Fast Algorithms for MAP Decoding of VLC-coded Markov Sequences over Noisy Channels

Xiaolin Wu

Department of Electrical and Computer Engineering

McMaster University

Joint Source-Channel Decoding

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.

Consider a Markov source sequence $\{X_i\}$ compressed by Huffman code that only approaches the self-entropy $H(X_i)$. The residue redundancy $H(X_{i+1}|X_i) = H(X_i, X_{i+1}) - H(X_i)$ can be use to combat channel noise.

MAP decoding is to estimate the channel input that maximizes the *a posterior probability* of the channel output. In other words, the decoder examines all the possible channel input sequences and finds the one with the maximal a posterior probability.

Channel Model

- error-and-erasure memoryless channel (EEC) of inversion and erasure errors.
- input $\mathbf{b} = b_1 b_2 \cdots b_k \in \{0,1\}^k$.
- output $\mathbf{b}' = b'_1 b'_2 \cdots b'_k \in \{0,1,\$\}^k$
- probability of receiving $\mathbf{b}'$ when $\mathbf{b}$ is sent:

$$P_e(\mathbf{b}'|\mathbf{b}) = \prod_{i=1}^{k} P_e(b'_i|b_i). \tag{1}$$

  This probability can be computed given the channel transition matrix.

Markov Sequence Coded by a VLC

- VLC codebook: $C = \{c_1, c_2, \ldots, c_N\}$.
- first order Markov source with conditional probabilities $P(c_j|c_k)$, $1 \le j, k \le N$.
- for an arbitrary sequence $\mathbf{x} = x_1 x_2 \cdots x_I \in C^I$, we have:

$$P(\mathbf{x}) = P(x_1) \prod_{i=2}^{I} P(x_i|x_{i-1}). \tag{2}$$

- The Markov sequence is coded by VLC C.

Markov Sequence Sent through the EEC

- the Markov sequence $\mathbf{x} = x_1 x_2 \cdots x_I \in C^I$ is sent through the EEC.
- channel output: $\mathbf{y} = y_1 y_2 \cdots y_M \in \{0, 1, \$\}^M$, where M equals the number of bits of the VLC-coded input sequence.
- a unique parsing of $\mathbf{y}$ exists s.t. the parsed i-th word is the output corresponding to the i-th codeword in the input sequence:

$$y_{m_0+1}..y_{m_1}, y_{m_1+1}..y_{m_2}, ..., y_{m_{I-1}+1}..y_{m_I}, \tag{3}$$

  where $m_0 = 0$ and $m_i - m_{i-1} = |x_i|$ for all $1 \leq i \leq I$. Let subsequence $y_{m_{i-1}+1}..y_{m_i}$ be $y(m_{i-1}, m_i]$, then

$$P_e(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{I} P_e(y(m_{i-1}, m_i]|x_i). \tag{4}$$

Objective of MAP Decoding

- given an output ternary sequence produced by the EEC channel,

$$\mathbf{y} = y_1 y_2 \cdots y_M \in \{0, 1, \$\}^M$$

  infer the channel input sequence

$$\mathbf{x} = x_1 x_2 \cdots x_I \in C^I$$

  such that $|\mathbf{x}| = M$ and the a posteriori probability $P(\mathbf{x}|\mathbf{y})$ is maximized.

Bayes' Theorem:

$$P(\mathbf{x}|\mathbf{y}) = \frac{P(\mathbf{x}) P_e(\mathbf{y}|\mathbf{x})}{P(\mathbf{y})}.$$

Since $P(\mathbf{y})$ is fixed

$$\begin{aligned} P(\mathbf{x}) P_e(\mathbf{y}|\mathbf{x}) &= P(x_1) P_e(y(m_0, m_1]|x_1) \cdot \\ & \prod_{i=2}^{I} P(x_i|x_{i-1}) P_e(y(m_{i-1}, m_i]|x_i) \end{aligned}$$

Optimization Problem:

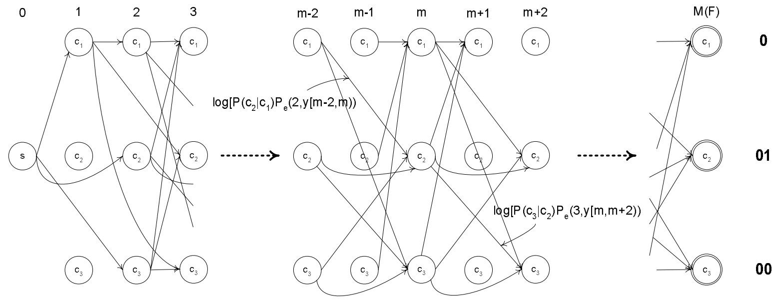
$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in C^*, |\mathbf{x}| = |\mathbf{y}|}{\arg\max} \{(\log P(x_1) + \log P_e(y(m_0, m_1]|x_1)$$
$$+ \textstyle\sum_{i=2}^{I} (\log P(x_i|x_{i-1}) + \log P_e(y(m_{i-1}, m_i]|x_i))\}.$$

Graph Representation

- weighted directed acyclic graph G with $NM + 1$ vertices, $M = |\mathbf{y}|$.
- a unique starting node s,
- other vertices grouped into M layers.
- each layer corresponds to a bit location in the received sequence $\mathbf{y}$.
- the nodes at the M-th (last) layer are so-called final nodes. Denote by F the set of all final nodes.
- n_i^m labels the i-th node at layer m, $1 \leq i \leq N$, which corresponds to codeword c_i parsed out of $\mathbf{y}$ at the m^{th} bit of $\mathbf{y}$.
- From n_j^m to $n_i^{m+|c_i|}$, $1 \leq m \leq M - |c_i|$, $1 \leq i, j \leq N$, there is an edge corresponding to decoding $y(m, m + |c_i|]$ as codeword c_i,

given that the previously decoded codeword is c_j.

- The weight of this edge is $\log P(c_i|c_j) + \log P_e(y(m, m+|c_i|]|c_i)$.
- Generally, for a node n_i^m, there are N incoming edges, one from each of the N nodes on layer $m-|c_i|$; there are N outgoing edges emitted from n_i^m, one to each of the nodes $n_j^{m+|c_j|}$, $1 \le j \le N$.
- Any input sequence $\mathbf{x}$ of $|\mathbf{y}|$ bits, can be mapped to a distinct path from s to F such that the weight of the path equals the value of the objective function in $\mathbf{x}$. Moreover, this mapping is one-to-one.
- The problem of MAP decoding is thus converted to finding the single-source longest path in the weighted directed acyclic graph G, from s to F.



Computing the Longest Path in G

- Let $\omega(m,i)$ be the weight of the longest path from s to the node n_i^m, then

$$\omega(m,i) = \max_{1\le j\le N}\{\omega(m-|c_i|,j) + \log P(c_i|c_j) + \log P_e(y(m-|c_i|,m]|c_i)\} \tag{5}$$

  for all $1 \le i \le N$ and $|c_i| \le m \le M$,
- with initial values

$$\omega(|c_i|,i) = \log P(c_i) + \log P_e(y(0,|c_i|]|c_i) \tag{6}$$

  and $\omega(m,i) = -\infty$ if $m < |c_i|$, $1 \le i \le N$.
- The MAP decoding is determined by

$$\tilde{\omega}(M) = \max_{1\le i\le N} \omega(M,i). \tag{7}$$

Dynamic Programming solution

- At each stage $m, 1 \le m \le M$, weights $\omega(m,i)$ are computed for $1 \le i \le N$, using (5).
- $\log P(c_i|c_j)$ and $\log P_e(y(m-|c_i|,m]|c_i)$ in (5) can be precomputed and stored in look-up tables so that they will be available to DP process in $O(1)$ time.
- The search in (5) takes $O(N)$ time for fixed m and i. Each stage is completed in $O(N^2)$ time and all the M stages in $O(N^2M)$ time. The step of (7) clearly takes $O(N)$ time. Therefore, the time complexity of this algorithm is $O(N^2M)$.

Complexity Reduction by Matrix Search

- Organize the computations in a different way: at each stage m compute the weights $\omega(m+|c_i|, i)$ for all $i, 1 \leq i \leq N$:

$$\begin{aligned}\omega(m+|c_i|,i) &= \max_{1\leq j\leq N}\{\omega(m,j)+\log P(c_i|c_j)+\\ &\quad \log P_e(y(m,m+|c_i|]|c_i)\},\end{aligned} \tag{8}$$

  for all $1 \leq i \leq N$ and $1 \leq m \leq M - |c_i|$.

- For each $1 \leq m \leq M - \max_{i,1\leq i\leq N}|c_i|$, consider the matrix G_m of dimension $N \times N$, with elements $G_m(i,j)$

$$\begin{aligned}G_m(i,j) &= \omega(m,j)+\log P(c_i|c_j)+\\ &\quad \log P_e(y(m,m+|c_i|]|c_i).\end{aligned} \tag{9}$$

- Then relation (8) is equivalent to

$$\omega(m+|c_i|,i) = \max_{1\leq j\leq N} G_m(i,j). \tag{10}$$

- Computing all $\omega(m+|c_i|,i)$ for given m and all $1 \leq i \leq N$, is equivalent to finding all row maxima of the matrix G_m.
- Straightforward solution: $O(N^2)$ time.
- If the matrix G_m is so-called totally monotone then the problem of row maxima can be solved in $O(N)$ time by a fast matrix search technique introduced by Aggarwal et al. in 1987 (Algorithm SMAWK).

Total monotonicity

- The matrix G_m is said to be totally monotone with respect to row maxima if the following relation holds:

$$\begin{aligned}G_m(i,j) \leq G_m(i,j') \Rightarrow G_m(i',j) \leq G_m(i',j'),\\ i<i', j<j'.\end{aligned} \tag{11}$$

- If all the matrices G_m are totally monotone, then the MAP decoding problem can be solved in $O(NM)$ time.

Algorithm SMAWK

- If a $k \times n$ matrix A with $k \leq n$, is totally monotone then the column index $j(i)$ corresponding to the maximum entry of row i, increases with i.
- If the maxima of all even rows are known, then the maxima of remaining odd rows can be computed in $O(n)$ time since for each odd row $2i+1$ the search is restricted to the interval between $j(2i)$ and $j(2i+2)$ and $\sum_{1\leq i<k/2}(j(2i+2)-j(2i)) = O(n)$.
- The elegant technique introduced by Aggarwal et al. in 1987 (SMAWK) can delete $n-k$ columns containing no row maxima of a $k \times n$ totally monotone matrix with $k < n$, in $O(n)$ time.
- The size of the matrix search problem can be reduced from $k \times n$ to $k \times k$ in $O(n)$ time. Then the $k \times k$ problem is reduced to $k/2 \times k/2$ in $O(k)$ time. The size of the new problem is further

reduced to $k/4 \times k/4$ in $O(k/2)$ time and so on.

- Let $T(k)$ be the time for solving the $k \times k$ problem and ck be the cost of the size reduction from $k \times k$ to $k/2 \times k/2$, then the following recurrence holds: $T(k) = T(k/2) + ck$, which clearly implies $T(k) = O(k)$.
- The solution of the $k \times n$ problem is obtained in $O(n)$ time.

Condition for total monotonicity

A sufficient condition for total monotonicity) is the Monge condition:

$$G_m(i,j') + G_m(i',j) \le G_m(i',j') + G_m(i,j), \quad i < i', j < j'. \tag{12}$$

which is equivalent to

$$\log P(c_i|c_{j'}) + \log P(c_{i'}|c_j) \le \log P(c_{i'}|c_{j'}) + \log P(c_i|c_j), \quad i < i', j < j'. \tag{13}$$

This condition does not depend either on the channel statistics or on the output sequence, but only on the source statistics. Therefore, the decoder can check if the condition holds before deciding whether to use the fast matrix search algorithm or the standard dynamic programming algorithm for MAP decoding.

Checking the Monge condition takes only $O(N^2)$ time. Indeed, in

order for (13) to hold, we only need the inequality

$$\log P(c_i|c_{j+1}) + \log P(c_{i+1}|c_j) \le \log P(c_{i+1}|c_{j+1}) + \log P(c_i|c_j) \tag{14}$$

to be valid for each pair $i, j, 1 \le i, j \le N$.

The fast matrix search approach can still be applied if there are two permutations ϕ and ψ on the integers between 1 and N, such that

$$\log P(c_{\phi(i)}|c_{\psi(j')}) + \log P(c_{\phi(i')}|c_{\psi(j)}) \le \log P(c_{\phi(i')}|c_{\psi(j')}) + \log P(c_{\phi(i)}|c_{\psi(j)}), \quad i < i', j < j'.$$

In this case the rows and columns of each matrix G_m have to be permuted by using ϕ, respectively ψ.

$O(N^2)$ time suffices to check if such permutations exist [Burkard et

al. 1996].

Markov Sources Satisfying the Monge Condition

Assume that the codewords c_i are the output of a scalar quantizer applied to a continuous Markov source.

Monge condition (13) is equivalent to

$$P(c_i|c_{j'})P(c_{i'}|c_j) \le P(c_{i'}|c_{j'})P(c_i|c_j) \quad i < i', j < j'. \tag{15}$$

Multiplying both sides by $P(c_j)P(c_{j'})$, we have

$$P(c_{j'}, c_i)P(c_j, c_{i'}) \le P(c_{j'}, c_{i'})P(c_j, c_i) \quad i < i', j < j'. \tag{16}$$

For each $i, 1 \le i \le N$, let S_i denote the quantization cell (interval) represented by the codeword c_i. Assume that for all $i < i'$ we have $u < v$ for all $u \in S_i$ and all $v \in S_{i'}$.

Relation (16) is equivalent to

$$\int_{S_{j'}}\int_{S_i} f(v', u)dudv' \int_{S_j}\int_{S_{i'}} f(v, u')du'dv \le \int_{S_{j'}}\int_{S_{i'}} f(v', u')du'dv' \int_{S_j}\int_{S_i} f(v, u)dudv, \tag{17}$$

further equivalent to

$$\int_{S_{j'}}\int_{S_i}\int_{S_j}\int_{S_{i'}} f(v', u)f(v, u')du'dvdudv' \le \int_{S_{j'}}\int_{S_i}\int_{S_j}\int_{S_{i'}} f(v', u')f(v, u)du'dvdudv'. \tag{18}$$

A sufficient condition for (18)

$$f(v', u)f(v, u') \le f(v', u')f(v, u), \tag{19}$$

or equivalently

$$\log f(v', u) + \log f(v, u') \le \log f(v', u') + \log f(v, u), \tag{20}$$

for any real values $u < u'$ and $v < v'$.

If the second partial derivative $\partial^2(\log f)/\partial u \partial v$ exists, then (20) holds iff $\partial^2(\log f)/\partial u \partial v \ge 0$ [Burkard et al. 1996].

Clearly $\partial^2(\log f)/\partial u \partial v \ge 0$ holds when the joint pdf $f(\cdot, \cdot)$ is Gaussian.

MAP Decoding with Length Constraint

- Assume that the number K of symbols of the input Markov sequence is known (transmitted reliably as side information).
- The objective of MAP decoding with length constraint is to find the Markov sequence $\mathbf{x}$ of exactly K symbols, of maximal a posterior probability $P(\mathbf{x}|\mathbf{y})$.
- The problem is equivalent to the maximum-weight K-link path in the graph G.
- Dynamic programming solution: $O(N^2M^2)$ time complexity [Park, Miller'98].

Technique Based on Parameterized Search

- For any real number τ, define a new weighted directed acyclic graph $G(\tau)$ 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 τ.
- The following results were proved in [Aggarwal, Schieber, and Tokuyama'94].
- **Lemma 1:** If for some real τ, the maximum-weight path in $G(\tau)$ has k edges, then this path is the maximum-weight k-link path in G.
- **Lemma 2:** Denote by $k(\tau)$ the number of edges in the maximum-weight path in $G(\tau)$. Then $k(\tau)$ is non-decreasing as τ increases.

Length-constrained MAP Decoding

- Find the maximum-weight path in $G(\tau)$ in conjunction with a binary search on τ until $k(\tau) = K$.
- No guarantee that a real value τ exists to satisfy $k(\tau) = K$. But in this case the algorithm will converge quickly to such a τ that $k(\tau) = K + \alpha$, where α is an integer whose absolute value is very small.
- To reduce the computational complexity, we limit the number of iterations in the binary search to be L, then the overall time complexity is $O(LMN^2)$.

Complexity Reduction by Matrix Search

The fast matrix search technique can be applied to find the longest path in $G(\tau)$ too.

The correspondent of matrix G_m is now the matrix $G_{m,\tau}$:

$$G_{m,\tau}(i,j) = \omega_\tau(m,j) + \log P(c_i|c_j) + \log P_e(y(m, m+|c_i|]|c_i) + \tau, \tag{21}$$

where $\omega_\tau(m,j)$ denotes the weight of the longest path from s to the node n_j^m in $G(\tau)$.

If the Monge condition:

$$G_{m,\tau}(i,j') + G_{m,\tau}(i',j) \leq G_{m,\tau}(i',j') + G_{m,\tau}(i,j), \quad i < i', j < j', \tag{22}$$

holds for all m, then the longest path in $G\tau$ can be found in $O(NM)$

time, leading to an $O(LMN)$ time algorithm for length-constrained MAP decoding.

The Monge condition (22) is equivalent to

$$\log P(c_i|c_{j'}) + \log P(c_{i'}|c_j) \leq \log P(c_{i'}|c_{j'}) + \log P(c_i|c_j) \quad i < i', j < j', \tag{23}$$

i.e., the same condition as for MAP decoding without length constraint.

Experimental Results

- Measurement
  - Symbol-by-symbol difference (PSNR)
  - Alignment with minimum Edit distance
  - Example

    $\mathbf{I}$: `121020010-2`

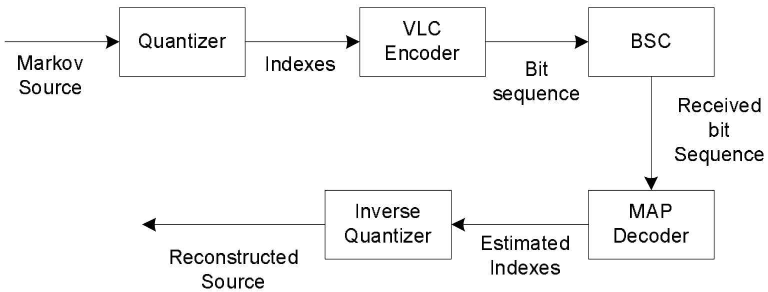
    $\hat{\mathbf{I}}$: `-2110001022`

    $\tilde{\mathbf{I}}$: `d--ss----i-`
  - $\hat{\mathbf{I}}$ is adjusted to $\tilde{\mathbf{I}} = \cdots s_i s_d s_j \cdots$;
    * s_i and s_j agree with $\mathbf{I}$ symbol-by-symbol;
    * s_d differs from $\mathbf{I}$ in all of its symbols.
- Mean error propagation length $\bar{e}_l$;
- Number of error propagation e_n.

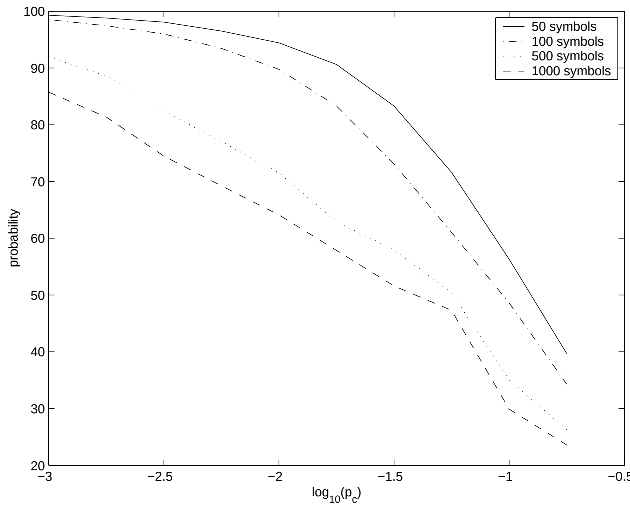
Experiment Configuration

- A zero-mean, unit-variance, first-order Gaussian-Markov process of correlation coeficient 0.9;
- Uniform scalar quantizer with 9 code cells;
- Sequences of different lengths $K = 50, 100, 500$ generated by the source model;
- Variable length encoded at average rate of about 3 bits per sample;
- Binary symmetric channel of various crossover probabilities.
- Averages of 1000 simulation;
- Comparison algorithms
  - M. Park and D. J. Miller - Approximate algorithm in [7];
  - Z. Wang and X. Wu - MAP without length constraint in [11].

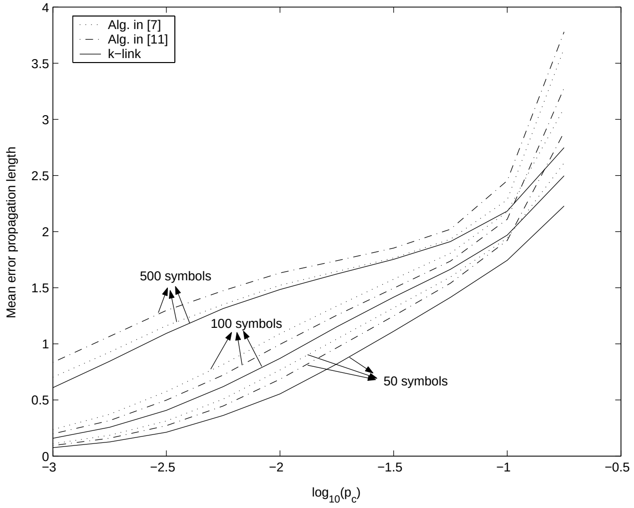
Channel model used in the experiment



Probability of Finding the Optimal Solution



Mean Error Propagation Length



Number of Error Propagations