \( M \), which is significant since \( M \) is very large in practice. Also, if the input is a Gaussian-Markov sequence, we can adopt a fast matrix search technique to reduce the complexity further to \( O(NM) \) [7].

<table>
<thead>
<tr>
<th>length</th>
<th>( p_c = 10^{-3} \cdot 10^{-2} \cdot 10^{-1} \cdot 10^{-0} \cdot 10^{+1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.99 0.97 0.96 0.83 0.58</td>
</tr>
<tr>
<td>100</td>
<td>0.99 0.96 0.83 0.58</td>
</tr>
<tr>
<td>500</td>
<td>0.98 0.95 0.84 0.67 0.49</td>
</tr>
</tbody>
</table>

Tab. 1. The probability that the proposed algorithm finds exact solution.

<table>
<thead>
<tr>
<th>length</th>
<th>( p_c = 10^{-3} \cdot 10^{-2} \cdot 10^{-1} \cdot 10^{+0} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>1.01 1.07 1.21 1.82 3.93</td>
</tr>
<tr>
<td>100</td>
<td>1.04 1.13 1.47 2.63 5.20</td>
</tr>
<tr>
<td>500</td>
<td>1.22 1.70 3.16 5.46 7.60</td>
</tr>
</tbody>
</table>

Tab. 2. Average number of steps in bisection search to find the optimal parameter \( \tau \), if it exists, for different sequence lengths and crossover probabilities.

IV. REFINEMENT OF APPROXIMATE SOLUTION

As shown in Table 3 for long sequences and high crossover probabilities, the proposed algorithm has a relatively high chance to miss the optimal solution. The algorithm knows when this happens after the bisection search fails to find a \( \tau \) value such that the maximum-weight path of graph \( G(\tau) \) has exactly \( K \) edges. However, the bisection search process generates a series of maximum-weight paths of different edge counts. Among these paths are the path \( P_0 \) of \( K_0 \) edges and the \( P_1 \) of \( K_1 \) edges such that \( K_0 \) is the largest edge count that is smaller than \( K \), and \( K_1 \) is the smallest edge count that is greater than \( K \). The two paths \( P_0 \) and \( P_1 \) are the closest to the optimal solution in the sense that they offer the tightest bound \( K_0 < K < K_1 \) in edge count.

Instead of taking either \( P_0 \) or \( P_1 \) as an approximate solution, we can refine the approximation by making adjustments to these two paths. The two paths \( P_0 \) and \( P_1 \), both starting from \( s \) and ending in \( F \), typically agree with each other in most parts. In other words, they share a few long common subpaths, and the subpaths where they differ from each other are short and far between. This observation suggests various heuristic techniques of combining and refining the two approximations into a path whose edge count meets the constraint of \( K \).

The output sequences of the two different MAP decodings by paths \( P_0 \) and \( P_1 \) consist of convergent segments and divergent segments, corresponding to different parsings of the received bit sequence \( y \) by the common and different subpaths of \( P_0 \) and \( P_1 \). A node of \( G \) at which \( P_0 \) and \( P_1 \) start to diverge is called divergent node; likewise, a node at which \( P_0 \) and \( P_1 \) converge is called convergent node. As shown in Fig. 2, we construct a new graph \( G' \) by removing from the original graph \( G \) the stages in which \( P_0 \) and \( P_1 \) have the same subpath, and concatenating the stages in which \( P_0 \) and \( P_1 \) have different subpaths. The concatenation is carried out at the consecutive divergent and convergent nodes. A dummy edge is introduced to connect two neighboring divergent and convergent nodes, whose weight is set to zero. Consequently, the new graph \( G' \) is a much reduced subgraph of \( G \) with a single starting node. Let \( K_c \) be the number of edges of all the common subpaths of \( P_0 \) and \( P_1 \). Now we convert the problem of constructing a maximum-weight path of \( K \) edges in \( G \) to one of finding the maximum-weight path of \( K' = K - K_c \) edges in \( G' \).

In practice the total length (in bits) of divergent segments is quite small. Table 1 lists the average divergent lengths for different sequence lengths. Interestingly, the total length of divergent segments relates to crossover probability only (which remains unexplained as of now). Thus, we can apply the exact algorithm in [4] on \( G' \) to find the optimal maximum-weight \( K' \)-link path in \( O(N^2M^2) \) time, where \( M' \) is the length of all divergent segments. Since \( M' << M \), this gives a very fast approximate solution to the original constrained optimization problem.

<table>
<thead>
<tr>
<th>length</th>
<th>( p_c = 10^{-3} \cdot 10^{-2} \cdot 10^{-1} \cdot 10^{+0} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>22.98 22.99 22.78 24.85 32.17</td>
</tr>
<tr>
<td>100</td>
<td>23.10 23.88 23.71 25.48 34.42</td>
</tr>
<tr>
<td>500</td>
<td>20.58 22.18 23.34 26.76 36.48</td>
</tr>
</tbody>
</table>

Tab. 3. Average length of divergent segment (in bits) for different length sequences.

V. EXPERIMENTAL RESULTS

The simplest performance measure of a MAP decoder is a symbol-by-symbol difference (inner product) such as PSNR in the case of signal compression. However, for most applications a meaningful distance between two sequences is not as simple as pairwise distortion, even if they have the same length. For instance, a MAP-decoded sequence with errors may contain subsequences that are in the original input sequence only with some shifts. But a symbol-by-symbol distortion measure
may mistakenly quantify these subsequences as complete loss. Instead, a string edit distance is more appropriate to measure the performance of MAP decoding. In the following evaluation of MAP decoding performance, a decoded sequence is first aligned to the original sequence by minimizing the Levenshtein distance between them. This is to find an alignment scheme with the minimum number of insertions, deletions and substitutions of symbols to transform the decoded sequence to the original one.

Suppose after the minimum edit distance alignment, the estimated MAP sequence $\hat{I}$ is adjusted to $\hat{I} = \cdots s_is_ds_j \cdots$, where the subsequences $s_i$ and $s_j$ agree with the original input symbol by symbol, but $s_d$ differs from the original in all of its symbols. Then we count $s_d$ as one error propagation of length $e_l = |s_d|$. We use the mean error propagation length $\bar{e}_l$, and the number of error propagations, $e_n$, as the measures of MAP decoding performance. The former measures the length of burst errors due to loss of synchronization of the VLC decoding, while the latter measures the frequency of desynchronization. Of course, $e_n \bar{e}_l$ equals to the total number of decoding errors after the alignment of minimum edit distance.

In our experiments MAP decoding is applied to a scalar-quantized (uniform with nine codecells) zero-mean, unit-variance, first-order Gaussian-Markov process of correlation coefficient 0.9. The performance results of the MAP decoding algorithms being evaluated were averages of 500 simulation runs on test sequences of different lengths generated by the above source model. The test sequences were encoded at an average rate of about 3 bits/sample and were transmitted through a BSC of various crossover probabilities $p_c$. In Fig. 3, we plot the mean error propagation length $\bar{e}_l$ of different MAP decoding algorithms versus crossover probability $p_c$. In order to evaluate the effects of sequence length on length-constrained MAP decoding performance three groups of curves for different sequence lengths $K = 50, 100, 500$ are plotted. The experiments show that our algorithm outperforms that in [4].

When measured by the number of error propagations $e_n$, our algorithms also outperforms Park and Miller’s approximate algorithm. In Fig. 4 we plot the number of error propagations $e_n$ as a function of $p_c$ for two different sequence lengths $K = 50, 100$. For sequences of 500 symbols, the relative ranking of the three different algorithms remains the same.

Fig. 4. Performances of different algorithms measured by the number of error propagations versus crossover probability and for different sequence lengths.

We also implemented the exact $O(N^2M^2)$ algorithm proposed in [4] as a reference of evaluation. As shown in Fig. 3 and Fig. 4, the performance of the presented algorithm is very close to the optimal, because it has a high probability to find the globally optimal solution; even when it fails, the local adjustment technique presented in the previous section can improve the performance. Another observation made in our experiments is that the length-constrained MAP decoding algorithms perform better for shorter sequences. This should be expected since short sequence length means stronger prior knowledge on a per-symbol basis.

REFERENCES


Fig. 3. Performances of different algorithms measured by mean error propagation length $\bar{e}_l$ versus crossover probability $p_c$ and for different sequence lengths.