The Moments of Matched and Mismatched Hidden Markov Models

ROY L. STREIT, SENIOR MEMBER, IEEE

Abstract—**An algorithm for computing the moments of matched and mismatched hidden Markov models from their defining parameters is presented. The algorithm is of general interest because it is an extension of the usual forward-backward linear recursion. The algorithm computes the joint moments of the posterior likelihood functions (i.e., the scores) by a multilinear recursion involving the joint moments of the random variables associated with the hidden states of the Markov chain. Examples comparing the first two theoretical moments to simulation results are presented. They are of independent interest because they indicate that the distribution of the posterior likelihood function scores for matched and mismatched models are asymptotically log-normal in important special cases and, therefore, are characterized asymptotically by the first two moments alone. One example discusses the effect of a noisy discrete communication channel on a suboptimal classification method based on the distributions of scores rather than on maximum likelihood classification.**

I. Introduction

HIDDEN Markov models (HMM's) are statistical models that are developed in diverse applications to characterize different classes of nonstationary time series or signals. Subsequently, HMM's are utilized for the automatic classification of an unknown signal into one of these signal classes. In speech applications, they are used to characterize the time variation of the short-term spectra of spoken words. An example is the speaker-independent isolated word recognition (SIIWR) problem where HMM's characterize the words (or parts of words) in a finite size vocabulary. Different words are characterized by different HMM's [1]. In target tracking applications, HMM's are used to characterize the time variation of a target track measurement sequence. A specific example is the narrow-band frequency line tracking problem where HMM's characterize possible target frequency shifts as well as noise in the measurement sequence for finite signal-to-noise ratio (SNR). Different HMM's characterize different target track dynamics and different SNR's [8]. A brief description of the mathematical structure of HMM's is given at the beginning of Section II.

An application-specific preprocessor is critical to the successful use of HMM's in the application. This preprocessor maps (or transforms) an arbitrary input signal $s(t)$, $t \geq 0$ into a discrete observation sequence $\{O(t), t = 1, 2, \cdots\}$. Reference [1, pp. 1077–1078] gives a description of one such preprocessor for the SIIWR problem, and [8] describes one suitable for frequency line tracking. Throughout this paper, it is assumed that a satisfactory preprocessor is available, but no assumptions are made about its specific nature. The output of the preprocessor constitutes the observation sequence. In practice, this sequence is truncated to have finite length T where T is selected according to the application needs. The truncated sequence is denoted by $O_T = \{O(t), t = 1, 2, \cdots, T\}$.

The act of computing specific numerical values for the various parameters of an HMM is called "training." Training takes place on the outputs of the preprocessor when it is given multiple realizations of a specific signal class. If the Baum–Welch reestimation algorithm is used for training, then training is equivalent to solving a mathematical optimization problem to determine maximum likelihood estimates of the HMM parameters [2]. In this paper, it is assumed that the training phase is completed and that the HMM's developed are adequate models for each of the signal classes of interest (e.g., the vocabulary words in the SIIWR problem or the target/SNR characteristics in the tracking application). We denote by HMM(i) the HMM parameter set defining the ith signal class. An important consequence of these training assumptions is that HMM(i) can be used as a synthetic signal source, that is, HMM(i) can be used to simulate the output of the preprocessor when the ith signal is input to it. We use the notation $O_T \in \text{HMM}(i)$ to mean that the observation sequence O_T is a realization of a random vector whose statistical distribution is defined implicitly by HMM(i).

HMM's are used for classification of an unknown observation sequence O_T by exploiting a probability measure or posterior likelihood function, as depicted in Fig. 1. The posterior likelihood function is defined on the set of all truncated sequences $\{O_T\}$ by utilizing the mathematical structure of HMM's. Thus, the likelihood of a given O_T depends critically on the numerical values of the parameters defining the underlying HMM. The ith HMM recognizer computes the posterior likelihood function $f_i(O_T)$. If HMM(i) is a finite symbol HMM (see Section II below), then $f_i(O_T)$ is equivalent to a probability, that is,

$$f_i(O_T) = \Pr\left[O_T \middle| \text{HMM}(i)\right], \quad i = 1, \cdots, p. \quad (1)$$

The maximum of the computed likelihoods identifies or classifies the original signal $s(t)$ that was input to the pre-

Manuscript received July 11, 1987; revised January 3, 1989.

The author is with the Naval Underwater Systems Center, New London, CT 06320.

IEEE Log Number 8934008.

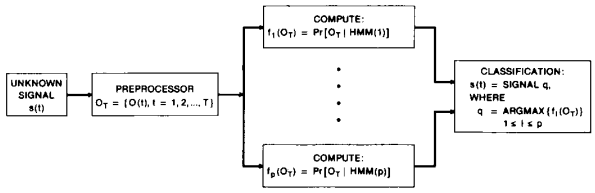


Fig. 1. Classification of unknown signal $s(t)$ as one of p signals for which trained HMM's are available.

processor. It is well known that this classifier is optimum in the Neyman–Pearson sense; that is, for a specified probability of incorrect classification, the probability of correct classification is a maximum [3].

The fundamental problem studied in this paper is the determination of the probability density function (pdf) of the test statistic $f_i(O_T)$ when $O_T \in \text{HMM}(j)$. In other words, if O_T is a random vector generated by HMM(j), what is the pdf of the numerical values of the ith posterior likelihood function $f_i(O_T)$? Note that the HMM's are matched if $i = j$ and mismatched if $i \neq j$. This paper presents an algorithm for computing explicitly the moments of the desired pdf up to any required order directly from the underlying parameters of the HMM's involved, and presents examples that compare the first two theoretical moments to simulation results. The algorithm is of general interest because it is an extension of the usual forward–backward linear recursion [2] for HMM's. It computes the joint moments of the likelihood functions $f_i(O_T)$ by a multilinear recursion involving the joint moments of the random (observation) variables uniquely associated with the hidden states of the HMM's. The examples are of independent interest as well. First, they indicate that the desired pdf is asymptotically log-normal in important special cases and, therefore, is completely characterized (asymptotically) by the first two moments alone. It is not obvious how the central limit theorem can be used to account for this result. Second, the examples show that a suboptimal classification method using preset detection thresholds for the likelihood functions $f_i(O_T)$ may be useful in certain instances. This point is discussed at the end of this section.

The distribution we seek is defined via its cumulative distribution function (cdf), denoted by $F_{ij}(x)$. It is intuitively appealing to attempt to define $F_{ij}(x)$ by setting

$$F_{ij}(x) = \Pr\left[f_i(O_T) < x \quad \text{and} \quad O_T \in \text{HMM}(j)\right]$$

where x is any real number; however, such a definition is ambiguous because the meaning of the probability measure $\Pr[\cdot]$ is unclear. Instead, for finite symbol HMM's, we define

$$F_{ij}(x) = \sum_{O_T} H(x - f_i(O_T)) f_j(O_T) \tag{2}$$

where the function $H(\cdot)$ is defined by

$$H(x) = 1 \qquad \text{if } x \geq 0$$
$$H(x) = 0 \qquad \text{if } x < 0.$$

From (2), it is clear that $F_{ij}(x)$ is a cdf because it is a nonnegative increasing right-continuous function, and the limit of $F_{ij}(x)$ is 0 as x goes to 0^- and 1 as x goes to $+\infty$. For continuous symbol HMM's, the summation over O_T in (2) must be replaced by integration over O_T. Algorithms that calculate F_{ij} directly from the HMM parameters are not known. For later reference, note that, in general, $F_{ij}(x) \neq F_{ji}(x)$.

The moments of $dF_{ij}(x)$ are defined by the Riemann–Stieltjes integral

$$M_{ij}(k, T) = \int_{-\infty}^{\infty} x^k dF_{ij}(x), \qquad k = 0, 1, 2, \cdots. \tag{3}$$

If $F_{ij}(x)$ is differentiable with derivative $F'_{ij}(x)$, then the moments can be written equivalently as the Riemann integral

$$M_{ij}(k, T) = \int_{-\infty}^{\infty} x^k F'_{ij}(x)\, dx.$$

The moments depend on the length T of the observation sequence because $F_{ij}(x)$ depends on T, as seen from (2). They uniquely determine $dF_{ij}(x)$ when they are all finite and the characteristic function of $dF_{ij}(x)$ has a finite radius of convergence [4]. For finite symbol HMM's, it is clear from (1) and (2) that $dF_{ij}(x) = 0$ for $x < 0$ and $x > 1$. Thus,

$$M_{ij}(k, T) = \int_0^1 x^k dF_{ij}(x) \leq 1$$

so that all the moments are finite. The series

$$\phi_{ij}(w) = \sum_{r=0}^{\infty} M_{ij}(r, T)(iw)^r / r!$$

for the characteristic function of $dF_{ij}(x)$ is absolutely convergent with an infinite radius of convergence because, for fixed $w_0 \neq 0$, each summand is bounded above in magnitude by $|w_0|^r / r!$, and thus the radius of convergence must be at least as large as $|w_0|$. Consequently, for finite symbol HMM's, the moments of $dF_{ij}(x)$ uniquely determine $dF_{ij}(x)$. A similar argument holds for continuous symbol HMM's, provided the likelihood functions $f_i(O_T)$ are bounded on the set of all sequences $\{O_T\}$. In this paper, we assume that the likelihood functions are bounded because such an assumption is not particularly restrictive for applications.

Receiver–operator characteristic (ROC) curves [3] are commonly used in the radar and sonar communities to provide quantitative assessments of the correct and incorrect classification rates for classification schemes based on likelihood functions. ROC curves can be used for the same purpose here. To develop a ROC curve for a given classification-related test statistic, say q, under two hypotheses H_i and H_j, the conditional pdf's (using the notation in [3])

$$p_{q|H_i}(Q|H_i) \quad \text{and} \quad p_{q|H_j}(Q|H_j)$$

that define the test statistic q under the different hypotheses H_i and H_j, respectively, must be known. For each real number u, $-\infty < u < +\infty$, we define the probability

$$P_F(u) = \int_u^\infty p_{q|H_i}(Q|H_i)\,dQ$$

and the probability

$$P_D(u) = \int_u^\infty p_{q|H_j}(Q|H_j)\,dQ.$$

The ROC curve for q is simply the locus of points $(P_F(u), P_D(u))$ parameterized by u. The parameter u is usually treated as a decision threshold in applications. Suppose the decision threshold u_{thresh} is selected. Then if $q \geq u_{\text{thresh}}$, the classifier decides H_j. The probability of this decision being correct is P_D, and the probability that it is incorrect is P_F. P_F and $1 - P_D$ are usually referred to as the false alarm and false dismissal probabilities, respectively. Analogous remarks pertain if $q < u_{\text{thresh}}$. Note that the ROC curve for $u = -\infty$ goes through the point $(1, 1)$ and for $u = +\infty$ it goes through the point $(0, 0)$.

The ROC curve of the optimum classifier depicted in Fig. 1, under the hypotheses $O_T\epsilon$ HMM(i) and $O_T\epsilon$ HMM(j), is determined for the likelihood ratio test statistic

$$q_{\text{opt}} = f_j(O_T)/f_i(O_T).$$

The required conditional pdf for q_{opt} is defined by the cdf

$$L_{ij}(x) = \sum_{O_T} H\big(x - \{f_j(O_T)/f_i(O_T)\}\big) f_j(O_T).$$

No recursion for $L_{ij}(x)$ is known, so the only way to evaluate it is by doing the summation; however, this is impractical because the number of terms in the summation grows exponentially in T. Simulation is probably the best method for estimating the ROC curves for the optimal test statistic q_{opt}. In any event, a decision threshold u_{ij} must be set to enable classification to proceed. The "natural" threshold to set is $u_{ij} = 1$ for all i and j, for then the maximum likelihood determines the classification, the classification decision is unique (except for ties) and the classifier depicted in Fig. 1 is obtained. However, in general, it is not necessary to make the natural choice. The best choice depends on the false alarm and false dismissal requirements for each pair of hypotheses $O_T\epsilon$ HMM(i) and $O_T\epsilon$ HMM(j) in the application.

The ROC curve of the suboptimal classifier, under the hypotheses HMM(i) and HMM(j), is determined for the test statistic

$$q_{\text{subopt}} = f_j(O_T).$$

The required conditional pdf's for q_{subopt} are given by $dF_{ji}(x)$ and $dF_{jj}(x)$, respectively. As shown in Section II, the moments of $dF_{ji}(x)$ and $dF_{jj}(x)$ can be computed to any desired order; hence, the ROC curve for q_{subopt} can, in principle, be approximated to any required accuracy without resorting to simulation. A natural choice of decision threshold u_{ij} for each pair of hypotheses $O_T\epsilon$ HMM(i) and $O_T\epsilon$ HMM(j) is not available. Instead, the thresholds must be set by direct examination of the ROC curves.

The test statistic q_{subopt} is identical to q_{opt} in one important special case. If HMM(i) is such that $f_i(O_T)$ in the denominator of q_{opt} is a constant function of O_T, then q_{subopt} can be scaled so that $q_{\text{subopt}} = q_{\text{opt}}$. A situation that might require such an HMM(i) is one in which white noise is being modeled, for then one might anticipate that all observation sequences at the output of the preprocessor are equally likely. The classification statistic is more appropriately referred to as a "detection" statistic in this instance. Thus, a ROC curve for the optimum detection statistic can be developed from the moments computed by the algorithm given in Section II.

The use of q_{subopt} in preference to q_{opt} is appropriate only if the associated conditional pdf's for the ROC curves are "well separated" from each other, and if the application places great emphasis on control of the false alarm or false dismissal probabilities. In this situation, both q_{opt} and q_{subopt} are very likely to perform well; however, estimated ROC curves for q_{opt} would have to be obtained from very large simulations, especially if very small false alarm probabilities or false dismissal probabilities are required in the application. On the other hand, ROC curves for q_{subopt} can be obtained reliably without simulation. In any event, classification performance using q_{subopt} should bound the classification performance using q_{opt}.

II. The Moment Algorithm

Every HMM is comprised of two basic parts: a Markov chain and a set of random variables. The Markov chain has a finite number of states, and each state is uniquely associated with one of the random variables. The state sequence generated by the chain is not observable, i.e., the Markov chain is "hidden." At each time $t = 0, 1, 2, \cdots$, the Markov chain is assumed to be in some state; it transitions to another state at time $t + 1$ according to its transition probability matrix. At each time t, one observation is generated by the random variable associated with the state of the Markov chain at time t. The observations are referred to as symbols. If the random variables assume only a finite set values, the HMM is referred to as a finite symbol HMM. If the random variables assume a continuum of values, the HMM is called a continuous symbol HMM. The full parameter set defining an HMM is comprised of the initial state probability density function of the Markov chain at time $t = 0$, the Markov chain state transition probability matrix, and the pdf's of each of the random observation variables.

The reader is referred to [2] for further discussion of HMM's and the basic algorithms related to them. Of particular importance is the forward–backward algorithm that is used extensively in this section. It is not necessary to read the remainder of this section to understand the examples presented in Section III.

The algorithm is presented separately for finite and continuous symbol HMM's in Section II-A and II-B, respectively. Since the presentation uses only the forward part of the forward–backward algorithm, the algorithm may be named the forward moment algorithm. Section II-C contains a statement of the backward moment algorithm and an identity that is analogous to the Baum identity of the usual forward–backward algorithm.

A. Finite Symbol HMM's

Let HMM(ν) be a hidden Markov chain with $n(\nu)$ states, $\nu = 1, \cdots$. Subscripted indexes will always be written as functions of their subscripts (for instance, $n(\nu)$ is used instead of n_ν) to avoid the later use of subscripted subscripts. Let the state transition probability matrix of HMM(ν) be denoted as $A^\nu = [a^\nu_{i(\nu),j(\nu)}]$ for $i(\nu), j(\nu) = 1, \cdots, n(\nu)$. Let the initial state probability vector of HMM(ν) be denoted as $\pi^\nu = [\pi^\nu_{i(\nu)}]$ for $i(\nu) = 1, \cdots, n(\nu)$.

We first restrict attention to finite symbol HMM's, that is, we suppose that every observation sequence $O_T = \{O(t), t = 1, \cdots, T\}$ is such that

$$O(t) \in V = \{V_1, \cdots, V_m\}$$

where V is the set of all possible output symbols of the preprocessor. The true nature of the symbols in V is of no importance here. HMM's assume that $O(t)$ is a random variable whose probability density function depends on the current state of the Markov chain. Let the discrete probability density function for HMM(ν) when it is in state $i(\nu)$ be denoted as $B^\nu_{i(\nu)}$ for $i(\nu) = 1, \cdots, n(\nu)$. Thus, each $B^\nu_{i(\nu)}$ is a row vector of length m. Stacking these row vectors gives the $n(\nu) \times m$ symbol probability matrix

$$B^\nu = [b^\nu_{i(\nu),j(\nu)}] = \begin{bmatrix} B^\nu_1 \\ B^\nu_2 \\ \vdots \\ B^\nu_{n(\nu)} \end{bmatrix}.$$

Note that

$$b^\nu_{i(\nu)}(V_{j(\nu)}) = b^\nu_{i(\nu),j(\nu)}$$

where we define

$$b^\nu_{i(\nu)}(O(t)) = \Pr[O(t) | \text{HMM}(\nu) \text{ and state } = i(\nu)].$$

The assumption that the training phase is completed means that the parameters HMM(ν) = (π^ν, A^ν, B^ν) are known.

For finite symbol HMM's, $F_{ij}(x)$ has a finite number of jump discontinuities. Let X_{ij} denote the set of all values of x for which $F_{ij}(x)$ is discontinuous. Definition (2) implies that the discontinuities of $F_{ij}(x)$ occur precisely at the different possible values of $f_i(O_T)$. Define the subset $S_i(x)$ of the set of all observation sequences $\{O_T\}$ by

$$S_i(x) = \{O_T : f_i(O_T) = x\}.$$

The sets $S_i(x)$ and $S_i(y)$ are disjoint if $x \neq y$. Also, the union of $S_i(x)$ over all x in X_{ij} is the set $\{O_T\}$ of all observation sequences. Now, from definition (2), it follows that

$$dF_{ij}(x) = F_{ij}(x+) - F_{ij}(x-) = \sum_{O_T \in S_i(x)} f_j(O_T). \tag{4}$$

Substituting (4) into (3) gives

$$\begin{aligned} M_{ij}(k, T) &= \sum_{x \in X_{ij}} x^k \sum_{O_T \in S_i(x)} f_j(O_T) \\ &= \sum_{x \in X_{ij}} \sum_{O_T \in S_i(x)} \{f_i(O_T)\}^k f_j(O_T) \\ &= \sum_{O_T} \Pr[O_T | \text{HMM}(i)]^k \Pr[O_T | \text{HMM}(j)] \end{aligned} \tag{5}$$

where, in the last equation, we have used (1). It is clear from (5) that $M_{ij}(k, T) \neq M_{ji}(k, T)$, in general, for $k > 1$. For $k = 1$, however, we have $M_{ij}(1, T) = M_{ji}(1, T)$ for all i, j, and T.

The expression in (5) is computable directly from the parameters of HMM(i) and HMM(j); however, such a calculation is not practical except for small T because the computational effort increases exponentially in T. To see this, note that the forward–backward algorithm [2] calculates $\Pr[O_T | \text{HMM}(\nu)]$ using $n^2(\nu)T$ multiplications. Thus, each summand in (5) requires $[n(i)n(j)]^2 T^2$ multiplications. There are m^T different possible observation sequences $O_T = \{O(t), t = 1, \cdots, T\}$ because each $O(t)$ can be any one of the m output symbols in V. Thus, direct calculation of (5) requires a total of $[n(i)n(j)]^2 T^2 m^T$ multiplications.

We now derive a recursion for (5) that requires computational effort that grows only linearly with T. The recursion is derived for a more general expression that contains (5) as a special case. For $k = 1, 2, \cdots$, define

$$R(k, T) = \sum_{O_T} \prod_{\nu=1}^{k} \Pr[O_T | \text{HMM}(\nu)]. \tag{6}$$

The application of (6) to compute moments is straightforward; for example, $R(k+1, T)$ equals $M_{21}(k, T)$ when HMM(2) = $\cdots$ = HMM($k+1$). Note that $R(k, T)$ can be interpreted as a joint moment of HMM's, that is, as a joint moment of the likelihood functions $f_i(O_T)$ of the HMM's.

The derivation of the recursion for $R(k, T)$ proceeds as follows. The forward recursion portion of the forward–backward algorithm gives the expression

$$\Pr[O_T | \text{HMM}(\nu)] = \sum_{j(\nu)=1}^{n(\nu)} \alpha^\nu_T(j(\nu)) \tag{7}$$

where, for $2 \leq t \leq T$,

$$\alpha^\nu_t(j(\nu)) = \left[\sum_{i(\nu)=1}^{n(\nu)} \alpha^\nu_{t-1}(i(\nu)) a^\nu_{i(\nu),j(\nu)}\right] b^\nu_{j(\nu)}(O(t)) \tag{8}$$

and

$$\alpha_1^\nu(j(\nu)) = \pi_{j(\nu)}^\nu b_{j(\nu)}^\nu(O(1)). \tag{9}$$

Substitute (7) into (6) to obtain

$$R(k, T) = \sum_{\substack{j(\nu)=1 \\ \nu=1,\cdots,k}}^{n(\nu)} \sum_{O_T} \prod_{\nu=1}^{k} \alpha_T^\nu(j(\nu)) = \sum_{\substack{j(\nu)=1 \\ \nu=1,\cdots,k}}^{n(\nu)} \mu_T(j(1), \cdots, j(k)) \tag{10}$$

where we define, for $t = 1, \cdots, T$,

$$\mu_t(j(1), \cdots, j(k)) = \sum_{O_t} \prod_{\nu=1}^{k} \alpha_t^\nu(j(\nu)). \tag{11}$$

One interpretation of μ_T is that it equals $R(k, T)$ given that HMM(ν) must end in state $j(\nu)$, $\nu = 1, \cdots, k$. We seek a recursion for μ_T. First suppose that $2 \le t \le T$. Then, substituting (8) into (11) gives

$$\begin{aligned}
&\mu_t(j(1), \cdots, j(k)) \\
&= \sum_{O_t} \prod_{\nu=1}^{k} \left\{ \sum_{i(\nu)=1}^{n(\nu)} \alpha_{t-1}^\nu(i(\nu))\, a_{i(\nu),j(\nu)}^\nu b_{j(\nu)}^\nu(O(t)) \right\} \\
&= \sum_{\substack{i(\nu)=1 \\ \nu=1,\cdots,k}}^{n(\nu)} \sum_{O_t} \left\{ \left[\prod_{\nu=1}^{k} \alpha_{t-1}^\nu(i(\nu)) \right] \left[\prod_{\nu=1}^{k} a_{i(\nu),j(\nu)}^\nu \right] \right. \\
&\qquad \left. \cdot \left[\prod_{\nu=1}^{k} b_{j(\nu)}^\nu(O(t)) \right] \right\} \\
&= \sum_{\substack{i(\nu)=1 \\ \nu=1,\cdots,k}}^{n(\nu)} \left[\prod_{\nu=1}^{k} a_{i(\nu),j(\nu)}^\nu \right] \left\{ \sum_{O_t} \left[\prod_{\nu=1}^{k} \alpha_{t-1}^\nu(i(\nu)) \right] \right. \\
&\qquad \left. \cdot \left[\prod_{\nu=1}^{k} b_{j(\nu)}^\nu(O(t)) \right] \right\}.
\end{aligned}$$

Because $\alpha_{t-1}^\nu(i(\nu))$ does not depend on the last symbol $O(t)$ in the observation sequence $O_t = \{O(1), \cdots, O(t)\}$, we have

$$\begin{aligned}
&\mu_t(j(1), \cdots, j(k)) \\
&= \sum_{\substack{i(\nu)=1 \\ \nu=1,\cdots,k}}^{n(\nu)} \left[\prod_{\nu=1}^{k} a_{i(\nu),j(\nu)}^\nu \right] \left\{ \sum_{O_{t-1}} \left[\prod_{\nu=1}^{k} \alpha_{t-1}^\nu(i(\nu)) \right] \right. \\
&\qquad \left. \cdot \sum_{O(t)} \left[\prod_{\nu=1}^{k} b_{j(\nu)}^\nu(O(t)) \right] \right\}.
\end{aligned}$$

Because the sum over $O(t)$ is independent of the observation sequence $O_{t-1} = \{O(1), \cdots, O(t-1)\}$, as well as the indexes $i(\nu)$, and because of (11), we have

$$\begin{aligned}
&\mu_t(j(1), \cdots, j(k)) \\
&\quad = \Gamma(j(1), \cdots, j(k)) \\
&\qquad \cdot \sum_{\substack{i(\nu)=1 \\ \nu=1,\cdots,k}}^{n(\nu)} \left[\prod_{\nu=1}^{k} a_{i(\nu),j(\nu)}^\nu \right] \mu_{t-1}(i(1), \cdots, i(k))
\end{aligned} \tag{12}$$

where

$$\begin{aligned}
\Gamma(j(1), \cdots, j(k)) &= \sum_{O(t)} \prod_{\nu=1}^{k} b_{j(\nu)}^\nu(O(t)) \\
&= \sum_{s=1}^{m} \prod_{\nu=1}^{k} b_{j(\nu)}^\nu(V_s).
\end{aligned} \tag{13}$$

Note that Γ is the joint moment of the random observation variables uniquely associated with the state $j(\nu)$ of HMM(ν).

Equation (12) is the desired recursion for $2 \le t \le T$. For $t = 1$, substituting (9) into (11) gives

$$\begin{aligned}
&\mu_1(j(1), \cdots, j(k)) \\
&\quad = \sum_{O(1)} \prod_{\nu=1}^{k} \alpha_1^\nu(j(\nu)) \\
&\quad = \left(\prod_{\nu=1}^{k} \pi_{j(\nu)}^\nu \right) \sum_{O(1)} \prod_{\nu=1}^{k} b_{j(\nu)}^\nu(O(1)) \\
&\quad = \Gamma(j(1), \cdots, j(k)) \prod_{\nu=1}^{k} \pi_{j(\nu)}^\nu.
\end{aligned} \tag{14}$$

Let $k = 1$. From the definition, it is clear that $R(1, T) = 1$ for all T, regardless of HMM(1) because the sum in (6) is over all O_T. To check independently the recursion (12)–(13), note that, from (13),

$$\Gamma(j(1)) = \sum_{s=1}^{m} b_{j(1)}^1(V_s) = 1, \qquad 1 \le t \le T.$$

From (14), we have

$$\mu_1(j(1)) = \pi_{j(1)}^1.$$

Hence, from (10), we obtain

$$R(1, 1) = \sum_{j(1)=1}^{n(1)} \pi_{j(1)}^1 = 1.$$

The recursion is verified for $T = 1$. For $T = 2$, from (12), we have

$$\begin{aligned}
\mu_2(j(1)) &= \sum_{i(1)=1}^{n(1)} \mu_1(i(1))\, a_{i(1),j(1)}^1 \\
&= \sum_{i(1)=1}^{n(1)} \pi_{i(1)}^1 a_{i(1),j(1)}^1
\end{aligned}$$

so that, from (10),

$$R(1,2) = \sum_{j(1)=1}^{n(1)} \left\{ \sum_{i(1)=1}^{n(1)} \pi^1_{i(1)} a^1_{i(1),j(1)} \right\}$$
$$= \sum_{i(1)=1}^{n(1)} \left\{ \pi^1_{i(1)} \sum_{j(1)=1}^{n(1)} a^1_{i(1),j(1)} \right\}$$
$$= 1$$

and the recursion is verified for $T = 2$.

The first nontrivial special case is $k = 2$. In this case, $R(2, T)$ is identically the first moment $M_{12}(1, T)$. From (12), we have for $2 \le t \le T$

$$\mu_t(j(1), j(2)) = \Gamma(j(1), j(2)) \cdot \sum_{i(1)=1}^{n(1)} \sum_{i(2)=1}^{n(2)} \mu_{t-1}(i(1), i(2)) a^1_{i(1),j(1)} a^2_{i(2),j(2)}$$

and, from (14),

$$\mu_1(j(1), j(2)) = \Gamma(j(1), j(2)) \pi^1_{j(1)} \pi^2_{j(2)}$$

where, from (13),

$$\Gamma(j(1), j(2)) = \sum_{s=1}^{m} b^1_{j(1)}(V_s)\, b^2_{j(2)}(V_s).$$

From (10), then, we have

$$R(2, T) = \sum_{j(1)=1}^{n(1)} \sum_{j(2)=1}^{n(2)} \mu_T(j(1), j(2)).$$

Computation of $R(2, T) = M_{12}(1, T)$ is therefore not excessively laborious.

The evaluation of $R(k, T)$ using the recursion (12) is properly broken into two parts. The first is the precalculation of $\Gamma(j(1), \cdots, j(k))$ for every possible value of the indexes $j(\nu)$. This requires $(k-1)\, m\, N^k$ multiplications and N^k storage locations, where

$$N = \left[\prod_{\nu=1}^{k} n(\nu) \right]^{1/k} \tag{15}$$

is the geometric mean of the number of different states in the various HMM's and is not necesarily an integer. If $N = 8$ and if there are $m = 16$ different observation symbols, then computing and storing Γ for $k = 16$ requires 262 144 storage locations and 2.1×10^7 multiplications. Storage is clearly more crucial an issue than multiplications.

It is possible in some cases to utilize the underlying symmetries of Γ to reduce both storage and computational effort. For example, if $\mathrm{HMM}(2) = \cdots = \mathrm{HMM}(k+1)$, then

$$\Gamma(j(1), j(2), \cdots, j(k+1)) = \Gamma(j(1), \sigma(j(2)), \cdots, \sigma(j(k+1))) \tag{16}$$

for every permutation σ of the k integers $j(2), \cdots, j(k+1)$. The proof of (16) follows easily from (13) because multiplication is commutative. Thus, one only need consider indexes that satisfy

$$1 \le j(1) \le n(1) \quad \text{and}$$
$$1 \le j(2) \le j(3) \le \cdots \le j(k+1) \le n(2).$$

The number of ordered index sets $(j(2), \cdots, j(k+1))$ is equal to the number of combinations of $n(2)$ letters taken k at a time, when each letter may be repeated any number of times up to k. Storage is therefore proportional to

$$N_{k+1} = \left(\frac{n(2)(n(2)+1)\cdots(n(2)+k-1)}{k!} \right) n(1)$$

which is significantly smaller than the $[n(2)]^k n(1)$ storage that would otherwise be necessary. The total multiplication count is also reduced proportionately.

Once Γ has been computed and stored for a given value of k, the recursion (12) can be computed for any length T of the observation sequence. For each of the N^k sets of indexes $\{j(\nu)\}$ in (12), the sum over all N^k indexes $\{i(\nu)\}$ must be undertaken. This sum appears to require kN^k multiplications; however, by using the nested form,

$$\sum_{i(1)=1}^{n(1)} a^1_{i(1),j(1)} \left[\sum_{i(2)=1}^{n(2)} \cdots \left[\sum_{i(k)=1}^{n(k)} a^k_{i(k),j(k)} \mu_{t-1}(i(1), \cdots, i(k)) \right] \cdots \right],$$

it is possible to use approximately

$$N^k + N^{k-1} + \cdots N^2 + N = \left(\frac{N}{N-1} \right)(N^k - 1)$$

instead. If lower order terms are neglected, computing one iteration of (12) requires about N^{2k} multiplications. For an observation sequence of length T, computing μ_T requires on the order of $N^{2k} T$ multiplications. If $N = 8$ and $T = 32$, then 2.2×10^{12} multiplications are required for $k = 6$. Assuming a multiplication takes 1 μs, the calculation requires 611 h and is clearly impractical.

Significant reduction in computational effort is possible in some cases by utilizing the underlying symmetries in μ_t. For example, if $\mathrm{HMM}(2) = \cdots = \mathrm{HMM}(k+1)$, then

$$\mu_t(j(1), j(2), \cdots, j(k+1)) = \mu_t(j(1), \sigma(j(2)), \cdots, \sigma(j(k+1))) \tag{17}$$

for every permutation σ of the k integers $j(2), \cdots, j(k+1)$. The proof of (17) follows easily by induction from (12) because multiplication is commutative and because Γ satisfies the same symmetry property (16) in this case.

Thus, the recursion (12) need be computed for only N_{k+1} sets of indexes. The total multiplication count is reduced to $4N_{k+1}^2 T$, which is significantly smaller than the $N^{2k}T$ multiplications that would otherwise be needed. For the above example requiring 611 h, if $N = n(1) = n(2) = 8$ and if the symmetry (17) is utilized, the calculation would be reduced to roughly a 96 min calculation. Utilizing symmetry is clearly significant in that it can turn an impractical long calculation into a feasible shorter one.

Underflow is potentially a problem when the recursion (12) is computed. It can be easily overcome in exactly the same manner as pointed out in [2] for preventing numerical underflow during the calculation of the forward-backward algorithm. Specifically, let μ_t be computed according to (12) and then multiplied by a scale factor c_t defined by

$$c_t = \left[\sum_{\substack{j(\nu)=1 \\ \nu=1,\cdots,k}}^{n(\nu)} \mu_t(j(1), \cdots, j(k)) \right]^{-1}. \quad (18)$$

Then use the scaled values of μ_t in the recursion (12) to compute μ_{t+1}, which is in turn scaled as shown in (18). If we continue in this fashion and recall the expression in (10), it follows that

$$R(k, T) = \left(\prod_{t=1}^{T} c_t \right)^{-1}. \quad (19)$$

Because the product cannot be evaluated without underflow, we compute instead

$$\log R(k, T) = -\sum_{t=1}^{T} \log c_t. \quad (20)$$

Any convenient scale factor can be used instead of (18). A potentially useful one might be to take $\bar{c}_t = N^k$. Using $\bar{c}_t$ would eliminate the effort of computing the sum in (18) before scaling.

B. Continuous Symbol HMM's

The objective of this section is to show that the moment algorithm for discrete symbol HMM's can be carried over essentially unchanged to continuous symbol HMM's. In fact, it holds also for continuous vector symbol HMM's; however, only the continuous symbol HMM's are treated here for simplicity.

Throughout this section, it is assumed that each output symbol $O(t)$ is a real random variable defined on some underlying event space V. The probability density function of $O(t)$ is uniquely defined for each state $i(\nu) = 1, \cdots, n(\nu)$ of each HMM(ν), $\nu = 1, 2, \cdots$, and is denoted as $b_{i(\nu)}^{\nu}(x)$. Thus, for real numbers α and β with $\alpha < \beta$, we have

$$\int_{\alpha}^{\beta} b_{i(\nu)}^{\nu}(x)\, dx = \Pr\left[\alpha \le O(t) \le \beta \,|\, \text{HMM}(\nu) \text{ and state } = i(\nu)\right]. \quad (21)$$

An observation sequence $O_T = \{x_t, t = 1, 2, \cdots, T\}$ is a sequence of real numbers x_t, with x_t being a realization of the random variable $O(t)$. The posterior likelihood function $f_\nu(O_T)$ is a probability density function for continuous symbol HMM's, as opposed to a simple probability [see (1)] for discrete symbol HMM's. Thus, for real vectors $\vec{\alpha}$ and $\vec{\beta}$ with $\vec{\alpha} < \vec{\beta}$, we have

$$\int_{\vec{\alpha}}^{\vec{\beta}} f_\nu(O_T)\, dO_T = \Pr\left[\vec{\alpha} \le O_T \le \vec{\beta} \,|\, \text{HMM}(\nu)\right] \quad (22)$$

where $dO_T = dx_1 \cdots dx_T$.

For continuous symbol HMM's, the functions $F_{ij}(x)$ are defined just as in (2), but with a T-fold integral over O_T replacing the T-fold sum over O_T. Thus, we have the differential

$$dF_{ij}(x) = \int_{O_T} \delta(x - f_i(O_T)) f_j(O_T)\, dO_T\, dx$$

where $\delta(\cdot)$ denotes the Dirac delta function. From (3), the moments are given by

$$\begin{aligned} M_{ij}(k, T) &= \int_{-\infty}^{\infty} x^k dF_{ij}(x) \\ &= \int_{-\infty}^{\infty} x^k \int_{O_T} \delta(x - f_i(O_T)) f_j(O_T)\, dO_T\, dx \\ &= \int_{O_T} f_j(O_T) \int_{-\infty}^{\infty} x^k \delta(x - f_i(O_T)\, dx\, dO_T \\ &= \int_{O_T} \{f_i(O_T)\}^k f_j(O_T)\, dO_T \end{aligned} \quad (23)$$

which is the continuous analog of (5). It is clear from (23) that $M_{ij}(k, T) = M_{ji}(k, T)$ in general only for the special case $k = 1$. The analog of (6) for continuous symbol HMM's is

$$R(k, T) = \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{T\text{-fold}} \prod_{\nu=1}^{k} f_\nu(O_T)\, dO_T. \quad (24)$$

The forward-backward algorithm for computing the posterior likelihood function for continuous symbol HMM's is modified [5] as follows:

$$f_\nu(O_T) = \sum_{j(\nu)=1}^{n(\nu)} \alpha_T^{\nu}(j(\nu)) \quad (25)$$

where $\alpha_T^{\nu}(j(\nu))$ is computed exactly as given by the recursion (8) and (9), with the only difference being that $b_{j(\nu)}^{\nu}(O(t))$ in (8) is now interpreted as the probability density function implicit in (21). Consequently, (10) still holds exactly if we define

$$\mu_t(j(1), \cdots, j(k)) = \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{t\text{-fold}} \prod_{\nu=1}^{k} \alpha_t^{\nu}(j(\nu))\, dO_t \quad (26)$$

as the analog of (11). Proceeding as before with t-fold integrals replacing t-fold summations gives exactly the recursion (12), but with the one-dimensional integral

$$\Gamma\big(j(1), \cdots, j(k)\big) = \int_{-\infty}^{\infty} \prod_{\nu=1}^{k} b_{j(\nu)}^{\nu}(x)\, dx \quad (27)$$

in place of (13).

The remarks in the preceding section concerning storage, multiplication counts, and symmetry properties all apply for continuous symbol HMM's. The primary difference is that (27) requires an integral evaluation instead of a finite sum as in (13). This evaluation increases the initial computational overhead, but once (27) is computed, the algorithm (12) proceeds exactly as before.

C. The Forward–Backward Moment Algorithm

The moment algorithm presented above in this section used the forward probabilities defined by (8)–(9). It is equally feasible to use the backward probabilities for the same purpose. They are defined by

$$\beta_T\big(j(\nu)\big) = 1$$

and, for $T - 1 \geq t \geq 1$, by

$$\beta_t\big(j(\nu)\big) = \sum_{i(\nu)=1}^{n(\nu)} a_{j(\nu),i(\nu)} b_{i(\nu)}\big(O(t+1)\big)\beta_{t+1}\big(i(\nu)\big).$$

The backward moment algorithm computes, for $1 \leq t \leq T - 1$, the function

$$\tau_t\big(j(1), \cdots, j(k)\big) = \sum_{\tilde{O}_t} \prod_{\nu=1}^{k} \beta_t\big(j(\nu)\big)$$

where $\tilde{O}_t = \{O(t+1), \cdots, O(T)\}$. The backward recursion is given by

$$\tau_T\big(j(1), \cdots, j(k)\big) = 1$$

and, for $T - 1 \geq t \geq 1$, by

$$\tau_t\big(j(1), \cdots, j(k)\big) = \sum_{\substack{i(\nu)=1 \\ \nu=1,\cdots,k}}^{n(\nu)} \left[\prod_{\nu=1}^{k} a_{j(\nu),i(\nu)}\right] \cdot \tau_{t+1}\big(i(1), \cdots, i(k)\big)\Gamma\big(i(1), \cdots, i(k)\big).$$

The derivation of this recursion is similar to that of (12)–(13).

It is straightforward to show that for any t, $1 \leq t \leq T$,

$$R(k, T) = \sum_{\substack{j(\nu)=1 \\ \nu=1,\cdots,k}}^{n(\nu)} \mu_t\big(j(1), \cdots, j(k)\big) \cdot \tau_t\big(j(1), \cdots, j(k)\big).$$

Note that the case $t = T$ is (10). This identity is the analog for $R(k, T)$ of the well-known Baum identity [2] for likelihood functions, i.e.,

$$f_\nu(O_T) = \sum_{i(\nu)=1}^{n(\nu)} \alpha_t\big(i(\nu)\big)\beta_t\big(i(\nu)\big).$$

III. Comparison of Theoretical Moments to Simulation

Ergodic Markov chains are those for which it is possible to transition from every state to every other state, although not necessarily in one step. Left-to-right Markov chains are those for which transitions to lower numbered states are not allowed, that is, have probability zero. These two types of chains are sufficiently different that they are considered separately in the examples.

One interpretation is that ergodic HMM's are models of quasi-stationary signals, while left-to-right HMM's are models of transient signals that ultimately become stationary (because the highest numbered state is not exited once it is entered). One might therefore expect these two types of HMM's to affect classification performance in different ways. The three examples given in this section support this expectation.

Using the above interpretation, the examples may be described as follows. The first example shows that classification using the suboptimal statistic q_{subopt} reliably distinguishes between sufficiently long quasi-stationary signals with a reasonable amount of computational effort. The second example shows that short quasi-stationary and transient signals look significantly different to the HMM transient recognizer, but *not* to the HMM recognizer based on the quasi-stationary signal. The third example shows that noisy observations of transient signals adversely affect classification performance by making the transient signal appear to have a stationary component, which is then misclassified by the HMM transient recognizer.

A. Two Ergodic HMM's

HMM(1) and HMM(2) are five-state, eight-symbol ergodic models whose parameters are given (rounded to three significant decimals) in Tables I and II, respectively. HMM(1) clearly generates observation sequences of uniformly distributed symbols. HMM(2) is more complex in structure, but every symbol can be generated in every state. The fundamental question of interest here is the following. How long must an observation sequence be to guarantee that the suboptimal classification statistic q_{subopt} is highly reliable (say, 99% correct) and has a low false dismissal rate (say, of 0.5%)? We will give what may best be described as a semiempirical answer to this question.

Because of the nature of HMM(1), it is easy to see that

$$f_1(O_T) = \Pr\big[O_T \,|\, \text{HMM}(1)\big] = 8^{-T}.$$

In other words, the posterior likelihood function based on HMM(1) is constant because all observation sequences are equally likely if $O_T \in \text{HMM}(1)$. In particular, $f_1(O_T)$ cannot distinguish $O_T \in \text{HMM}(1)$ from $O_T \in \text{HMM}(2)$ and thus is useless for classification.

The posterior likelihood function based on HMM(2), instead of HMM(1), is useful for classification. Ten-thousand observation sequences O_T of each HMM were generated, and the posterior likelihood $f_2(O_T)$ was com-

TABLE I
PARAMETERS OF HMM(1)

```
NUMBER OF MARKOV STATES = 5

NUMBER OF SYMBOLS PER STATE = 8

INITIAL STATE PROBABILITY VECTOR:
 2.00E-01  2.00E-01  2.00E-01  2.00E-01  2.00E-01

TRANSITION PROBABILITY MATRIX:
 2.00E-01  2.00E-01  2.00E-01  2.00E-01  2.00E-01
 2.00E-01  2.00E-01  2.00E-01  2.00E-01  2.00E-01
 2.00E-01  2.00E-01  2.00E-01  2.00E-01  2.00E-01
 2.00E-01  2.00E-01  2.00E-01  2.00E-01  2.00E-01
 2.00E-01  2.00E-01  2.00E-01  2.00E-01  2.00E-01

SYMBOL PROBABILITY MATRIX (TRANSPOSED):
 1.25E-01  1.25E-01  1.25E-01  1.25E-01  1.25E-01
 1.25E-01  1.25E-01  1.25E-01  1.25E-01  1.25E-01
 1.25E-01  1.25E-01  1.25E-01  1.25E-01  1.25E-01
 1.25E-01  1.25E-01  1.25E-01  1.25E-01  1.25E-01
 1.25E-01  1.25E-01  1.25E-01  1.25E-01  1.25E-01
 1.25E-01  1.25E-01  1.25E-01  1.25E-01  1.25E-01
 1.25E-01  1.25E-01  1.25E-01  1.25E-01  1.25E-01
 1.25E-01  1.25E-01  1.25E-01  1.25E-01  1.25E-01
```

TABLE II
PARAMETERS OF HMM(2), ROUNDED TO THREE SIGNIFICANT DIGITS

```
NUMBER OF MARKOV STATES = 5

NUMBER OF SYMBOLS PER STATE = 8

INITIAL STATE PROBABILITY VECTOR:
 1.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00

TRANSITION PROBABILITY MATRIX:
 1.40E-01  2.35E-01  3.08E-01  1.24E-01  1.94E-01
 1.40E-01  1.14E-01  2.99E-01  2.13E-01  2.34E-01
 4.37E-02  3.20E-01  1.72E-01  1.27E-01  3.38E-01
 9.73E-02  4.97E-01  1.53E-02  1.15E-01  2.75E-01
 2.36E-01  2.49E-02  4.27E-01  2.82E-01  2.98E-02

SYMBOL PROBABILITY MATRIX (TRANSPOSED):
 1.81E-01  1.22E-01  7.89E-03  1.48E-01  7.04E-02
 1.39E-01  8.28E-02  3.23E-02  9.13E-02  1.33E-01
 2.67E-02  1.60E-01  5.87E-02  1.08E-01  2.34E-01
 1.79E-01  1.66E-01  2.18E-01  1.30E-01  5.97E-02
 1.56E-01  1.58E-01  2.15E-01  2.09E-01  2.35E-01
 1.19E-01  5.75E-02  1.11E-01  1.02E-01  1.03E-01
 1.76E-01  1.32E-01  2.40E-01  6.61E-02  1.76E-02
 2.37E-02  1.22E-01  1.17E-01  1.46E-01  1.47E-01
```

puted using the forward-backward algorithm. Fig. 2 shows a histogram of the natural logarithm of $dF_{22}(x)$ for $T = 25$. The observation sequences are thus matched to the posterior likelihood function. Fig. 3 shows a histogram of $\log dF_{21}(x)$ for $T = 25$. In Fig. 3, then, O_T is mismatched to the likelihood function. As is clear from Figs. 2 and 3, the difference between the mean values of the log likelihood functions is about 1.4 standard deviations. Thus, the potential exists for using $\log dF_{22}(x)$ to classify observation sequences; however, $T = 25$ is not long enough to classify with a high probability of detection (i.e., P_D) and a low false alarm probability (i.e., P_F).

A useful observation drawn from Figs. 2 and 3 is that the probability density function of $\log dF_{2j}(x)$ is nicely approximated by the normal distribution. Let μ_{ij} and σ_{ij} denote the mean and standard deviation of $\log dF_{ij}(x)$. Then, if $dF_{ij}(x)$ is log-normal, it is easy to show that μ_{ij}

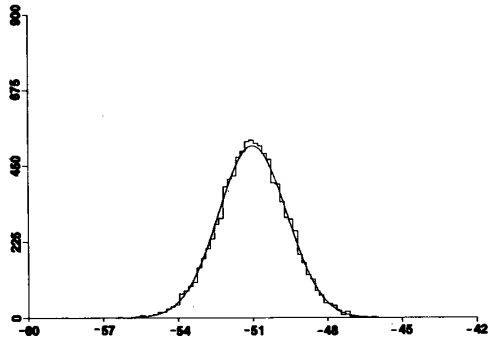


Fig. 2. Histogram of 10 000 values of $\log dF_{22}(x)$ for $T = 25$. (The normal curve has the sample mean and variance given in Table III.)

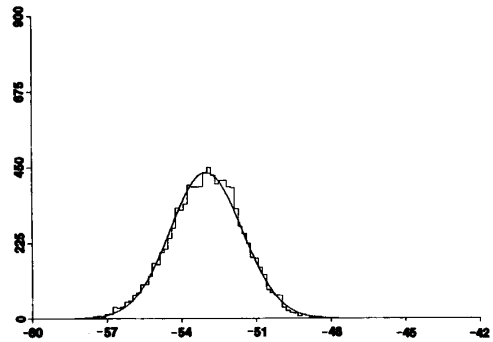


Fig. 3. Histogram of 10 000 values of $\log dF_{21}(x)$ for $T = 25$. (The normal curve has the sample mean and variance given in Table III.)

and σ_{ij} are related to the moments $M_{ij}(k, T)$ by the formulas

$$\mu_{ij} = 2 \log M_{ij}(1, T) - (1/2) \log M_{ij}(2, T) \quad (28)$$

$$\sigma_{ij}^2 = \log M_{ij}(2, T) - 2 \log M_{ij}(1, T). \quad (29)$$

It is stressed that (28) and (29) hold exactly if and only if $dF_{ij}(x)$ is truly log-normal. For finite symbol HMM's, $dF_{ij}(x)$ is necessarily discrete, so that both (28) and (29) must be viewed as approximations. Sufficient conditions under which it may be proved that $dF_{ij}(x)$ is, in some sense, approximately log-normal are unknown. Although the central limit theorem is surely responsible for this log-normal behavior, it is not clear how to apply it in this setting.

Table III gives a comparison between the mean and standard deviations of $\log dF_{2j}(x)$ estimated from 10 000 observation sequences O_T and those calculated from (28) and (29). This table shows good agreement between the approximations of (28) and (29) and the sample means and variances. It also establishes that observation sequences of length $T = 400$ are long enough to distinguish between $O_T \in \text{HMM}(1)$ and $O_T \in \text{HMM}(2)$ with high reliability. That is, the difference between the mean value of $\log dF_{21}(x)$ and the mean value of $\log dF_{22}(x)$ is about 5.2 standard deviations. Assuming $\log dF_{21}(x)$ and $\log dF_{22}(x)$ are normally distributed, as they appear to be,

TABLE III
COMPARISON OF TWO ESTIMATES FOR THE MEAN AND STANDARD DEVIATION OF $\log dF_{2j}(x)$ FOR $j = 1, 2$

	T	Mean Value		Standard Deviation	
		Sample	Eq. 28	Sample	Eq. 29
j = 1	5	-10.8	-10.6	0.95	0.71
	10	-21.3	-21.2	1.11	0.92
	15	-31.9	-31.8	1.24	1.10
	20	-42.4	-42.4	1.35	1.25
	25	-53.0	-52.9	1.47	1.38
	50	-105.8	-105.8	1.93	1.91
	100	-211.4	-211.5	2.62	2.67
	200	-422.6	-423.0	3.60	3.76
	400	-845.0	-845.8	5.09	5.30
j = 2	5	-10.1	-10.1	0.69	0.59
	10	-20.3	-20.3	0.90	0.84
	15	-30.6	-30.5	1.08	1.03
	20	-40.8	-40.8	1.23	1.20
	25	-51.0	-51.0	1.37	1.34
	50	-102.1	-102.1	1.92	1.90
	100	-204.4	-204.4	2.66	2.69
	200	-408.9	-409.0	3.77	3.80
	400	-818.0	-818.1	5.33	5.37

TABLE IV
PARAMETERS OF HMM(3)

NUMBER OF MARKOV STATES = 5

NUMBER OF SYMBOLS PER STATE = 8

INITIAL STATE PROBABILITY VECTOR:

1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

TRANSITION PROBABILITY MATRIX:

6.00E-01	4.00E-01	0.00E+00	0.00E+00	0.00E+00
0.00E+00	7.00E-01	2.00E-01	1.00E-01	0.00E+00
0.00E+00	0.00E+00	6.00E-01	4.00E-01	0.00E+00
0.00E+00	0.00E+00	0.00E+00	7.00E-01	3.00E-01
0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00

SYMBOL PROBABILITY MATRIX (TRANSPOSED):

9.00E-01	1.00E-01	0.00E+00	0.00E+00	0.00E+00
1.00E-01	6.00E-01	0.00E+00	0.00E+00	0.00E+00
0.00E+00	2.00E-01	3.00E-01	0.00E+00	0.00E+00
0.00E+00	1.00E-01	6.00E-01	1.00E-01	0.00E+00
0.00E+00	0.00E+00	1.00E-01	2.00E-01	0.00E+00
0.00E+00	0.00E+00	0.00E+00	4.00E-01	1.00E-01
0.00E+00	0.00E+00	0.00E+00	3.00E-01	6.00E-01
0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.00E-01

then classification using q_{subopt} has a probability of correct classification of 99% for a false alarm rate of 0.5%.

Computing the posterior likelihood function $f_2(O_T)$ for $T = 400$ requires $n^2T = 10\,000$ multiplications; thus, computational requirements for $f_2(O_{400})$ are small enough for practical application. Furthermore, the forward–backward algorithm for computing $f_2(O_T)$ is mathematically equivalent to a nested sequence of matrix–vector multiplications. Consequently, it is possible to reduce total computation time by the design of a "black box" to exploit this special structure in hardware.

B. Mixed Ergodic and Left-to-Right HMM's

HMM(3) is a five-state, eight-symbol left-to-right model whose parameters are given in Table IV. It has a structure that might conceivably arise in the SIIWR problem. Note that HMM(3) never leaves the fifth state once it is entered. Consequently, all sufficiently long observation sequences ultimately contain only the three symbols V_6, V_7, and V_8. Note also that the symbol V_8 occurs if and only if the fifth state has been entered. It follows that an observation sequence O_T containing the symbol V_8 and subsequently containing any of the five symbols V_1, V_2, V_3, V_4, or V_5 must have posterior likelihood zero, i.e., $f_3(O_T) = 0$. Other forbidden symbol sequences may also be noticed. It will be seen that these facts make $f_3(O_T)$ a powerful discriminator against ergodic observation sequences. To summarize briefly, this example will show that short observation sequences of quasi-stationary and transient HMM's look very different to the transient HMM recognizer. On the other hand, all observation sequences look somewhat alike to ergodic HMM recognizers.

When HMM(3) enters its fifth state, it becomes stationary and, consequently, significantly less interesting. Insight into the length of the transient portion of HMM(3) observation sequences is gained by estimating the first passage time of HMM(3) into its fifth state, that is, the number of transitions in the Markov chain before its fifth state is entered. The mean and variance of first passage times may be computed explicitly [6]; however, simulation was used here instead. In 10 000 observation sequences generated for HMM(3), it was found that the mean and standard deviation of the first passage time was 10.9 and 4.8, respectively. The least first passage time was three transitions, and the largest first passage time was 43 transitions. Thus, observation sequences for practical purposes become stationary for $t \geq 50$.

Fig. 4 and Table V clearly show that $dF_{33}(x)$ is a "well-behaved" distribution, even though HMM(3) is not ergodic. However, $dF_{33}(x)$ is not as closely approximated by a log-normal distribution as are $dF_{21}(x)$ and $dF_{22}(x)$, as evidenced by the discrepancy in Table V between the sample statistics and the statistics that would hold if $dF_{33}(x)$ were truly log-normal.

Ten-thousand observation sequences of HMM(1) and HMM(2) were generated and the posterior likelihood $f_3(O_T)$ was computed using the forward–backward algorithm. The observation sequences are thus mismatched to the posterior likelihood function. Table VI gives the number of sequences for which $f_3(O_T) = 0$. Better than 99% rejection of the simulated ergodic HMM observations was attained when $T = 10$, that is, when the observation sequences were about as long as the mean first passage time of HMM(3) into state 5. Total rejection of the 10 000 ergodic observations occurred for $T = 20$.

The ability of $f_3(O_T)$ to reject observations of $O_T \in$ HMM(2) is much more impressive than the $f_2(O_T)$ rejection of $O_T \in$ HMM(3). The lack of symmetry $F_{ij}(x) \neq F_{ji}(x)$ is striking in this instance. Table VII gives estimates of the mean and standard deviation of $\log dF_{23}(x)$, and Fig. 5 is a histogram of the case $T = 25$. The mean values of the 10 000 samples and those predicted by (28) agree very well; however, $dF_{23}(x)$ is not as well approximated by a log-normal as $dF_{22}(x)$ and $dF_{11}(x)$, as seen from the discrepancy in the sample versus the predicted standard deviations. In any event, it is clear by comparing Table VII to the lower half of Table III that $f_2(O_T)$ cannot

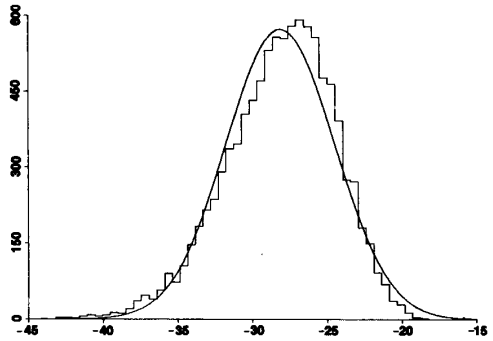


Fig. 4. Histogram of 10 000 values of $\log dF_{33}(x)$ for $T = 25$. (The normal curve has the sample mean and variance given in Table V.)

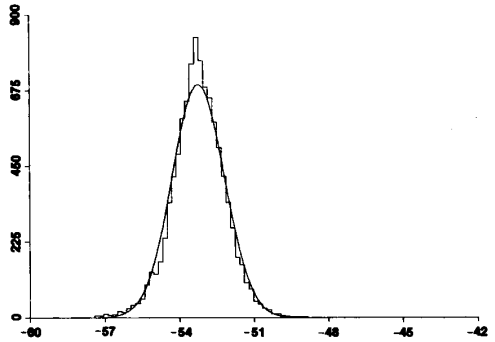


Fig. 5. Histogram of 10 000 values of $\log dF_{23}(x)$ for $T = 25$. (The normal curve has the sample mean and variance given in Table VII.)

TABLE V
COMPARISON OF TWO ESTIMATES FOR THE MEAN AND STANDARD DEVIATION OF $\log dF_{33}(x)$

T	Mean Value		Standard Deviation	
	Sample	Eq. 28	Sample	Eq. 29
5	- 5.6	- 4.9	1.92	1.13
10	-12.8	-11.8	2.30	1.91
15	-18.6	-18.2	2.72	2.33
20	-23.5	-22.6	3.22	2.29
25	-28.1	-26.3	3.61	2.15
50	-50.5	-47.2	4.59	2.75

TABLE VI
NUMBER OF $O_T \in \text{HMM}(i)$ FOR WHICH $f_3(O_T) = 0$, $i = 1, 2$

T	HMM(1)	HMM(2)
5	9389	9172
10	9937	9918
15	9997	9988
20	10000	10000

TABLE VII
COMPARISON OF TWO ESTIMATES FOR THE MEAN AND STANDARD DEVIATION OF $\log dF_{23}(x)$

T	Mean Value		Standard Deviation	
	Sample	Eq. 28	Sample	Eq. 29
5	- 10.8	-10.8	0.51	0.60
10	- 21.4	-21.4	0.91	0.89
15	- 32.0	-32.0	1.02	1.04
20	- 42.6	-42.7	1.04	1.15
25	- 53.2	-53.5	1.05	1.12
50	-106.4	-106.8	1.11	1.43

reliably distinguish $O_T \in \text{HMM}(3)$ from $O_T \in \text{HMM}(2)$ when $T = 50$. However, since the first passage time of HMM(3) is almost certainly less than $T = 50$, increasing the observation sequence length to improve reliability is not appropriate if the underlying intent is the classification of the transient portion of HMM(3).

C. Left-to-Right HMM with Noise

In this example, the effect of noise on the reliability of the q_{subopt} classifier is assessed for the left-to-right model HMM(3). The right way to study noise in finite symbol HMM's is to add the noise to the original time series $s(t)$ and then analyze the particular preprocessor under consideration to determine the noisy symbol sequence. However, no particular preprocessor is proposed here, and so we resort to modeling noise in much the same way that Shannon modeled noisy discrete memoryless channels [7]. This approach can give an indication of the successful classification rate as a function of the probable number of incorrect symbols in an observation sequence, but it cannot provide an assessment of the effect of signal-to-noise ratio on classification because such an assessment requires knowledge of the preprocessor.

Denote by h_{kj} the probability that the observation symbol V_k is altered to symbol V_j by the noise mechanism and define the $m \times m$ noise probability matrix $H = [h_{kj}]$. It is assumed that H is independent of the state of the Markov chain and of time t. Consequently, the output of a given HMM corrupted by noise is equivalent to another HMM that is noiseless. If $\lambda = (\pi, A, B)$ are the parameters of a given HMM with noise matrix H, the parameters of the equivalent noiseless HMM are $\tilde{\lambda} = (\pi, A, BH)$. The proof is straightforward: the product $b_{ik}h_{kj}$ is the probability that the state of the Markov chain is i and that symbol j is produced, given that symbol k was the output of the given HMM. The sum over k of $b_{ik}h_{kj}$ gives the component $\tilde{b}_{ij}$ of the equivalent noiseless HMM symbol probability matrix $\tilde{B}$. Clearly, $\tilde{b}_{ij}$ equals the (i, j) component of the product BH, so that $\tilde{B} = BH$.

The noise probability matrix H must be row stochastic, that is, every row sum must equal one. The HMM-generated symbol V_k is altered by noise to one of the available symbols, so that row k must sum to one.

Because H has row sums equal to one, the matrix $\tilde{B}$ is a valid symbol probability matrix for the equivalent noiseless HMM, that is, each row of $\tilde{B} = BH$ sums to one. We have

$$\begin{aligned}\sum_{j=1}^{m} \tilde{b}_{ij} &= \sum_{j=1}^{m}\sum_{k=1}^{m} b_{ik}h_{kj} \\ &= \sum_{k=1}^{m} b_{ik} \sum_{j=1}^{m} h_{kj} \\ &= \sum_{k=1}^{m} b_{ik} \\ &= 1.\end{aligned}$$

The worst case noise probability matrix, denoted H^0, has the constant entry $h_{ij}^0 = 1/m$ for all i and j. In this case,

$$\tilde{b} = \sum_{k=1}^{m} b_{ik}h_{kj} = \frac{1}{m}\sum_{k=1}^{m} b_{ik} = \frac{1}{m}.$$

Consequently, HMM's with noise probability matrix H^0 are indistinguishable. In fact, H^0 makes all HMM's statistically equivalent to the ergodic HMM(1) given in Table I.

Let $\Pr[V_i]$ be the relative frequency of occurrence of the symbol V_i in observation sequences of length T before the addition of noise. Thus, we have $\Sigma \Pr[V_i] = 1$. After alteration by noise, the probability of correct occurrences of V_i in O_T is then $\Pr[V_i]h_{ii}$. The probability that the symbol $O(t) \in O_T$ is correct is

$$D_T = \sum_{i=1}^{m} \Pr[V_i]h_{ii} \tag{30}$$

and the probability that $O(t)$ is incorrect is

$$E_T = 1 - D_T. \tag{31}$$

For the examples here, given a specific value of E_T, we choose the simple noise probability matrix H defined by

$$h_{ii} = 1 - E_T, \qquad \text{all } i$$

$$h_{ij} = \frac{E_T}{m-1}, \qquad \text{all } i \neq j. \tag{32}$$

For this choice of H, D_T is independent of the actual values of $\Pr[V_i]$, as is clear from (30) and the fact that $\Sigma \Pr[V_i] = 1$.

Noise tends to make observations of all HMM's look like observations of HMM(1), and ergodic observation sequences tend to have forbidden symbol sequences for the left-to-right HMM(3). The first natural issue is therefore to determine how many forbidden symbol sequences occur as a function of the incorrect symbol probability E_T. Table VIII gives the results for various values of T and E_T, based on simulations of 10 000 observation sequences. It shows that forbidden symbol sequences are less likely for small T than for large T. This table also shows that noisy observations of HMM(3) do not have as high a proportion of forbidden symbol sequences as observations of HMM(1) and HMM(2), even for $E_T = 10\%$, as can be seen by comparing Tables VI and VIII. One may conclude from Table VIII that E_T must be small and T must be short to minimize misclassification due to forbidden symbol sequences. For instance, if $T = 25$ and $E_T = 0.001$, the false dismissal probability is apparently at least 1.21%. Shorter T, however, causes smaller shifts in the statistics in the likelihood function, and thus increases the misclassification rate. Consequently, a tradeoff exists between short T and long T.

The total false dismissal probability can be expressed as the sum of the false dismissal probability due to forbidden symbol sequences and the false dismissal probability due to noise-induced shift in the statistics of the nonzero values of the posterior likelihood function. We examine the total false dismissal probability for HMM(4), which is the HMM equivalent to HMM(3) with the noise matrix H given by (32) with $E_T = 0.001$. The parameters of HMM(4) are given explicitly in Table IX.

TABLE VIII
NUMBER OF $O_T \in$ HMM(3) + NOISE FOR WHICH $f_3(O_T) = 0$ AT VARIOUS VALUES OF E_T

T	E_T = 0.1	0.01	0.001	0.0001
5	2194	236	23	1
10	3906	443	37	1
15	5305	651	64	11
20	6625	986	103	13
25	7643	1303	121	11
50	9643	2684	345	34

TABLE IX
PARAMETERS OF HMM(4), ROUNDED TO THREE SIGNIFICANT DIGITS

NUMBER OF MARKOV STATES = 5

NUMBER OF SYMBOLS PER STATE = 8

INITIAL STATE PROBABILITY VECTOR:

1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

TRANSITION PROBABILITY MATRIX:

6.00E-01	4.00E-01	0.00E+00	0.00E+00	0.00E+00
0.00E+00	7.00E-01	2.00E-01	1.00E-01	0.00E+00
0.00E+00	0.00E+00	6.00E-01	4.00E-01	0.00E+00
0.00E+00	0.00E+00	0.00E+00	7.00E-01	3.00E-01
0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00

SYMBOL PROBABILITY MATRIX (TRANSPOSED):

8.99E-01	1.00E-01	1.43E-04	1.43E-04	1.43E-04
1.00E-01	5.99E-01	1.43E-04	1.43E-04	1.43E-04
1.43E-04	2.00E-01	3.00E-01	1.43E-04	1.43E-04
1.43E-04	1.00E-01	5.99E-01	1.00E-01	1.43E-04
1.43E-04	1.43E-04	1.00E-01	2.00E-01	1.43E-04
1.43E-04	1.43E-04	1.43E-04	4.00E-01	1.00E-01
1.43E-04	1.43E-04	1.43E-04	3.00E-01	5.99E-01
1.43E-04	1.43E-04	1.43E-04	1.43E-04	3.00E-01

Denote by $F_{ij}^0(x)$ the cumulative distribution function

$$F_{ij}^0(x) = [F_{ij}(x) - F_{ij}(0)]/[F_{ij}(\infty) - F_{ij}(0)]. \tag{33}$$

Ten-thousand observation sequences O_T were generated from HMM(4) for $T = 25$. As given in Table VIII, 121 sequences resulted in zero posterior likelihood function values (that is, $f_3(O_T) = 0$) and the remaining 9879 nonzero values of $f_3(O_T)$ give the histogram shown in Fig. 6. By comparison to Fig. 4, it is clear that no significant difference between $\log dF_{34}^0(x)$ and $\log dF_{33}(x)$ is evident. Therefore, the misclassification rate due to noise-induced shifts in the statistics of $dF_{34}^0(x)$ is very small. The suboptimal classifier q_{subopt} for HMM(3) thus gives 98.8% correct classification and a 1.2% false dismissal probability when used with noisy observations characterized by $O_T \in$ HMM(4).

Because $E_T = 0.001$ in this example, each observation sequence O_{25} has probability 0.025 of having at least one incorrect symbol. Of 10 000 observation sequences, the expected number with at least one incorrect symbol is 250. Nearly half (121) contained forbidden symbol sequences and caused the only significant misclassification problem. The other half apparantly made no contribution to the probability of false dismissal.

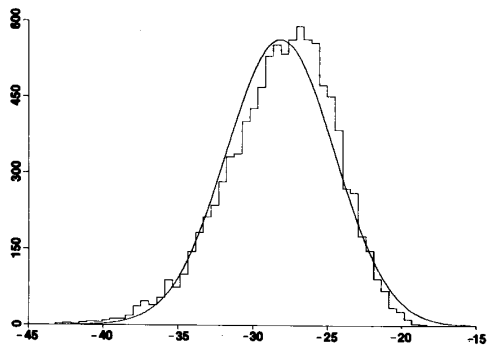


Fig. 6. Histogram of 9879 samples of $\log dF_{34}^{0}(x)$ for $T = 25$. (The normal curve has the sample mean $= -28.156$ and the variance $= 3.6167$.)

It would be desirable to be able to compute the moments of $F_{ij}^{0}(x)$ instead of $F_{ij}(x)$. Alternatively, it would be desirable to be able to compute the amplitude of the impulse (delta function) in $dF_{ij}(x)$ that seems to be present in the left-to-right HMM's considered here. In other words, if we write

$$dF_{ij}(x) = A_0\delta(x) + dF_{ij}^{0}(x), \tag{34}$$

then an algorithm to compute A_0 directly would be worthwhile. Knowing A_0 and the moments of F_{ij} gives the moments of $F_{ij}^{0}(x)$. However, developing such an algorithm requires further work.

IV. Concluding Remarks

If the distribution $dF_{ij}(x)$ is approximately log-normal, the first two moments $M_{ij}(1, T)$ and $M_{ij}(2, T)$ can be used to develop a continuous approximation to $dF_{ij}(x)$. Simulations suggest that $dF_{ij}(x)$ is approximately log-normal whenever HMM(i) and HMM(j) are ergodic and nontrivial. (A "trivial" HMM is an HMM whose likelihood function $f(O_T)$ is constant.) A proof of approximate log-normality that relies on the central limit theorem is not obvious in the present context. If the distribution $dF_{ij}(x)$ is not approximately log-normal, the higher order moments $M_{ij}(k, T)$ are needed to develop reasonable continuous approximations to $dF_{ij}(x)$. The forward–backward moment algorithm presented in this paper computes these moments explicitly from the defining HMM parameter sets.

The use of the suboptimal classification statistic q_{subopt} in preference to the optimum statistic q_{opt} is probably not appropriate in many speech applications because of the ready availability of both likelihoods needed to form the likelihood ratio q_{opt}. Unfortunately, simulations are required to determine the ROC curves for q_{opt}. Consequently, for applications that require small incorrect classification probability and high probability of correct classification, very large simulations are necessary to confidently establish the required performance. An alternative in this case is to use the suboptimal statistic q_{subopt} because the ROC curves can be approximated in principle to any required accuracy without simulations.

The suboptimal statistic q_{subopt} is identical (to within a constant scale factor) to the optimal statistic q_{opt} when the problem is more akin to detection than to classification. That is, if the application is that of distinguishing the presence of a signal embedded in noise from the presence of noise alone, and if the HMM noise model is a "trivial" model as defined above, then the optimal detection statistic and q_{subopt} are identical. As a result, in this case, the moments of the optimal detection statistic can be computed using the forward–backward moment algorithm, and the ROC curves for the optimal detection statistic can be approximated to any required accuracy.

References

[1] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the application of vector quantization and hidden Markov models to speaker-independent isolated word recognition," *Bell Syst. Tech. J.*, vol. 62, pp. 1075–1105, Apr. 1983.

[2] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Tech. J.*, vol. 62, pp. 1035–1074, Apr. 1983.

[3] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*. New York: Wiley, 1968, sect. 2.2.2.

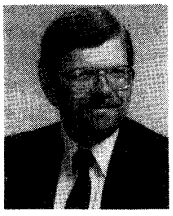
[4] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 1965, p. 158.

[5] L. A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 729–734, Sept. 1982.

[6] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. Princeton, NJ: Van Nostrand, 1960, ch. 3.

[7] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, 1948.

[8] R. L. Streit and R. F. Barrett, "Frequency line tracking using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Processing*, this issue, pp. 586–598.



Roy L. Streit (SM'84) was born in Guthrie, OK, on October 14, 1947. He received the B.A. degree (with Honors) in mathematics and physics from East Texas State University, Commerce, in 1968, the M.A. degree in mathematics from the University of Missouri, Columbia, in 1970, and the Ph.D. degree in mathematics from the University of Rhode Island, Kingston, in 1978.

He was a Visiting Scholar in the Department of Operations Research, Stanford University, Stanford, CA, during 1981–1982, and an Exchange Scientist in the Signal Processing and Classification Group at the Defence Science and Technology Organization, Adelaide, South Australia, from 1987 to 1989. He joined the staff of the Naval Underwater Systems Center (then the Navy Underwater Sound Laboratory), New London, CT, in 1970. He is an Applied Mathematician and has published work in several areas, including towed array design, complex function approximation, semi-infinite programming, and applications of hidden Markov models. His current interests include image analysis, tracking problems, and training algorithms for neural networks.