exciting a pitch predictor synthesis structure with noise, uncorrelated between pitch cycles. Then, an LPAS coder with a closed-loop pitch predictor reconstructs this signal with the same periodicity as the original signal. However, for other signals, the periodicity of the reconstructed signal can increase or decrease. When the original signal consists of a periodic signal and additive noise, the periodicity of the reconstructed signal increases. This likely is relevant for voicing onsets and transitions. In a practical speech coder, other effects, such as the frequency dependence of the periodicity and the stepwise changes in the pitch period, also play a role in the periodicity of the reconstructed signal.


during can cause it to assign a positive probability to an observation not previously assigned a positive probability. This can result in very long execution times. The authors point out that additional savings in execution time can be achieved by precomputation.

1. BEHAVIOR OF THE ANY-PATH PROCEDURE

In [1], hidden Markov model (HMM) recognition is considered using a state-space-like formulation:

\[ x(t+1) = Ax(t) + u(t)\delta(t) \]  
\[ y(t) = Bx(t) \]

(1) (2)

where the \((i,j)\)th element of \(A\) is the probability of making a transition from state \(j\) to state \(i\) and the \((k,j)\)th element of \(B\) is the probability of observing output \(k\) from state \(j\). The following state-space transformation is used:

\[ x(t) = UPx(t) \]  
\[ y(t) = UBy(t) \]

(3) (4)

If the columns of \(P^{-1}\) are the normalized eigenvectors of \(A\) and if \(U\) is diagonal, then the new transition matrix, \(\tilde{A}\), is diagonal, consisting of the system’s eigenvalues:

\[ \tilde{A} = UP\Lambda U^{-1} \]  
\[ \tilde{B} = BP^{-1}U^{-1} \]

(5) (6) (7)

If the diagonal matrix, \(U\), is chosen such that \(\text{diagonal}(U^{-1}) = P\), then \(\hat{w}(0)\) will consist of all ones.

Let \(N\) denote the number of states and \(T\) the number of observations. At time \(t\), the probability of the current output symbol, \(O_t\), is taken as the dot product of \(x(t)\) and the \(k\)th row of \(B\), where \(k = O_t\):

\[ p(x(t) | O_1, \ldots, O_T) = \prod_{j=1}^{T} p(x(t) | O_j) \]

The entire observation sequence is taken into account by forming the products of the individual dot products:

\[ \text{prob}(O_1, \ldots, O_T | \text{model}) = \prod_{t=1}^{T} p(x(t)), \]

(9)

This formulation assumes that the observations are unconditionally independent. By contrast, both the forward algorithm and the Viterbi algorithm assume that the observations are independent only if the state path is known. The incorrect assumption of the any-path procedure can cause it to assign a positive probability to an observation not previously assigned a positive probability.

Manuscript received January 29, 1994; revised May 5, 1994. The associate editor coordinating the review of this paper and approving it for publication was Dr. Xuedong Huang.

The authors are with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907-1285 USA.

IEEE Log Number 9403975.

A and \(B\) are defined here as in [1], but these definitions differ from standard HMM notation by a transpose.

1
sequence that is impossible according to the HMM paradigm. This behavior is illustrated with a simple example. Consider the following observation sequence, HMM parameters, and diagonalized model

\[
O = \{2 \ 1 \ 3 \ 2\}
\]

\[
A = \begin{bmatrix}
0.0 & 0.0 \\
0.6 & 0.0
\end{bmatrix}, \quad B = \begin{bmatrix}
0.3 & 1.0 \\
0.7 & 0.1
\end{bmatrix}, \quad \pi(0) = 0.5
\]

\[
P^{-1} = \begin{bmatrix}
1.0 & -0.7071 \\
0.0 & 0.7071
\end{bmatrix}, \quad U = \begin{bmatrix}
1.0 & 0.0 \\
0.0 & 1.4142
\end{bmatrix}
\]

\[
A = \begin{bmatrix}
0.0 & 0.45 \\
0.4 & -0.15
\end{bmatrix}, \quad B = \begin{bmatrix}
1.6 & -0.3 \\
0.0 & 1.0
\end{bmatrix}, \quad \pi(0) = 1.0
\]

Using 1, 2, and 10, the any-path procedure without diagonalization yields the following results

\[
x(t) = \begin{bmatrix}
0.5000 \\
0.3500 \\
0.2450 \\
0.1715 \\
0.1200
\end{bmatrix}
\]

\[
yo_0(t) = \begin{bmatrix}
0.5000 \\
0.3500 \\
0.2450 \\
0.1715 \\
0.1200
\end{bmatrix}
\]

\[
prob(O_1, \ldots, O_T \mid \text{model}) = \prod_{i=1}^{T} yo_0(t) = 1.6377 \times 10^{-4}
\]

Using 3, 4, and 12, the diagonalized version of the any-path procedure leads to the same likelihood

\[
x(t) = \begin{bmatrix}
1.0000 \\
0.7000 \\
0.4900 \\
0.3430 \\
0.2401
\end{bmatrix}
\]

\[
yo_0(t) = \begin{bmatrix}
1.0000 \\
0.7000 \\
0.4900 \\
0.3430 \\
0.2401
\end{bmatrix}
\]

\[
prob(O_1, \ldots, O_T \mid \text{model}) = \prod_{i=1}^{T} yo_0(t) = 1.6377 \times 10^{-4}
\]

In the above example, the any-path likelihood is 1.6377 \times 10^{-4}, even though the given observation sequence cannot be produced by the specified model. To see this, note that the first observation (i.e., \(O_1 = 2\)) is impossible in state 2, that state 1 can never make a transition to state 2, and that the second observation (i.e., \(O_2 = 1\)) is only possible in state 2 (see (10)). Both the forward and Viterbi algorithms would yield a zero probability for this example.

Although assigning a nonzero probability to an impossible observation sequence is an extreme example, it is important to note that the any-path procedure will lead to incorrect HMM probabilities in nearly all cases. However, since the observation probabilities typically have a much larger impact on likelihood than the transition probabilities in HMM recognition, the modeling errors may be acceptable for some applications.

II. FURTHER REDUCTION IN RECOGNITION TIME BY PRECOMPUTING

Even though the any-path method suffers the drawback described above, this procedure may be chosen because it has complexity \(O(NT)\) on a fully connected model, compared to \(O(N^2 T)\) for the forward or Viterbi algorithms. Deller and Snider describe how to reduce computation in the diagonalized any-path procedure by merging HMM states with similar eigenvalues. If \(k\) is the fraction of states that are eliminated after merging, then the complexity of the diagonalized any-path procedure for the reduced state model is \(O\left((1 - k)NT\right)\).

Table I shows the number of multiplications for the three variations of the any-path procedure discussed in [1]. The number of states

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Transition Eq. 1 or 3</th>
<th>Output Eq. 8 and 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any-Path</td>
<td>(NT)</td>
<td>(NT + T)</td>
</tr>
<tr>
<td>Diagonalized Any-Path</td>
<td>(NT)</td>
<td>(NT + T)</td>
</tr>
<tr>
<td>Reduced States</td>
<td>((1 - k)NT)</td>
<td>((1 - k)NT + T)</td>
</tr>
</tbody>
</table>

is denoted by \(N\), and \(T\) is the number of output symbols in the observation sequence. The average number of predecessors, \(M\), would be equal to \(N\) for a fully connected model or 3 for the Bakis model used in [1].

Deller and Snider [1] point out that diagonalization can reduce computation of the any-path procedure by a factor of three for a Bakis model \((M = 3)\) before states are merged. We note that since the observations are unconditionally independent, the transition trellises (e.g., the matrix in (13)) can be precomputed using (1). For the 79 word task given in [1], storing the precomputed transition trellises as 64-bit floating point numbers would require less than one megabyte of memory, assuming the maximum observation length, \(T_{max}\), is less than 275 (i.e., \(275 \times 79 \times 6 \times 8 = 1,042,800\) bytes). With precomputation, the computations represented by the middle column of Table I are performed only once. The cost of recognition is dominated by the computations in the last column, i.e., the cost of multiplying the precomputed trellis by the appropriate output probabilities (i.e., (8)) and forming the product (i.e., (9)).

REFERENCES