**2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011**

Bounds on the Probability of Misclassification among Hidden Markov Models

Christoforos Keroglou and Christoforos N. Hadjicostis

***Abstract*— Given a sequence of observations, classification among two known hidden Markov models (HMMs) can be accomplished with a classifier that minimizes the probability of error (i.e., the probability of misclassification) by enforcing the maximum *a posteriori* probability (MAP) rule. For this MAP classifier, we are interested in assessing the *a priori* probability of error (before any observations are made), something that can be obtained (as a function of the length of the sequence of observations) by summing up the probability of error over all possible observation sequences of the given length. To avoid the high complexity of computing the exact probability of error, we devise techniques for merging different observation sequences, and obtain corresponding upper bounds by summing up the probabilities of error over the merged sequences. We show that if one employs a deterministic finite automaton (DFA) to capture the merging of different sequences of observations (of the same length), then Markov chain theory can be used to efficiently determine a corresponding upper bound on the probability of misclassification. The result is a class of upper bounds that can be computed with polynomial complexity in the size of the two HMMs and the size of the DFA.**

***Index Terms*— hidden Markov model, probability of error, classification, probabilistic diagnosis, stochastic diagnoser.**

I. INTRODUCTION

We consider classification among systems that can be modeled as hidden Markov models (HMMs). Given a sequence of observations that is generated by underlying (unknown) activity in one of two known HMMs, we analyze the performance of the MAP classifier, which minimizes the probability of misclassification [1], by characterizing the *a priori* probability of error, i.e., the probability of error before any observations are made. The precise calculation of the probability of error (for sequences of observations of a given finite length) is a combinatorial task of high complexity (typically exponential in the length of the sequences). In this paper, we circumvent this problem by focusing on obtaining upper bounds on the probability of misclassification. In particular, we employ finite automata to merge sequences of observations of the same length in different ways; calculating in each case an upper bound on the probability of misclassification by summing up the individual probabilities of misclassification over the merged sequences.

Our analysis and bounds can find application in many areas where HMMs are used, including speech recognition [2], [3], [4], pattern recognition [5], bioinformatics [6], [7] and failure diagnosis in discrete event systems [1], [8], [9]. Our work also relates to approaches dealing with the *distance* or dissimilarity between two HMMs [10], [11], [12] and the construction we devise to obtain our bounds encompasses the concept of a stochastic diagnoser [9]. Directly related previous work can be found in [1], which introduces an upper bound on the probability of misclassification, applicable to the case when the two HMMs have different languages.[1] More specifically, given two models $S^{(1)}$ and $S^{(2)}$ with languages $L(S^{(1)})$ and $L(S^{(2)})$ respectively, [1] obtains an upper bound on the probability of misclassification by focusing on the probability of strings in $L(S^{(1)})-L(S^{(2)})$ or $L(S^{(2)})-L(S^{(1)})$. Under certain conditions (which require, among other things, that $L(S^{(1)}) \neq L(S^{(2)})$), this bound tends to zero exponentially with the number of observation steps.

The contribution of this paper is the characterization of a class of upper bounds on the *a priori* probability of error when classifying among two known HMMs that may not necessarily have different languages. By introducing an appropriate deterministic finite automaton (DFA), we systematically merge different sequences of the same length in a way that allows easy computation of an upper bound on the probability of misclassification. In particular, for sequences of observations of a given length n, our bounds can be obtained with linear complexity in n, which should be contrasted against the generally exponential complexity in n for obtaining the exact probability of error. Our approach also allows us to use Markov chain theory to obtain an upper bound for asymptotically large n (in all cases, the approach has complexity polynomial in the size of the two given HMMs and the size of the DFA that is used).

II. NOTATION AND BACKGROUND

An HMM is described by a five-tuple $(Q, E, \Delta, \Lambda, \pi_0)$, where $Q = \{q_1, q_2, ..., q_{|Q|}\}$ is the finite set of states; $E = \{e_1, e_2, ..., e_{|E|}\}$ is the finite set of outputs; $\Delta : Q \times Q \rightarrow [0\ 1]$ captures the state transition probabilities; $\Lambda : Q \times E \times Q \rightarrow [0\ 1]$ captures the output probabilities associated with transitions; π_0 is the initial state probability distribution vector. For $q,\ q' \in Q$ and $\sigma \in E$, the state

This material is based upon work supported in part by the European Community (EC) 7^{th} Framework Programme (FP7/2007-2013), under grants INFSO-ICT-223844 and PIRG02-GA-2007-224877. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of EC.

The authors are with the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus. E-mails: {keroglou.christoforos, chadjic}@ucy.ac.cy.

[1] The language of an HMM consists of the set of all finite length sequences of outputs (observations) that can be generated by the HMM starting from a valid initial state.

978-1-61284-799-3/11/$26.00 ©2011 IEEE

transition probabilities are defined as

$$\Delta(q,q') \equiv P(q[n+1]=q' \mid q[n]=q) \ ,$$

and the output probabilities associated with transitions are given by

$$\Lambda(q,\sigma,q') = P(q[n+1]=q', E[n+1]=\sigma \mid q[n]=q) \ ,$$

where $q[n]$ ($E[n]$) is the state (output/observation) of the HMM at time step n. The output function Λ describes the conditional probability of observing the output σ associated with the transition to state q' from state q. The state transition function needs to satisfy

$$\Delta(q,q') \quad = \quad \sum_{\sigma\in E}\Lambda(q,\sigma,q') \qquad (1)$$

and also $\sum_{i=1}^{|Q|}\Delta(q,q_i)=1$, $\forall q\in Q$.

We define the $|Q|\times|Q|$ matrix A_σ, associated with output $\sigma\in E$ of the HMM, as follows: the $(k,j)^{th}$ entry of A_σ captures the probability of a transition from state q_j to state q_k that produces output σ, i.e., $A_\sigma(k,j)=\Lambda(q_j,\sigma,q_k)$. Note that $A=\sum_{\sigma\in E}A_\sigma$, is a column stochastic matrix whose $(k,j)^{th}$ entry denotes the probability of taking a transition from state q_j to state q_k, without regard to the output produced, i.e., $A(k,j)=\Delta(q_j,q_k)$.

Suppose that we are given two HMMs, captured by $S^{(1)}=(Q^{(1)},E^{(1)},\Delta^{(1)},\Lambda^{(1)},\pi_0^{(1)})$ and $S^{(2)}=(Q^{(2)},E^{(2)},\Delta^{(2)},\Lambda^{(2)},\pi_0^{(2)})$, with prior probabilities for each model given by P_1 and $P_2=1-P_1$, respectively. Given $E^{(j)}=\{e_1^{(j)},e_2^{(j)},...,e_{|E^{(j)}|}^{(j)}\}$, $j=\{1,2\}$, for the two HMMs, we define $E=E^{(1)}\cup E^{(2)}$ with $E=\{e_1,e_2,...,e_{|E|}\}$ and let $A_{e_i}^{(j)}$ be the transition matrix for $S^{(j)}$, $j=\{1,2\}$, under the output symbol $e_i\in E$. We set $A_{e_i}^{(j)}$ to zero if $e_i\in E-E^{(j)}$. If we observe a sequence of n outputs $Y_1^n=y[1],y[2],...,y[n],y[i]\in E$, that is generated by one of the two underlying HMMs, the classifier that minimizes the probability of error needs to implement the maximum *a posteriori* probability (MAP) rule. Specifically the MAP classifier compares

$$P(S^{(1)}\mid Y_1^n) \ \gtrless \ P(S^{(2)}\mid Y_1^n) \Rightarrow \frac{P(Y_1^n\mid S^{(1)})}{P(Y_1^n\mid S^{(2)})} \ \gtrless \ \frac{P_2}{P_1},$$

and decides in favor of $S^{(1)}$ ($S^{(2)}$) if the left (right) quantity is larger. It is obvious that when we decide in favor of one or the other model, then we have probability of error proportional to the probability of the model that was not selected. With some algebra, it can be shown that $P(\text{error},Y_1^n)=\min\{P_1\cdot P(Y_1^n\mid S^{(1)}),P_2\cdot P(Y_1^n\mid S^{(2)})\}$. Clearly, if $E^{(1)}\neq E^{(2)}$ and at least one symbol $y[i]$ is unique to $S^{(1)}$ (i.e., $y[i]\in E-E^{(2)}$) or to $S^{(2)}$ (i.e., $y[i]\in E-E^{(1)}$), then we will choose the model with nonzero probability of error (assuming the sequence of observations was indeed generated by one of the two models) and will make an error with zero probability.

III. Probability of Misclassification

Step 1. Probability of Misclassification

To calculate the *a priori* probability of error before the sequence of observations of length n is observed, we need to consider all possible observation sequences of length n, so that

$$P(\text{error at } n) \quad = \quad \sum_{Y_1^n\in E^n}P(\text{error},Y_1^n), \qquad (2)$$

where E^n is the set of all sequences of length n with outputs from E (some of these sequences may have zero probability under one of the two models or even both models).

We arbitrarily index each of the d^n ($d=|E|$) sequences of observations via $Y(i)$, $i\in\{1,2,...,d^n\}$, and use $P_i^{(j)}$ to denote $P_i^{(j)}=P(Y(i)|S^{(j)})$. The probability of misclassification between the two systems, after n steps, can then be expressed as

$$\begin{aligned} P(\text{error at } n) \quad &= \quad \sum_{i=1}^{d^n}P(\text{error},Y(i)) \\ &= \quad \sum_{i=1}^{d^n}\min\{P_1\cdot P_i^{(1)},P_2\cdot P_i^{(2)}\}. \qquad (3)\end{aligned}$$

We can calculate $P_i^{(j)}=P(Y(i)|S^{(j)})$ with an iterative algorithm, a description of which can be found in [1]. For sequence $Y_1^n=y[1],y[2],...,y[n]$, we calculate $\rho_n^{(j)}=A_{y[n]}^{(j)}A_{y[n-1]}^{(j)}...A_{y[1]}^{(j)}\pi_0^{(j)}$, which is essentially a vector whose k^{th} entry captures the probability of reaching state $q_k\in Q^{(j)}$ while generating the sequence of outputs Y_1^n (i.e., $\rho_n^{(j)}(k)=P(q[n]=q_k,Y_1^n)$). If we sum up the entries of $\rho_n^{(j)}$ we obtain $P_{Y_1^n}^{(j)}=P(Y_1^n\mid S^{(j)})=\sum_{k=1}^{|Q^{(j)}|}\rho_n^{(j)}(k)$.

We can obtain the probability of error over all sequences of n observations by calculating and comparing the $\rho_n^{(j)}$, $j=1,2$, for all possible sequences of observations of length n. We can arrange the computations in terms of two d-ary trees of depth n, as shown in Fig. 1. Each node at level L represents $\rho_L^{(j)}$, $j=1,2$, after a specific sequence (of exactly L) observations has been seen. For each node at level L, we create d child-nodes, and we repeat this procedure until having n-levels in the tree.

Once we expand these trees to n-levels, each of the d^n leaf nodes corresponds to a unique sequence of length n, which, in the worst case scenario, can be produced by both HMM models. We assign to each leaf-node a probability of occurring $P_i^{(j)}=P(Y_1^n=Y(i)\mid S^{(j)})$, where $j\in\{1,2\}$ represents the model and $i\in\{1,2,...,d^n\}$ corresponds to the index of each sequence of n observations.

Example 1:

Suppose we are given the two HMMs shown in Fig. 2, with $E^{(1)}=E^{(2)}=E=\{\alpha,\beta\}$, $\pi_0^{(1)}=\pi_0^{(2)}=\left[\begin{array}{cc}1 & 0\end{array}\right]^T$, and $P_1=P_2=0.5$. The corresponding $A_\alpha^{(1)},A_\beta^{(1)},A_\alpha^{(2)},A_\beta^{(2)}$ are as follows:

$$\mathcal{A}_\alpha^{(1)}=\left[\begin{array}{cc}0 & 0.95\\ 0 & 0.05\end{array}\right], \mathcal{A}_\beta^{(1)}=\left[\begin{array}{cc}0 & 0\\ 1 & 0\end{array}\right],$$

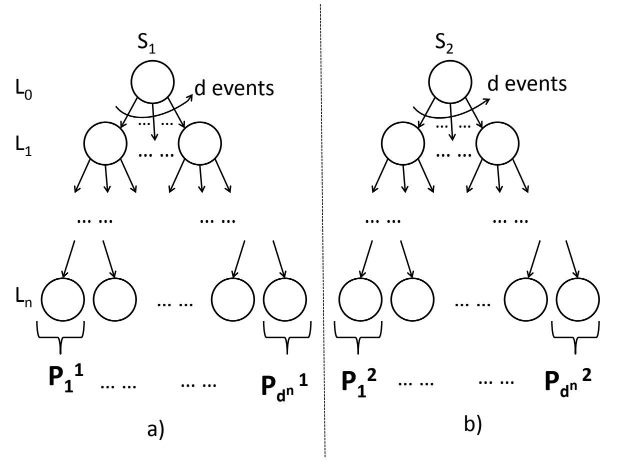

Fig. 1. Computation on two d-ary trees of depth n, one for $S^{(1)}$ and one for $S^{(2)}$.

$$\mathcal{A}_\alpha^{(2)} = \begin{bmatrix} 0 & 0.05 \\ 0 & 0.95 \end{bmatrix}, \mathcal{A}_\beta^{(2)} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

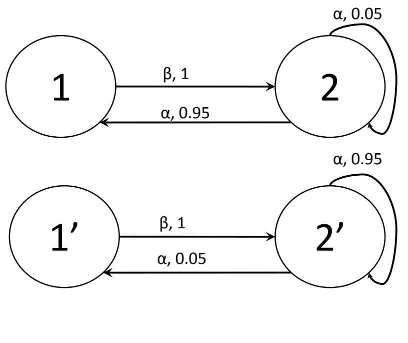

Fig. 2. $S^{(1)}$ (left) and $S^{(2)}$ (right) in Example 1.

If the sequence $Y(\ell) = baba$ is observed, we have $P_\ell^{(1)} = \sum_{k=1}^{|Q^{(1)}|} \rho_4^{(1)}(k) = 0.05$, where $\rho_4^{(1)} = A_a^{(1)}A_b^{(1)}A_a^{(1)}A_b^{(1)}\pi_0^{(1)}$ and $P_\ell^{(2)} = \sum_{k=1}^{|Q^{(2)}|} \rho_4^{(2)}(k) = 0.95$, where $\rho_4^{(2)} = A_a^{(2)}A_b^{(2)}A_a^{(2)}A_b^{(2)}\pi_0^{(2)}$. Thus, the probability of error between the two models if this specific sequence is observed is $P(\text{error}, Y(\ell)) = 0.025$.

Step 2. Upper Bound for Probability of Error

If we have two sequences $Y(1)$ and $Y(2)$ of length n, we can obtain an upper bound on the probability of error for these sequences as follows:

$$P(\text{error}, \{Y(1), Y(2)\}) = \sum_{i=1}^{2} \min\{P_1 \cdot P_i^{(1)}, P_2 \cdot P_i^{(2)}\} \\ \leq \min\{P_1 \cdot \sum_{i=1}^{2} P_i^{(1)}, P_2 \cdot \sum_{i=1}^{2} P_i^{(2)}\}\ . \tag{4}$$

The above can be shown easily by considering the different cases and observing that $\min\{a_1, a_2\} + \min\{b_1, b_2\} \leq \min\{a_1 + b_1, a_2 + b_2\}$. We can easily generalize the above discussion to any number of merged sequences of the same length. The next step is to find an upper bound for the probability of error at n steps. In particular, if we take any partition of the index set $I = \{1, 2, ..., d^n\}$, into subsets $D_1, D_2, ..., D_m$ (such that $D_i \cap D_j = \emptyset$ for $i \neq j$ and $\cup_{i=1}^m D_i = I$), then we have

$$\begin{aligned} P(\text{error at } n) &= \sum_{\ell=1}^{d^n} P(\text{error}, Y(\ell)) \\ &= \sum_{k=1}^{m} \sum_{\ell \in D_k} \min\{P_1 \cdot P_\ell^{(1)}, P_2 \cdot P_\ell^{(2)}\} \\ &\leq \sum_{k=1}^{m} \min\{\sum_{\ell \in D_k} P_1 \cdot P_\ell^{(1)}, \sum_{\ell \in D_k} P_2 \cdot P_\ell^{(2)}\}. \end{aligned} \tag{5}$$

Step 3. Calculation of Upper Bound via a DFA

We now discuss how we can obtain a partition of the index set I, via a deterministic finite automaton (DFA) H with language E^*. The reason we consider this particular partitioning of I will become clearer later when we discuss efficient ways of calculating the quantities $\sum_{\ell \in D_k} P_1 \cdot P_\ell^{(j)}$, $j = 1, 2$.

A DFA H is described by a four-tuple (X, E, δ, x_0), where $X = \{x_1, x_2, ..., x_{|X|}\}$ is the finite set of states; $E = \{e_1, e_2, ..., e_{|E|}\}$ is the finite set of inputs (alphabet); $\delta : X \times E \rightarrow X$ is the transition function; and $x_0 \in X$ is the initial state. For a sequence of events $s = s[n]s[n-1]...s[1]$, $s[i] \in E$, $i = 1, 2, ..., n$, we define $\delta(q, s) = \delta(...\delta(\delta(q, s[1]), s[2]), ..., s[n])$.

A sufficient condition for the requirement that the language of H is E^* is that δ is defined for all pairs of states $x \in X$ and outputs $e \in E$. Consider the following subsets of sequences of observations of length n: $D_k = \{s \in E^n \mid \delta(x_0, s) = x_k\}$, $k = 1, 2, ..., |X|$. It is not hard to argue that D_k, where $k = 1, 2, ..., |X|$, form a partition of E^n.

For each $e \in E$, we can construct the binary transition matrix T_e of H, following the rule that if $\delta(x_i, e) = x_{i'}$, then $T_e(i', i) = 1$, otherwise $T_e(i', i) = 0$. This matrix captures all possible transitions from a state to another, under event e; since H is deterministic, T_e for $e \in E$ is a binary matrix with exactly a single "1" in each column. We can also define the binary column vector π'_0 to have a single nonzero element with value "1" at its i^{th} location, if $x_0 = x_i$ (in other words, π'_0 is an indicator vector for the initial state of H). With this notation at hand, $\delta(x_0, s) = x_k$ for $s = s[n]s[n-1]...s[1]$ is equivalent to $\pi'_n = \underbrace{T_{s[n]}T_{s[n-1]}...T_{s[1]}}_{T_s}\pi'_0$ being a vector with all zero entries except a single "1" at the k^{th} location. This is easy to establish by induction.

More generally, the entries of the matrix $T_s = T_{s[n]}T_{s[n-1]}...T_{s[1]}$ are such that $T_s(k, i) \in \{0, 1\}$ and $T_s(k, i) = 1$ if and only if $\delta(x_i, s) = x_k$. If we let the two vectors $c^{(j)} = P_j[1...1]$, of size $1 \times |Q^{(j)}|$ for $j = 1, 2$, we can show that the probability of error in Eq. (5) is smaller or equal to

$$\sum_{k=1}^{|X|} \min\{\sum_{s \in D_k} c^{(1)} A_s^{(1)} \pi_0^{(1)}, \sum_{s \in D_k} c^{(2)} A_s^{(2)} \pi_0^{(2)}\} \qquad (6)$$

where for $s = s[n]s[n-1]...s[1]$ we have $A_s^{(j)} \pi_0^{(j)} = A_{s[n]}^{(j)} A_{s[n-1]}^{(j)} ... A_{s[1]}^{(j)} \pi_0^{(j)}$. We now discuss how the above bound can be computed rather efficiently.

We define the matrix $\mathcal{A}^{(j)} = \sum_{e \in E} T_e \otimes A_e^{(j)}$, $j = 1, 2$, where $T_e \otimes A_e^{(j)}$ denotes the Kronecker product defined as the $(|X||Q^{(j)}|) \times (|X||Q^{(j)}|)$ matrix

$$\begin{bmatrix} T_e(1,1)A_e^{(j)} & T_e(1,2)A_e^{(j)} & \cdots & T_e(1,|X|)A_e^{(j)} \\ T_e(2,1)A_e^{(j)} & T_e(2,2)A_e^{(j)} & \cdots & T_e(2,|X|)A_e^{(j)} \\ \vdots & \vdots & \ddots & \vdots \\ T_e(|X|,1)A_e^{(j)} & T_e(|X|,2)A_e^{(j)} & \cdots & T_e(|X|,|X|)A_e^{(j)} \end{bmatrix}$$

Note that each $T_e(i', i)A_e^{(j)}$, x_i, $x_{i'} \in X$, is a matrix of size $(|Q^{(j)}|) \times (|Q^{(j)}|)$. We also define the (i', i) block of $\mathcal{A}^{(j)}$ as $\mathcal{A}^{(j)}(B_{i'}, B_i) = \mathcal{A}^{(j)}(b_i^{(j)} : f_i^{(j)}, b_{i'}^{(j)} : f_{i'}^{(j)})$, i.e., a $(|Q^j|) \times (|Q^j|)$ submatrix starting from row $b_i^{(j)} = (i-1)Q^{(j)} + 1$ to row $f_i^{(j)} = iQ^{(j)}$, and from column $b_{i'}^{(j)} = (i'-1)Q^{(j)}$ to column $f_{i'}^{(j)} = i'Q^{(j)}$. Letting $p_0^{(j)} = \pi_0' \otimes \pi_0^{(j)}$, we can write[2] (for $s = s[n]s[n-1]...s[1] \in E^n$)

$$\begin{aligned} p_n^{(j)} &= (\mathcal{A}^{(j)})^n p_0^{(j)} \\ &= \left(\sum_{e \in E} T_e \otimes A_e^{(j)}\right)^n (\pi_0' \otimes \pi_0^{(j)}) \\ &= \sum_{s \in E^n} (T_{s[n]}...T_{s[1]})\pi_0' \otimes (A_{s[n]}^{(j)}...A_{s[1]}^{(j)})\pi_0^{(j)} \\ &= \sum_{k=1}^{|X|} \sum_{s \in D_k} T_s \pi_0' \otimes \rho_{n,s}^{(j)} \\ &= \sum_{k=1}^{|X|} \sum_{s \in D_k} u_k \otimes \rho_{n,s}^{(j)} \\ &= \sum_{k=1}^{|X|} u_k \otimes \sum_{s \in D_k} \rho_{n,s}^{(j)} \ , \end{aligned}$$

where u_k is a column vector of size $|X| \times 1$, with zeros on all of its entries except a single one at its k^{th} entry, and $\rho_{n,s}^{(j)}$ is the vector $\rho_n^{(j)}$ for the sequence of observations s.

If we focus on the k^{th} block of $p_n^{(j)}$ of size $|Q^{(j)}| \times 1$ (i.e., entries $(k-1)Q^{(j)} + 1$ to $kQ^{(j)}$), we see that

$$p_n^{(j)}(B_k) = \sum_{s \in D_k} \rho_{n,s}^{(j)} = \sum_{s \in D_k} A_s^{(j)} \pi_0^{(j)} \ .$$

Following Eqs. (5) and the bound in (6), we can write

$$P(\text{error at } n) \leq \sum_{k=1}^{|X|} \min\{c^{(1)} p_n^{(1)}(B_k), c^{(2)} p_n^{(2)}(B_k)\} \ , \quad (7)$$

which can be used to compute an upper bound on the probability of error between the two systems ($S^{(1)}$ and $S^{(2)}$) by taking advantage of how the DFA H creates the partitions D_k, $k = 1, 2, ..., |X|$.

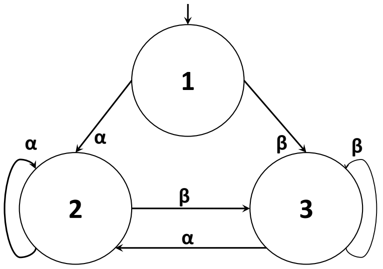

Fig. 3. DFA H_s for Example 2.

Example 2:

Consider the two HMMs in Fig. 2 and the DFA H_s in Fig. 3, with $X = \{1, 2, 3\}$, language $E^* = (\alpha + \beta)^*$, and initial state $x_0 = 1$ (which means that $\pi_0' = [1\ 0\ 0]^T$). Assume that the priors are $P_1 = 0.6$, $P_2 = 0.4$, so that

$$c^{(1)} = \left[\begin{array}{cc} 0.6 & 0.6 \end{array}\right], c^{(2)} = \left[\begin{array}{cc} 0.4 & 0.4 \end{array}\right],$$

and also that

$$\pi_0^{(1)} = \left[\begin{array}{cc} 1 & 0 \end{array}\right]^T, \pi_0^{(2)} = \left[\begin{array}{cc} 0.5 & 0.5 \end{array}\right]^T.$$

We create, according to the previous definitions, the matrices T_α, T_β for H_s as

$$T_\alpha = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}, T_\beta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

and obtain the matrices $\mathcal{A}^{(1)}$, $\mathcal{A}^{(2)}$ as

$$\mathcal{A}^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0.05 & 0 & 0.05 \\ 0 & 0.95 & 0 & 0.95 & 0 & 0.95 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix},$$

$$\mathcal{A}^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.95 & 0 & 0.95 & 0 & 0.95 \\ 0 & 0.05 & 0 & 0.05 & 0 & 0.05 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

Similarly, we obtain $p_0^{(j)} = \pi_0' \otimes \pi_0^{(j)}$, for $j = 1, 2$, as

$$p_0^{(1)} = \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right]^T,$$

$$p_0^{(2)} = \left[\begin{array}{cccccc} 0.5 & 0.5 & 0 & 0 & 0 & 0 \end{array}\right]^T.$$

For a sequence of observations of length n, we can write

$$P(\text{error at } n) \leq \sum_{i \in \{1,2,3\}} \min\{c^{(1)} p_n^{(1)}(B_i), c^{(2)} p_n^{(2)}(B_i)\} \ ,$$

[2] One of the properties of the Kronecker product is that $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ for matrices A, B, C, D of appropriate sizes [13].

where $p_n^{(j)} = \left(\mathcal{A}^{(j)}\right)^n \pi_0^{(j)}$, $j = 1, 2$. The plot of the bound as a function of n is provided in Fig. 4. As n becomes infinite, this bound stabilizes at 0.2349.

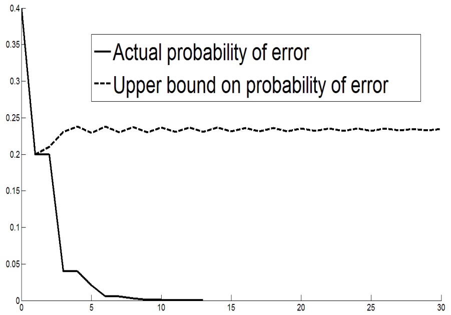

Fig. 4. Actual probability of error (continuous line) and upper bound (dashed line) with DFA H_s in Fig. 3.

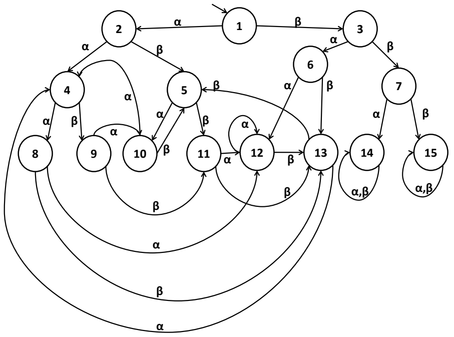

Fig. 5. DFA H in Example 3.

Example 3: We can extend the construction of the previous example to the larger DFA H in Fig. 5 with $X = \{1, 2, ..., 15\}$, language $E^* = (\alpha + \beta)^*$, and initial state $x_0 = 1$ (which means that $\pi_0' = [1\ 0\ 0\ \dots\ 0]^T$). We omit the details of the construction due to space considerations (but the steps are identical to the steps in Example 2).

The resulting upper bound on the probability of error is plotted in Fig. 6 as a function of the number of observations. As $n \to \infty$, we see that this upper bound tends to the constant value 0.0166. Note that this bound can perhaps be reduced by employing a DFA with more states and/or different transition functionality (to try and achieve a better partitioning of the set of possible sequences). In this particular example, in order to find this H, we tried all possible DFAs of 15 states, and presented the one that asymptotically results in the least upper bound.

IV. Connections to a Stochastic Diagnoser

We can reduce the number of states or even the size of all transition submatrices $A_e^{(j)}$, $j = 1, 2$, for each model ($S^{(1)}$, $S^{(2)}$) if we are able to remove all states that are not reachable under specific conditions (e.g., unreachability from a specific starting state). An example of such a deterministic finite automaton was the stochastic diagnoser introduced in [9], for the purpose of fault diagnosis. We describe this connection via the following example, where we use an appropriate DFA to create the stochastic diagnoser for the two models shown in Fig. 2.

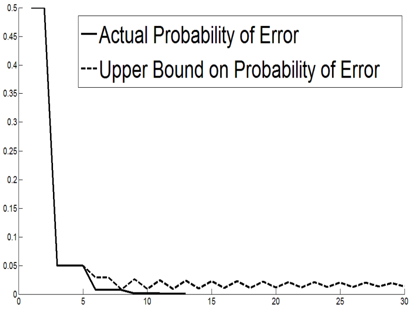

Fig. 6. Actual probability of error (continuous line) and upper bound (dashed line) with the DFA H in Fig. 5.

Example 4.

Suppose that the models in Fig. 2 capture the Normal (S_1) and Faulty (S_2) behaviour of a system. Also we define $Q^{(1)} = \{1N, 2N\}$, and $Q^{(2)} = \{1F, 2F\}$, with priors $P_1 = P_2 = 0.5$, and initial states, $q_0^{(1)} = \{1N\}$, $q_0^{(2)} = \{1F\}$. We want to find all transition matrices for the stochastic diagnoser, and relate them to the previous analysis (the original work in [9] uses the transpose of the matrices we use here). We analyze the system using the previous method, with the only difference being that the construction of the matrices A_e, $e \in E$, considers the behavior in each system simultaneously, e.g.,

$$A_b = \begin{bmatrix} \mathcal{A}_b^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathcal{A}_b^{(2)} \end{bmatrix} = \left[\begin{array}{ccccc} & 1N & 2N & 1F & 2F \\ 1N & 0 & 0 & 0 & 0 \\ 2N & 1 & 0 & 0 & 0 \\ 1F & 0 & 0 & 0 & 0 \\ 2F & 0 & 0 & 1 & 0 \end{array}\right].$$

If we only keep elements on nonzero rows and columns, we obtain the reduced matrix

$$A_b^{(s)} = \left[\begin{array}{ccc} & 1N & 1F \\ 2N & 1 & 0 \\ 2F & 0 & 1 \end{array}\right].$$

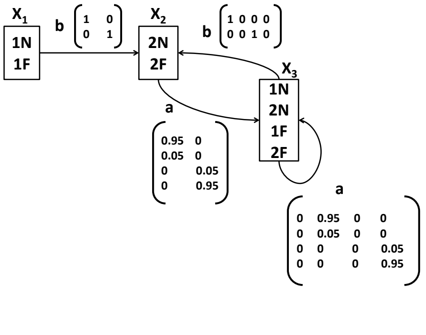

Fig. 7. Stochastic Diagnoser for $S^{(1)}$ and $S^{(2)}$.

Following this approach, we can create all possible different states and apply the reduced transition matrices. The stochastic diagnoser for our example is shown in Fig. 7. We can create the S matrix which includes all submatrices, according to each state $\{X_1, X_2, X_3\}$ (e.g., $S(1,7)$ captures the transition probability from state X_1^{1N} to X_3^{1F}). If the states are ordered as follows: state 1→ X_1^{1N}, state 2→ X_1^{1F}, state 3→ X_2^{1N},..., state 8 → X_3^{2F}, the matrix S is given by

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.95 & 0 & 0 & 0.95 & 0 & 0 \\ 0 & 0 & 0.05 & 0 & 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0 & 0 & 0 & 0.05 \\ 0 & 0 & 0 & 0.95 & 0 & 0 & 0 & 0.95 \end{bmatrix}.$$

Using S we can compute the upper bound of the probability of error as in the previous example (using, however, blocks of different sizes, due to the fact that entries that are zero in each block are dropped). Alternatively, we can use the automaton shown in Fig. 8 and follow the approach in the previous section to obtain $p_n^{(j)} = (\mathcal{A}^{(j)})^n p_0^{(j)}$. Note that by construction, a stochastic diagnoser checks if an output symbol is possible or not, so that the underlined symbols in Fig. 8 do not appear in the stochastic diagnoser in Fig. 7. For large n, we find the upper bound to be 0.2802.

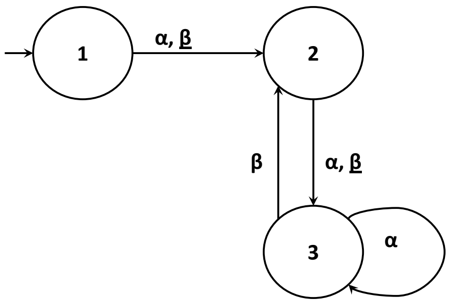

Fig. 8. Equivalent DFA to Stochastic Diagnoser in Example 4.

A probabilistic finite automaton that is AA-stochastically diagnosable [9] is essentially an automaton for which the probability of misclassification[3] goes to zero as the number of observations becomes asymptotically large. It is evident that our method can be used to establish whether the probability of misclassification goes to zero (by determining whether its upper bound goes to zero) using constructions quite distinct from a stochastic diagnoser. Thus, a sufficient condition for AA-stochastic diagnosability would be the existence of a DFA that leads to an upper bound on the probability of misclassification that goes to zero as the number of observations increases.

Remark: The complexity of computing the exact probability of error is an exponential function of n (it is of $O(n \times d^n \times (|Q^{(1)}|^2 + |Q^{(2)}|^2)))$. In obtaining the upper bound, we only require complexity linear in n (the complexity is of $O(n \times |X|^2 \times (|Q^{(1)}|^2 + |Q^{(2)}|^2)))$. In addition, for an arbitrarily large number of observations, we can compute the asymptotic upper bound with complexity of $O(|X|^3 \times (|Q^{(1)}|^3 + |Q^{(2)}|^3))$ by employing eigenvalue decomposition to obtain the steady-state of the Markov chains with transition matrices $\mathcal{A}^{(j)}$, $j = 1, 2$.

V. Conclusions

In this work we obtain an upper bound on the probability of error when classifying among two HMMs, based on a sequence of observations of length n. We use a specific class of DFAs to split the sequences of observations into different partitions and apply Markov chain theory to efficiently compute an upper bound on the *a priori* probability of misclassification among the two HMMs for sequences in each partition. The choice of DFA affects the partitioning which in turn affects the tightness of the upper bound. An open problem is the choice of a specific DFA (of a fixed number of states) that results in the least upper bound.

References

[1] E. Athanasopoulou and C. N. Hadjicostis, "Probability of error bounds for failure diagnosis and classification in hidden Markov models," in *Proceedings of the IEEE Conference on Decision and Control*, 2008, pp. 1477–1482.

[2] L. R. Rabiner, "Readings in speech recognition," A. Waibel and K.-F. Lee, Eds. Morgan Kaufmann Publishers Inc., 1990, ch. A tutorial on hidden Markov models and selected applications in speech recognition, pp. 267–296.

[3] F. Jelinek, *Statistical methods for speech recognition.* MIT Press, 1997.

[4] L. R. Bahl, F. Jelinek, and R. L. Mercer, "Readings in speech recognition," A. Waibel and K.-F. Lee, Eds. Morgan Kaufmann Publishers Inc., 1990, ch. A maximum likelihood approach to continuous speech recognition, pp. 308–319.

[5] K. S. Fu, *Syntactic Pattern Recognition and Applications.* Prentice-Hall, 1982.

[6] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, 1998.

[7] T. Koski, *Hidden Markov Models of Bioinformatics.* Kluwer Academic Publishers, 2001.

[8] J. Lunze and J. Schröder, "State observation and diagnosis of discrete event systems described by stochastic automata," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 11, no. 4, pp. 319–369, 2001.

[9] D. Thorsley and D. Teneketzis, "Diagnosability of stochastic discrete event systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 4, pp. 476–492, 2005.

[10] B.-H. Juang and L. Rabiner, "A probabilistic distance measure for hidden Markov models," *AT&T Technical Journal*, pp. 391–408, 1985.

[11] M. Falkhausen, H. Reininger, and D. Wolf, "Calculation of distance measures between hidden Markov models," in *Proc. Eurospeech*, 1995, pp. 1487–1490.

[12] S. M. E. Sahraeian and B. Yoon, "A novel low-complexity HMM similarity measure," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp. 87–90, 2010.

[13] P. A. Regalia and S. K. Mitra, "Kronecker products, unitary matrices and signal processing applications," *Society for Industrial and Applied Mathematics*, vol. 31, no. 4, pp. 586–613, December 1989.

[3] Strictly speaking AA-stochastic diagnosability is only concerned with faulty behavior that might be considered as non-faulty (and whether its probability goes to zero as the number of observations increases); thus, one should exclude the probability of misclassification that arises from strings generated by the non-faulty system that are more likely to have been generated by the faulty system.