Policy Analysis Framework for Conversational Biometrics

Upendra V. Chaudhari, Ganesh N. Ramaswamy

IBM T.J. Watson Research Center
Rt. 134, Yorktown Heights, NY 10598
{uvc,ganeshr}@us.ibm.com

Abstract

Modern speaker verification systems rely on a variety of input sources in making a decision on the validity of an identity claim. The meaning of the evidence these sources produce must be reconciled if robust decisions are to be made. In the case of Conversational Biometrics (CB), for example, this is typically accomplished via the specification of a verification policy implemented as a finite state machine. This paper presents a framework for the analysis of such complex systems guided by a policy finite state machine. The Receiver Operating Curve (ROC) associated with the acoustic speaker recognition task is transformed into a multi-dimensional Receiver Operating Map (ROM), which results from a probabilistic analysis of the policy state machine. A DCF Map can be similarly generated and we show results indicating that optimization over this surface is an appropriate way to set thresholds.

1. Introduction

Contemporary speaker recognition systems [1] [2] depend on a multiplicity of information sources which provide evidence for the assessment of speaker identity. Conversational Biometrics is one such system [1], and relies on a speaker’s acoustic characteristics as well as knowledge. Chief among the benefits of this approach are the ability to compensate for corruption of any one source and increased confidence in the result due to independent corroborative information [1] [3]. However, it is also possible that the various sources could provide contradictory evidence, which on the surface would make the results inconclusive. But, context may be able to disambiguate the results. Thus, to effectively use all of the information available, we must have a "policy" that guides the analysis. This paper presents a method to analyze such a policy within the context of a Conversational Biometrics verification architecture. A multi-dimensional Receiver Operating Map is generated as a transformation of the acoustic Receiver Operating Curve under the operation of the verification policy. Each dimension of the input vector represents a separate parameter, such as a threshold, and the output can be either the probability of ending up in the accept state or the reject state. In addition to the acoustic ROC data, the analysis of the policy requires estimates of the probability of incorrect answers to the posed questions, which are dependent on whether a target or non-target user is assumed. Optimization over the map can be used to set system parameters, such as the thresholds.

2. Policy Based Verification

Conversational Biometrics (CB) [1] combines speech and speaker recognition technologies to validate an identity claim. A CB session is a turn based directed dialog where claimant responses to posed questions and auxiliary information constitute a dynamic context based on which a decision is made, e.g. accept, reject, or continue to another question. The speech input is analyzed with respect to voiceprint match as well as knowledge (information content) match. Accordingly, speaker models have both an acoustic component and a knowledge profile. The knowledge profile can be represented as a table of questions and speaker specific answers. In general, however the profile can contain more information. The acoustic model is based on the GMM-UBM framework [4] [5]. The acoustic scores are based on the likelihood ratio statistic, whereas knowledge match is taken to be binary. It is possible to extend the knowledge match to indicate partial match, however, this case is not analyzed in the paper.

2.1. Policy Specification

The dialog in a CB session is guided by a verification policy, which can be represented as a finite state machine (Policy State Machine). The following assumptions are made: The Policy State Machine is defined by a set of states, each with a specified set of questions (or, more generally, topics) and possible transitions to be taken if the associated condition is satisfied. The transition conditions partition the decision variable space, and so only one condition can ever be valid. The conditions themselves are boolean expressions with intermediate values determined by expressions with relational and binary operators on the decision variables, which can represent quantities such as the number of correctly answered ques-
tions, the acoustic score, etc. Consider the example policy state machine specification shown in Figure 1. The states, except for ACCEPT and REJECT which are terminal, have topics which indicate their question pool and the transitions are labeled with conditions.

![Example Policy State Machine](image)

Figure 1: Example Policy State Machine

The expressions for the conditions use a number of variables and constants. There are three acoustic score thresholds: l, m, and h (low, medium, high). The variables are \( T \) = Number of topics covered before assessing the transition conditions, \( W \) = Number of topics covered for which the answer given was incorrect, and \( S \) = the current acoustic score. See section 3.1.1 for more details.

### 2.2. Conversational Biometrics Session

A CB session generates a path through the state machine, which consists of asking a question associated with a state (initially the start state) that has not been asked before on the path, obtaining the response and score, and processing the conditions associated to the state transitions in a sequential (or random) manner until one is satisfied. The corresponding transition is taken. The process is repeated at the next state, unless it is a terminal state, whereupon an accept or reject decision is made.

### 3. Policy Analysis: Path Enumeration

The use of a verification policy implies that the analysis of the overall system performance is not straightforward, because for example the different components may give contradictory evidence. Herein we develop a probabilistic analysis of the behavior of the policy state machine which affords a general view of system performance and which furthermore facilitates the tuning of parameters.

Let \( \mathcal{P} \) = set of all possible paths determined by the topographical structure of the Policy State Machine. Policy analysis first determines \( \mathcal{P}_{\text{sat}} \subset \mathcal{P} \) = subset of paths which are allowed (satisfiable, see section 3.1) by the transition conditions via a recursive procedure which starts in the initial state of the policy and follows paths depending on whether or not the conditions associated with the state transitions can be satisfied. Subsequently, each path is assigned a probability leading to the generation of the ROM (see section 4). Note that the transition conditions in Figure 1 ensure a finite number of paths.

#### 3.1. Satisfiability

To determine if a path condition is satisfiable, transition conditions are transformed into a set of linear constraints on variables. These constraints are propagated along the given path. At any point in the path, the linear constraints can be analyzed to determine whether or not a solution exists. A Linear Program is constructed from the set of constraints and solved. The feasible set is analyzed to determine the nature of the solutions possible for the program. If the feasible set is empty, then that sequence of transitions is not possible. If desired, the objective function for the Linear Program can be constructed to find the volume of scores and variable values that determine that path. In general the nature of the variables require a Mixed Integer (MI) Program.

#### 3.1.1. Variable Transformation

To facilitate the analysis, the variables in the transition conditions must be represented as sums of variables up to the current depth, which is equal to the number of turns completed in the dialog. Thus, \( T = \sum_{i=1}^{\text{depth}} q_i \), where \( q_i \) is the indicator function indicating if a question was asked at turn \( i \). \( W = \sum_{i=1}^{\text{depth}} w_i \), where \( w_i \) is the indicator function indicating if an incorrect answer was given at turn \( i \). \( S = \sum_{i=1}^{\text{depth}} \delta_i \), where \( \delta_i \) is the change in acoustic score for turn \( i \).

#### 3.1.2. Condition Transformation

Then, for example, a condition such as

\[
(T = 3) \& (W = 2) \& (S \leq 3.4) \quad (1)
\]

at depth = 3 (where \( \& \equiv \text{AND} \)) is transformed to

\[
\begin{align*}
-q_1 & + q_2 & + q_3 & & \leq 3 \\
-w_1 & + w_2 & + w_3 & & \leq 2 \\
-d_1 & + d_2 & + d_3 & & \leq 3.4 \\
\end{align*}
\]

1 \( \leq q_i \leq 1, \quad w_i \in \{0, 1\}, \quad LB \leq \delta_i \leq UB,

where \( LB \) and \( UB \) are lower and upper bounds derived from the acoustic ROC data. A long path will have many such combinations and the associated Linear (MI) Program will be large.

#### 3.1.3. Constraint Propagation

For a path to be viable, all conditions must be satisfied that are associated with the sequence of transitions needed to generate the path. We make this determination...
at any point in the path by transforming and propagating the constraints through the path to that point and solving the associated program. As an example, consider two segments of one possible path through the policy: Condition (1) yields the first transition, followed by condition (2) (if OR) for the second transition (at depth = 4).

\[(T - W \leq 2) \& (S \geq 4.0)] \quad | \quad (S \geq 5.1) \quad (2)\]

Combining (1) and (2) gives (OR and AND are used for emphasis):

\[(T = 3) \& (W = 2) \& (S \leq 3.4) \& \left( (T - W \leq 2) \& (S \geq 4.0) \right) \quad | \quad (S \geq 5.1)\]

which is the same as

\[(T = 3) \& (W = 2) \& (S \leq 3.4) \& (T - W \leq 2) \& (S \geq 4.0)\]

OR

\[(T = 3) \& (W = 2) \& (S \leq 3.4) \& (S \geq 5.1)\]

and which corresponds to the following set of ORed programs (bounds as before):

\[\begin{bmatrix} q_1 + q_2 + q_3 \\ -q_1 - q_2 - q_3 \\ w_1 + w_2 + w_3 \\ -w_1 - w_2 - w_3 + w_4 \\ -\delta_1 - \delta_2 - \delta_3 - \delta_4 \end{bmatrix} \leq \begin{bmatrix} 3 \\ -3 \\ 2 \\ 3.4 \\ -4.0 \end{bmatrix}\]

OR

\[\begin{bmatrix} q_1 + q_2 + q_4 \\ -q_1 - q_2 - q_4 \\ w_1 + w_2 + w_3 \\ -w_1 - w_2 - w_3 + w_4 \\ -\delta_1 - \delta_2 - \delta_3 + \delta_4 \end{bmatrix} \leq \begin{bmatrix} 3 \\ -3 \\ 2 \\ 3.4 \\ -5.1 \end{bmatrix}\]

4. Policy Analysis: Construction of ROM

One purpose of the ROC curve is to map a threshold to false accept and false reject rates. Here we define a ROM which is a function of the multiple thresholds, or in general the parameters, in the policy. The output can be either the accept rate, which is the probability of ending up in the ACCEPT state or the reject rate = 1 - accept rate, which is the probability of ending up in the REJECT state. The interpretation of the output of the map (i.e. whether it is the false accept rate or the false reject rate) is dependent on whether a target or non-target (imposter) session is assumed. The ROM surface is generated by calculating these values over a grid in a volume of threshold (parameter) values.

The first step is to associate with each condition, a probability (or density where appropriate) of occurrence by treating each variable in the policy as a random variable with a known or derived distribution. For example, let \(p(S)\) be the distribution of the score variable for a target model. It is used, along with the non-target distribution, in determining the acoustic only ROC curve. This is the distribution used on the first point in the path. At the second point, the distribution is conditioned on the transition condition that was satisfied to bring the system to the current state (point in the path). Here it is assumed that if the first condition contained the statement \(S > 1.1\), then the new score distribution is \(p(S|S > 1.1)\) which is easily derivable from \(p(S)\), etc. for subsequent path points. The distributions depend on the depth along the path and previous variable observations. Note that the analysis is simplified by expanding the policy state machine, replacing a transition whose condition has ORed components with a set of individual transitions for each component. For the present, assume that the acoustic score, \(T\), and \(W\) are conditionally independent given whether the session is target or non-target. The value of \(W\) is based on the (hypothized) probability of having a given piece of knowledge, i.e. the likelihood that a target will know the answer to a particular topic question as well as the likelihood that a non-target will know the answer. These may be determined, for example, via the difficulty of the questions. Since the transition conditions for each state partition the decision variable space, the sum of the probabilities of all allowable (satisfiable) paths from the start node to the ACCEPT and REJECT nodes is 1.

Given the probability assignments, the ROM can be specified. Let \(P_{\text{ACCEPT}} \subset P_{\text{SUB}}\) be the subset of paths that end in the ACCEPT state and \(P_{\text{REJECT}}\), the subset that end in the REJECT state. Let \(\mathbf{t}\) be a threshold vector defined by the grid. Then the specification of the ROM is the computation, for every \(\mathbf{t}\) in the grid, of \(\text{Prob}(P_{\text{ACCEPT}}|\mathbf{t}) = \text{the sum of the probabilities of all paths in } P_{\text{ACCEPT}} \text{ given } \mathbf{t}\).

5. Results

We illustrate via an example: Consider, again, the policy state machine specification shown in Figure 1. Define a grid of threshold values (see description of Figure 2 below) in 3 dimensions, one each for the low (l), medium (m), and high (h) score thresholds. For each threshold vector given by the grid, a probability of accept (1 - reject) is computed as outlined above. Figure 2 shows the resulting ROM surfaces (\(R^2 \mapsto R\), or accept rate as a function of m and h, where l is kept constant) for three different classes of users for the policy in Figure 1 (the h axis ranges from 1 to 8 corresponding to unit threshold increments from 0 up to 7, and the m axis runs from 1 to 6 corresponding to unit threshold increments from 0 up to 5, the low threshold is fixed at -0.5). The upper surface corresponds to targets, who are the correct speakers. The middle surface corresponds to informed imposters, i.e. impersonators who have gained access to the correct knowledge. The lowest surface corresponds to uninformed imposters, who are impersonators without correct knowledge. Each class of user corresponds to different probability assignments for the transition conditions, determined by the input score distribution, which is taken from real usage of the policy in Figure 1, and the input probability of errors for the questions for both targets and non-targets. The latter can be estimated from the perplexity of the questions, the speech recognition error
rate, etc. In this analysis, they were kept fixed at 0.05 for the targets and 0.95 for the non-targets, rates which were consistent with empirical results.

Figure 2: (ROM) Probability of Accept Surfaces \((l = -0.5)\)

The figure clearly highlights the fact that a poor choice of thresholds will make the system unusable, even if the components themselves perform well. Thus, we develop a method to properly set the thresholds: Given that we can compute the probabilities of accept and reject for targets and imposters, we can compute the values of the Detection Cost Function (DCF), a weighted combination of false accept and false reject rates, defined for the NIST Speaker Recognition Evaluations [6], as a function of the dimensions (thresholds). This generates a corresponding surface to the ROMs. Figure 3 shows a slice of the DCF surface at \(m = 3.0\) for both an informed and uninformed imposter. The slice is interpolated over the \(h\) axis on 36 equal spaced points from 1 to 7. The values to which the curves converge depend on the fixed thresholds and could be reduced by their adjustment. Note that as \(h\) is reduced, an informed imposter will readily be accepted. To understand the predictive power of this analysis, also shown as circles and squares are the locations of the hand tuned operating points for real test data generated during use of the policy. We see that optimization (minimization) along the DCF slices would yield performance very close to the hand tuned operating points. This remains true for other slices at different thresholds, suggesting that searching for the minimum point on the full DCF surface is a good way to automatically set the multiple thresholds. This range of thresholds is also indicated in figure 2 by the contrasting rectangles.

6. Discussion

We have presented a method to analytically and graphically assess the error rates associated with policy based verification procedures whose evolution is controlled by a state machine with transitions conditioned on the context of the process. The ROM was developed as a graphical means to view system performance and replaces the ROC for these complex verification systems. Evidence was presented to show that the analysis based tuning of threshold parameters agrees with actual performance data. Other applications, such as policy based dialog management systems may also benefit from the presented analysis methods.

7. References