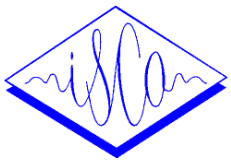


2nd European Conference on
Speech Communication and Technology
EUROSPEECH '91
Genova, Italy, September 24-26, 1991

A MATRIX REPRESENTATION OF HMM-BASED SPEECH RECOGNITION ALGORITHMS

Shigeki Sagayama

ATR Interpreting Telephony Research Laboratories
Sanpeidani, Inuidani, Seika-cho, Soraku-gun, Kyoto 619-02, Japan

ABSTRACT

This paper describes a matrix representation from which we can derive a new formulation of HMM-based speech recognition algorithms. This idea provides not only an alternative mathematical formulation equivalent to conventional trellis and Viterbi algorithms but also better understanding of HMM algorithms under grammatical constraints as well as more efficient computational possibilities.

In this formulation, a *likelihood matrix* is defined by an $(N+1) \times (N+1)$ dimensional upper triangular matrix whose (t,s) component is the observation likelihood of the given signal in a time span between $t+1$ and s. First, it is shown that the likelihood matrix for a pair of serially connected signal sources is the product of matrices $(P = P_1 P_2)$ and the parallel connection is represented by the sum $(P = P_1 + P_2)$.

From these basic properties, matrix-based HMM computation algorithms are derived. Explicit duration control at all levels, such as state, phoneme, syllable, and word, can be easily done. Grammatical rewriting rules are directly interpreted as matrix operations. A matrix parser is suggested for generalization of a CYK parser. This algorithm is particularly effective in large vocabulary systems where same phone units (phonemes) appear in many syntactic paths.

1 Introduction

This paper describes a matrix representation of HMM-based speech recognition algorithms.

Although trellis and Viterbi algorithms are currently popular in HMM-based speech recognition, and are regarded as computationally efficient, they have difficulties in explicit duration control at the state, phoneme, and word levels and are sometimes inefficient in large vocabulary systems with grammatical constraints.

This paper introduces a new formulation of HMM likelihood computation using a matrix representation which provides not only efficient alternative algorithms equivalent to conventional trellis and Viterbi algorithms, but also a better understanding of HMM algorithms under grammatical constraints.

2 Matrix Representation of Stochastic Signal Sources

2.1 Signal sources and likelihood matrices

Denote an observed signal sequence by $\boldsymbol{x} = \{x_1, x_2, x_3, \cdots, x_N\}$. Here, the word "signal" has a wide interpretation. It can be a parameter vector sequence, such as a speech LPC cepstrum vector sequence, or a symbol sequence, such as a sequence of vector quantization codes.

Let us now consider a stochastic signal source, which is not necessarily stationary, and represent it by a set of parameters Θ. For example, an HMM such as shown in Fig.1 consists of multiple states and can be considered as a non-stationary signal source. An HMM word model can be also regarded as a signal source in this sense.

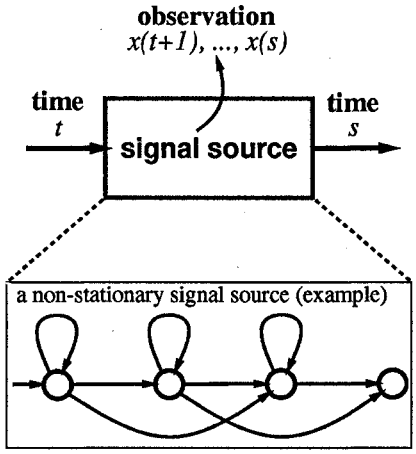


Figure 1: A non-stationary signal source and signal generation

Now, let us define the likelihood of observing a partial sequence of the signal, $\{x_{t+1}, x_{t+2}, x_{t+3}, \cdots, x_s\}$, $t < s$, from the given signal source model by $p_{jk}(\Theta | x_{t+1}, x_{t+1}, x_{t+3}, \cdots, x_s)$ or $p(t,s)$ for simplicity. Then,

Definition. A likelihood matrix of a signal source to produce the observed signal $\boldsymbol{x} = \{x_1, x_2, x_3, x_4, x_5 \cdots, x_N\}$ is defined by an triangular matrix

$$P = \begin{pmatrix} p(0,0) & p(0,1) & p(0,2) & \cdots & p(0,N) \\ & p(1,1) & p(1,2) & \cdots & p(1,N) \\ & & p(2,2) & \cdots & p(2,N) \\ & & & \ddots & \vdots \\ 0 & & & & p(N,N) \end{pmatrix} \quad (1)$$

where the (t,s) component is the likelihood of a probabilistic signal source model Θ that a partial sequence $\{x_{t+1}, x_{t+2}, x_{t+3}, \cdots, x_s\}$ is observed:

$$p(t,s) = p(\Theta | x_{t+1}, x_{t+2}, x_{t+3}, \cdots, x_s). \quad (2)$$

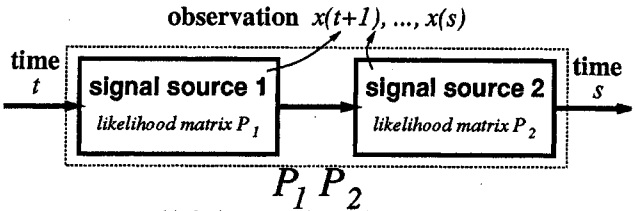
Usually, diagonal components are all zero except for a null transition which is represented by a unit likelihood matrix because it is regarded as a signal source which emits no output and consumes no time clock.

The likelihood of observing the whole sequence $\mathbf{x}$ is given by $L = \boldsymbol{b}^T P \boldsymbol{e}$ when

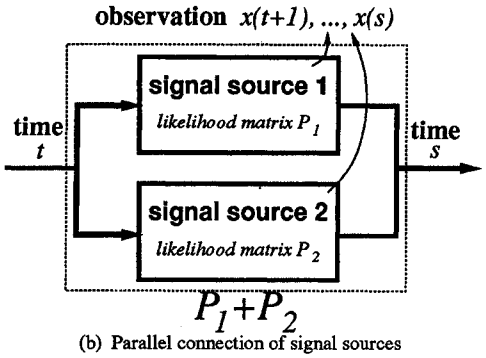
$$\boldsymbol{b}^T = (1, 0, \cdots, 0) \quad \text{and} \quad \boldsymbol{e}^T = (0, \cdots, 0, 1) \quad (3)$$

More generally, the likelihood of observing $\mathbf{x}$ between beginning and ending times which probabilistically distribute according to the $(N+1)$-dimensional probability vectors $\boldsymbol{b}$ and $\boldsymbol{e}$ is:

$$L = \boldsymbol{b}^T P \boldsymbol{e}. \quad (4)$$



(a) Series connection of signal sources



(b) Parallel connection of signal sources

Figure 2: Series and parallel connections of signal sources

2.2 Serial and parallel connections of signal sources

If P_1 and P_2 represent the likelihood matrices of two different signal sources for a signal sequence $x_{t+1}, x_{t+2}, \cdots, x_s$, the series connection of these sources, as shown in Fig. 2(a), yields a matrix product, because the likelihood of entering the first source at time t, and exiting at time s, from the second source with arbitrary transition time τ between t and s $(t < \tau < s)$ is

$$p(t,s) = \sum_{\tau=t+1}^{s-1} p_1(t,\tau)p_2(\tau,s) = \sum_{\tau=0}^{N} p_1(t,\tau)p_2(\tau,s) \qquad (5)$$

which is the (t,s) component of the composite matrix $P = P_1P_2$. Thus,

Serial connection. The likelihood matrix of the series connection of two signal sources, represented individually by likelihood matrices P_1 and P_2, is given by their product:

$$P = P_1P_2 \qquad (6)$$

For example, if likelihood matrices for phonemes /a/ and /i/ are represented by P_1 and P_2, the likelihood matrix for a sequence of phoneme pairs /ai/ is given by P_1P_2.

Since a component of the matrix representation of a parallel connection of two singnal source models is given by

$$p(t,s) = p_1(t,s) + p_2(t,s), \qquad (7)$$

Parallel connection. The likelihood matrix of the parallel connection of two signal sources, represented by likelihood matrices P_1 and P_2, is given by their sum:

$$P = P_1 + P_2 \qquad (8)$$

For example, if the likelihood matrices for allophones of the same phoneme /a/ are represented by P_1 and P_2, the likelihood matrix for both possibilities is given by $P_1 + P_2$.

Fig.3 shows an example of a network which includes both serial and parallel connections. If the likelihood matrices are given as in the figure, the matrix for the total network is given by

$$P = (P_1 + P_4)P_2P_3 + (P_1 + P_4P_5)P_6. \qquad (9)$$

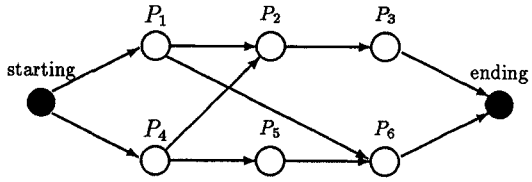


Figure 3: Serial and parallel connections of signal sources

2.3 Matrix representation of the Viterbi algorithm

Speech recognition is often formulated as an optimual search problem, where likelihood values for different paths are required instead of their sum.

Viterbi-type optimal path search, instead of a trellis-type algorithm, can also be defined by a matrix representation, where the multiplication in Eq.6 is defined by selection of the maximum product, namely:

$$p(t,s) = \max_{t<\tau<s} p_1(t,\tau)p_2(\tau,s) \qquad (10)$$

and the addition in Eq.8 is defined by choosing the larger:

$$P(t,s) = \max(p_1(t,s), p_2(t,s)). \qquad (11)$$

Needless to say, "book-keeping" is necessary for Viterbi path search by tracing back the optimal path in practical speech recognition applications.

3 Matrix Representation of HMM

3.1 Likelihood matrix of HMM state transitions

Let us consider a particular case, a state in an HMM, which is also a signal source. The likelihood matrix of an HMM state for a single clock has non-zero components only in the upper off-diagonal components:

HMM state transition. The likelihood matrix of a transition from state i to j of an HMM to observe x_t is

$$Q_{ij}(\mathrm{x}) = a_{ij} \begin{pmatrix} 0 & b_{ij}(x_1) & & & 0 \\ & 0 & b_{ij}(x_2) & & \\ & & \ddots & \ddots & \\ & & & 0 & b_{ij}(x_N) \\ 0 & & & & 0 \end{pmatrix} \qquad (12)$$

where a_{ij} is the transition probability and $b_{ij}(x_t)$ is the output probability.

In the case of self loops $(i = j)$, the likelihood matrix of staying at state i for one clock time is given by Q_{ii} letting $i = j$ in the above matrix. From the formulation of series connection, the likelihood matrix of staying at i for two clocks is Q_{ii}^2, which has non-zero values only at the second upper off-diagonal components. Generally, the likelihood matrix of staying at state i for k clocks is given by Q_{ii}^k. Therefore, if the maximum duration of the observed signal is n clocks, the likelihood of staying at i for arbitrary time is given by the triangular matrix:

$$P_{ii} = \sum_{k=1}^{N} Q_{ii}^k$$

$$= \begin{pmatrix} 0 & a_{ii}b_{ii}(x_1) & a_{ii}^2 b_{ii}(x_1)b_{ii}(x_2) & \cdots & a_{ii}^N b_{ii}(x_1)\cdots b_{ii}(x_N) \\ & 0 & a_{ii}b_{ii}(x_2) & \cdots & a_{ii}^{N-1} b_{ii}(x_2)\cdots b_{ii}(x_N) \\ & & 0 & \cdots & a_{ii}^{N-2} b_{ii}(x_3)\cdots b_{ii}(x_N) \\ & & & \ddots & \vdots \\ & & & & a_{ii}b_{ii}(x_N) \\ 0 & & & & 0 \end{pmatrix} \tag{13}$$

In this computation, $a_{ii}b_{ii}(x_t)$ is computed only once for $t = 1, 2, \cdots, n$ because other components are computed by matrix multiplications.

For example, if an HMM (4-state 3-loop) is defined by the likelihood matrices P_{11}, P_{22}, P_{33} of self loops at states 1, 2, and 3 and the likelihood matrices P_{12}, P_{23}, P_{34} of inter-state transitions, the total likelihood matrix P is computed by

$$P = P_{11}P_{12}P_{22}P_{23}P_{33}P_{34}. \tag{14}$$

If self loops and transitions are "tied", then P_{12}, P_{23}, and P_{34} are identical to P_{11}, P_{22}, and P_{33} except for constant factors.

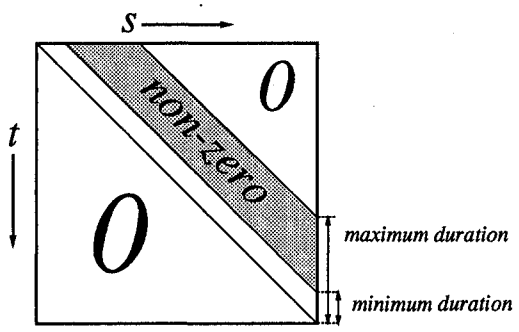


Figure 4: A likelihood matrix with minimum and maximum duration constraints (only shaded components are not zero)

4 Matrix Representatin of Speech Recognition Algorithms

4.1 Durational constraints of matrices

As signal sources often have durational contraints, namely, minimum and maximum durations, they often have non-zero likelihood matrix components only along a band with limited width, as shown in Fig.3.1. This means advantages not only in the computational and memory requirements, but also in explicit duration control at each level, i.e., HMM state, phoneme, syllable, word, and phrase.

4.2 Matrix representation of rewriting rules

A grammar is often defined by a set of rewriting rules. One of the important basic properties of the matrix representation is as follows, where '|' means a logical "or":

Rewriting rules. Rewriting rules:

$$A \rightarrow B\,C, \quad X \rightarrow Y \mid Z \tag{15}$$

correspond to matrix representations

$$P_A = P_B P_C, \quad P_X = P_Y + P_Z. \tag{16}$$

where P_α denotes the likelihood matrix of a symbol α.

Because, in usual cases, only the optimal path is accepted as the recognition result, the matrix computation must be of Viterbi-type (Eq.10).

This algorithm can be regarded as a multi-stage DP algorithm which is an extension of two-stage [3][4] and three-stage [6] DP algorithms. For example, the first stage may be phonemes followed by syllables, words, phrases, and sentences.

4.3 Sentence likelihood in matrix formulation

Since, as stated before, the series connection of signal sources is represented by a product of likelihood matrices, the likehood of a series of grammatical symbols $1, 2, 3, \cdots, n$ for the whole signal time span is given by

$$L = \boldsymbol{b}^T P_1 P_2 P_3 \cdots P_n \boldsymbol{e} \tag{17}$$

where $\boldsymbol{b}$ and $\boldsymbol{e}$ are $(N+1)$-dimensional vectors specifying the starting and ending times as, defined by Eq.3. If a grammatical constraint is provided by a state transition network, the optimal path is found, in principle, by examining all possible paths to find the maximum likelihood.

By matrix computation, multiple paths are simultaneously computed if a Viterbi-type matrix definition (Eq.10) is used. In this case, it is possible to compute likelihood matrices for grammatical units such as phonemes or words in trellis-type definition of matrices and search for the optimal path in a Viterbi sense.

4.4 A consideration on word spotting

A relatively large component in a phoneme likelihood matrix suggests phoneme spotting. The same principle applies to a word likelihood matrix which is obtained by multiplication of phoneme likelihood matrices.

Conversely, a word spotting result derived from some other technique (such as word template matching) can be regarded as a likelihood matrix with only one non-zero component. Multiple word spotting results correspond to a sparse matrix. Thus, likelihood matrices can be constructed by manners other than matrix sums and products according to a grammatical structure.

The island-driven approach is one possible solution which can avoid the computational explosion at the beginning of speech caused by ambiguous utterances. In the matrix representation, significant matrix components in phoneme likelihood matrices correspond to "islands", from which a grammatical search may be started.

4.5 A matrix parser

The CYK (Cocke-Younger-Kasami) parser is considered as a special case of this matrix approach where the analyzing table is regarded as an upper triangular matrix with components of only 0 or 1 instead of probabilistic values. This idea leads to a generalized form of the CYK algorithm, where matrix components $p(t, s)$ are the likelihood of the model gives the observation sequence between $t+1$ and s. This matrix representation can handle with probabilistic weights according to word attributes, a priori word occurance probabilities, and probabilistic rewriting rules.

5 Computational Aspects of the Matrix Representation

5.1 Time-synchronous computation of likelihood matrices

Although the matrix computation basically can be done only after the whole observation $\boldsymbol{x}$ is obtained, time-synchronous computation is also possible. Time-synchronous matrix computation for an HMM state proceeds as:

$$\begin{pmatrix} 0 & p(0,1) \\ 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & p(0,1) & p(0,2) \\ 0 & 0 & p(1,2) \\ 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & p(0,1) & p(0,2) & p(0,3) \\ 0 & 0 & p(1,2) & p(1,3) \\ 0 & 0 & 0 & p(2,3) \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdots \tag{18}$$

where, at time s, only the last column needs to be computed. Particularly if the signal source is stationary, as in HMM self-loops, all of the

components $p(t,s)$ $(t=1,2,\cdots,s-2)$ are derived from a component $p(s-1,s)$ by

$$p(t,s) = p(t,s-1)p(s-1,s), \quad t=1,2,3,\cdots,s-2. \tag{19}$$

Similarly, if P_2 represents a stationary signal source such as an HMM self-loop, the time-synchronous computation of the matrix product $P = P_1P_2$ is given by the recursive formula:

$$\begin{aligned} p(t,s) &= \sum_{\tau=t+1}^{s-1} p_1(t,\tau)p_2(\tau,s) \\ &= \sum_{\tau=t+1}^{s-2} p_1(t,\tau)p_2(\tau,s-1)p_2(s-1,s) + p_1(t,s-1)p_2(s-1,s) \\ &= \{p(t,s-1)+p_1(t,s-1)\}\, p_2(s-1,s), \; t=1,2,\cdots,s-1. \end{aligned} \tag{20}$$

The matrix sum, $P = P_1 + P_2$, is simply given by:

$$p(t,s) = p_1(t,s) + p_2(t,s), \quad t=1,2,3,\cdots,s-1. \tag{21}$$

Note that these formullae are essentially equivalent to trellis algorithms for series and parallel HMM state transition networks. This means the trellis computation has been derived from the matrix formulation without consideration of any graph structure such as a trellis.

5.2 Computation by a matrix-specific computer

The computation for a self loop of a state in an HMM consists of $(n-1)$ computations of the output probability $b_{ii}(x_t)$

If matrix arithmetic operations run very fast on a computer, P is obtained through $(2\log_2(n+1))$ matrix operations. An example of a 15-dimensional case shown is below

$$\begin{aligned} Q_2 &= Q^2 \\ Q_4 &= Q_2^2 \\ Q_8 &= Q_4^2 \\ Q_{0,4,8,12} &= (E+Q_4)(E+Q_8) \\ Q_{0,2,4,6,8,10,12,14} &= (E+Q_2)Q_{0,4,8,12} \\ P &= (E+Q)Q_{0,2,4,6,8,10,12,14} - E. \end{aligned} \tag{22}$$

The matrix dimension is $(N+1)$, when the length of the input signal is N, and this implies that the matrix computational cost becomes huge as the signal length increases. One possible solution is decimation in time[2]. Time division can be coarse when the signal is near stationary or fine when the signal is transient. Such non-uniform time decimation is advantageous in reducing the computational cost, without loss of significant information.

6 Relationship with Markov Chains

Another, but similar, formulation of the likelihood matrix has been given[5] in terms of a time synchronous state transition likelihood matrix $A(x_t)$ whose (i,j) component is the likelihood of transition from state i to state j when a sample x_t is observed. If the initial and final state probabilities are given by $\boldsymbol{i}$ and $\boldsymbol{f}$, the total likelihood of observing $\boldsymbol{x} = \{x_1, x_2, \cdots, x_N\}$ is given by $L = \boldsymbol{i}^T A \boldsymbol{f}$. This quadratic form is similar to that of the present matrix formulation.

The difference between the two matrix representations is interpreted as follows. In the matrix representation presented in this paper, matrices are computed for each of the grammatical units (phonemes, words, etc.) and the matrix dimension is the total time duration of the observed signal. In contrast, in the Markov chain representation, likelihood matrices are computed for all observed signal points $\{x_t\}$ and the matrix dimension is the number of states contained in the model. From this reason, there is a duality between these different matrix formulations.

7 Conclusion

Matrix representations for general signal sources, HMM's, and speech recognition algorithms have been described. The most important point of this work is its simple and straightforward formulation. Although the computational advantage of the matrix formulation is still not clear, as it seems to depend on vocabulary size and the grammatical complexity, it has the advantage of being able to easily incorporate such things as duration control at all grammatical levels, word spotting, and matrix parsing.

8 Acknowledgement

The initial stages of this work were done at NTT Human Interface Laboratories[1]. The author wish to thank S. Matsunaga and members of the speech group both at NTT and ATR for their fruitful discussions.

9 References

References

[1] S.Sagayama, S.Matsunaga, and S. Honma, "Phrase sentence speech recognition system based on phonetic HMM," Nat. Conv. IECE Japan, A-19, pp.1-20-21 (1989).

[2] S.Matsunaga and S.Sagayama, "Minimal phrase recognition using bidirectional parsing and effect of decomation in time," Tech. Rep. IECE Japan, SP89-16, (1989-6).

[3] H. Sakoe, "Two-level DP-matching – a dynamic programming based pattern matching algorithm for connected word recognitions," IEEE Trans. Acoust., Speech & Signal Process., ASSP-27, 6, pp. 588-595 (1979).

[4] H. Sakoe, "A generalized two-level DP-matching for continuous speech recognition," Trans. IECE Japan, E65, 11, pp.649-656 (Nov. 1982).

[5] K. S. Fu, *Syntactic Methods in Pattern Recognition*, Academic Press, 1974.

[6] E. Tsuboka, "A spoken word recognition method for large vocabulary based on syllable recognition," Trans. Committee on Speech Res. S84-68 (1984-12).