SANDIA REPORT

SAND2004-4385
Unlimited Release
Printed October 2004

Markov Sequential Pattern Recognition: Dependency and the Unknown Class

Mark W. Koch, Greg B. Haschke, and Kevin T. Malone

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2004-4385
Unlimited Release
Printed October 2004

Markov Sequential Pattern Recognition: Dependency and the Unknown Class

Mark W. Koch
Signal and Image Processing Systems Department

Greg B. Haschke
Radar Analysis Department

Kevin T. Malone
Signal Processing Applications

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1163

Abstract

The sequential probability ratio test (SPRT) minimizes the expected number of observations to a decision and can solve problems in sequential pattern recognition. Some problems have dependencies between the observations, and Markov chains can model dependencies where the state occupancy probability is geometric. For a non-geometric process we show how to use the effective amount of independent information to modify the decision process, so that we can account for the remaining dependencies.

Along with dependencies between observations, a successful system needs to handle the unknown class in unconstrained environments. For example, in an acoustic pattern recognition problem any sound source not belonging to the target set is in the unknown class. We show how to incorporate goodness of fit (GOF) classifiers into the Markov SPRT, and determine the worse case nontarget model. We also develop a multiclass Markov SPRT using the GOF concept.

Index Terms**: Markov dependence, dependent observations, sequential pattern recognition, sequential probability ratio test, unknown class.**

Intentionally Left Blank

Contents

Figures

1 Acoustic Sensor System

This paper discusses the problem of distinguishing and classifying acoustic signatures using the sequential probability ratio test (SPRT) [27]. The acoustic system hardware typically consists of a microphone array, and analog to digital converters with a digital signal processor system for processing the data.

Figure 1 shows a diagram of a typical approach. The first block represents *sensor data processing and measurement formation*. Here, the analog acoustic signals are amplified, filtered, digitized, and broken into a series of overlapping time-slices. Next, the fast Fourier transform (FFT) magnitude of a time-slice for each acoustic channel is computed and averaged. This accounts for the microphone offsets and improves the signal to noise ratio.

The second block represents *feature extraction and target tracking*. Feature vectors are extracted from each time-slice based on source characteristics. Typical approaches may use wavelet based processing or spectral information. A tracking algorithm associates the feature vectors from a moving target with a track. Thus, for each track we have an associated list of feature vectors, where each list represents an *event*. The problem of extracting feature vectors and associating them to a track is a difficult problem in itself and beyond the scope of this paper. Here, we minimize the role of the tracker by testing with relatively clean data; we use data with only a single sound source and more than 30 feature vectors in a track

The final block represents *feature classification and target identification*. This is the block we will present in more detail. Here, a classifier takes an event and tries to identify the sequence as one of the possible known targets. We assume we have a sequence of feature vectors correctly associated to a track and we will concentrate on using the sequential probability ratio test (SPRT) to identify the sound source producing this sequence.

A variety of sensors using the approach in Figure 1 are conceivable. One such sensor may be part of a larger system processing data in non-real-time. Another variation might be a stand-alone version with the target tracking and identification algorithms running in real-time on the digital signal processor local to the sensor. Here the sensor has a local database containing targets of interest. At the end of an event the sensor could report to an operator if the moving object belongs to the database of known targets or if the object is unknown. The system could also report a "confidence" on its decision. Hardware constraints for stand-alone sensors may necessitate simplification of portions of the SPRT algorithm.

2 The Problem and Approach

We have an event ξ comprising of a sequence of features vectors $\mathbf{Y}(k)$ from time, $k = 1 \ldots n,$ and a set of *m* target classes $\theta_1,...,\theta_m$. We also have an unknown class denoted by θ_0. A local database contains template information $(\mathbf{M}_\theta, \mathbf{\Sigma}_\theta)$ describing the mean vector and covariance matrix of $\mathbf{Y}(k)$ for target classes θ. The target identification problem is to decide the class of ξ as $\theta_1,...,\theta_m$ or θ_0. It is important to note that we typically have a lot of information about $\theta_1,...,\theta_m$, but very little information about θ_0. Along with the problem of the unknown class, the target identification approach needs to handle dependencies between $\mathbf{Y}(k)$ and contamination of $\mathbf{Y}(k)$.

We use a goodness of fit (GOF) classifier to classify the $\mathbf{Y}(k)$ as belonging to a target of interest or not. The GOF allows us to control the errors from θ_0. For targets, we restrict the output of the GOF classifier to have a normal probability density function (PDF). While this may seem limiting, application of the central limit theorem allows us to satisfy this restriction. The normal PDF allows the use of power analysis [8], [22] to model θ_0. The output of the GOF

classifier $w(k)$ becomes our observations, and we then use a sequential probability ratio test (SPRT) [27], to combine the observations to make a decision. The SPRT minimizes the expected number of required observations to a decision [27].

We use Markov chains to describe the dependencies between $w(k)$. For the dependencies not completely described by a Markov process, we can use the effective number of independent observations [5] to modify the decision process and account for any remaining dependencies.

3 Goodness of Fit Classifiers: Controlling Out-of-Class Errors

In making any decision, we want to control two types of errors: *missed-detection* and *false alarm* errors. Missed-detection (MD) errors can result from missing a target signature by calling it a nontarget, and false alarm (FA) errors can result from alarming on a nontarget signature by calling it a target. Here, we take the viewpoint of a *one-class* classifier. In this viewpoint we are just interested in one specific target θ_1 represented by the alternative hypothesis, and the null hypothesis represents the non-target $\overline{\theta_1}$ class. If we have other targets of interest $\theta_2,...,\theta_m$ then we would design a one-class classifier for each of them. For the θ_1 one-class classifier we can divide the nontargets into two groups: $\theta_2,...,\theta_m$ and θ_0. This allows us to further distinguish between two types of false alarm errors: *between-class* and *out-of-class* errors. Between-class errors occur when alarming on another target $\theta_2,...,\theta_m$ by calling it the target θ_1. Out-of-class errors occur when alarming on an unknown signature θ_0 by calling it the target θ_1.

The unknown class, which causes the out-of-class errors, is a significant problem in real-world pattern recognition problems in unconstrained environments. For example, suppose we are developing a pattern recognition system to recognize a specific object with an imaging sensor. A

Bayesian classifier approach, while minimizing the between-class errors, would require models of the all the possible objects that could be imaged by the sensor to control the out-of-class errors. This model-the-whole-world approach is untenable for realistic systems.

Whereas Bayesian classifiers minimize the between-class error, they do nothing to control the out-of-class errors. Figure 2a illustrates this potential problem. The figure shows a two-dimensional feature space, with samples from two targets: target *A* represented by stars and target B represented by circles. Assuming normal distributions and equal covariance matrices for the targets, the Bayes decision boundary has a linear form. Whereas the Bayes classifier minimizes the between-class errors of the *A* and the *B* targets, it does not control the out-of-class errors caused by unknown objects represented by "*x*" symbols. Depending on which side of the boundary the nontarget falls, the classifier will assign the unknowns to one of the known classes and make 100% out-of-class errors.

Our approach utilizes goodness of fit (GOF) classifiers for dealing with an unknown class. A common GOF metric uses the Mahalanobis distance shown in the following equation:

$$d_\theta(k) = (\mathbf{Y}(k) - \mathbf{M}_\theta)' \boldsymbol{\Sigma}_\theta^{-1} (\mathbf{Y}(k) - \mathbf{M}_\theta) \qquad \textbf{(1)}$$

Distance classifiers allow us to differentiate a single class from all other classes with a template and a GOF metric. As shown in Figure 2b, large distances from the template will give decisions that the feature vector belongs to a nontarget class and small distances indicate feature vectors from the target class. Since the GOF classifier is not equivalent to a Bayesian classifier, the GOF classifier will not necessarily have minimal between-class errors.

The random variable $\mathbf{Y}(k)$, representing the feature vector from the k^{th} time slice, is assumed to have a multivariate normal distribution with a mean vector $\mathbf{M}$ and covariance matrix $\boldsymbol{\Sigma}$. Empirical results show this is a reasonable approximation for the feature vectors in our

acoustic identification problem. For our feature vectors, this gives the Mahalanobis distance (1) a chi-square distribution with ten degrees of freedom $v = 10$.

As discussed earlier we require the GOF output to have a normal PDF. We accomplish this by using a cube root transform $d_\theta^{1/3}(k)$ [17][29]. This approximation allows for the fast computation of the SPRT log likelihood ratio in the sensor's local processor and allows us to conveniently model the worst-case nontarget distribution. As we discussed in [18] the approximation is very good. Thus the GOF classifier for class θ uses the following test:

$$w_\theta(k) = \frac{d_\theta^{1/3}(k) - \mu_T}{\sigma_T} \quad \textbf{(2)}$$

where the parameters μ_T and σ_T represent the mean and standard deviation of $d_\theta^{1/3}(k)$ and are given by [17][29]:

$$\mu_T = \frac{(9v-2)}{9v^{2/3}}, \quad \text{and } \sigma_T = \frac{\sqrt{2}}{3v^{1/6}} \quad \textbf{(3)}$$

Here, v represents the number of degrees of freedom for the chi-square distribution.
The test $w_\theta(k)$ has a N(0,1) (normal distribution with mean 0 and standard deviation 1) distribution given observations from class θ.

4 SPRT for Pattern Recognition

The SPRT has been widely used for RADAR target detection [9], [16], [25], [28], and also for pattern recognition [20], [13] and multisensor fusion systems [2], [6], [15], [19] . For notational simplicity we assume we are interested in only one class $\theta_1 = T$. Later on we will extend our framework to multiple targets. We set hypothesis H_0 to the nontarget class (unknown) $\overline{T}$ and H_1 to the target class *T*. In our problem the observations $x_1, x_2, \ldots, x_n$ represent a multinomial transformation of the GOF output $w_T(k)$ **Error! Reference source not found.**. We will discuss

this transform in more detail in the next section. The Type I error α represents the probability of alarming on a nontarget, or the FA rate, and the Type II error β represents the probability of not detecting a target and calling it a nontarget, or the MD rate. In general, the random variables x_k have a PDF of $f(x_k \mid \theta)$, where θ represents the parameter we want to test (T or $\overline{T}$). For now, assuming x_k are independently and identically distributed (iid), Wald's test [27] computes the likelihood ratio:

$$\Lambda(n) = \prod_{i=1}^{n} \frac{f(x_i \mid T)}{f(x_i \mid \overline{T})} = \prod_{i=1}^{n} \lambda_i \qquad \textbf{(4)}$$

and makes a decision based on the constants *A* and *B* representing the upper and lower stopping boundaries respectively.

We find it more convenient to work in the log-likelihood space:

$$Z(n) = \log(\Lambda(n)) = \sum_{i=1}^{n} \log(\lambda_i) = \sum_{i=1}^{n} z_i, \qquad \textbf{(5)}$$

We call z_i the *weight of evidence*, and if $f(x_i \mid H_1) < f(x_i \mid H_0)$ giving $z_i < 0$ then we say x_i leads to *negative weight of evidence* for the target and if $f(x_i \mid H_1) > f(x_i \mid H_0)$ giving $z_i > 0$ we say x_i leads to *positive weight of evidence* for the target. The test then becomes:

$$\begin{array}{ll} \text{Reject } H_0 & \text{If } Z(n) \geq a \\ \text{Accept } H_0 & \text{If } Z(n) \leq b \\ \text{Get more data} & \text{If } b \leq Z(n) \leq a \end{array} \qquad \textbf{(6)}$$

where

$$a = \log(A) = \log\frac{1-\beta}{\alpha} \text{ and } b = \log(B) = \log\frac{\beta}{1-\alpha}. \qquad \textbf{(7)}$$

Since the test almost never ends exactly at the boundaries, the equations in (7) are only an approximation. Approaches for computing the exact boundaries exist, but require numerical integration [1], recursion [25], or a multinomial distribution [26].

5 Markov Modeling for Dependence

To determine the PDF of $w_T(k)$ we quantize $w_T(k)$ and then use a Markov chain to model the multinomial random variable. The reasons for this decision are twofold. Firstly, a Markov model allows us to model the dependencies between the observations. Secondly, the quantization produces decisions robust to outliers or contamination that can appear in the data. A Markov model is completely described by Q states, an a priori probability vector p_k, and transition matrix T_k.

For each H_k we have a Markov model with parameters subscripted by *k*. For Markov model *k*, the following equation gives p_k:

$$p_k = [\Pr_k(0) \ldots \Pr_k(Q_i) \ldots \Pr_k(Q-1)], \tag{8}$$

where $\Pr_k(Q_i)$ represents the a priori probability of being in state Q_i. The transition probabilities represent the probability of state Q_j following state Q_i in an event sequence or $\Pr_k(Q_j \mid Q_i)$. For Markov model *k*, the following equation gives the transition matrix T_k:

$$T_k = \begin{bmatrix} \Pr_k(0 \mid 0) & \ldots & \Pr_k(Q-1 \mid 0) \\ \vdots & & \vdots \\ & \Pr_k(Q_j \mid Q_i) & \\ \Pr_k(0 \mid Q-1) & \ldots & \Pr_k(Q-1 \mid Q-1) \end{bmatrix}. \tag{9}$$

The vector p_k and matrix T_k have the property that the sum of their rows equals one. For stationary processes we have the following relationship between the a priori probability vector p_k and T_k:

$$P_k = [1,1,\ldots,1]' p_k = \lim_{n\to\infty} T_k^n \quad \textbf{(10)}$$

where $[1,1,\ldots,1]'$ is a column vector of length Q. A first order Markov chain has the important property that:

$$\Pr(x(i) \mid x(1),\ldots x(i-1)) = \Pr(x(i) \mid (x(i-1)) . \quad \textbf{(11)}$$

In general, to quantize the test $w(k)$ into Q quantiles or states we need a set of $Q+1$ thresholds $\tau = \{\tau_0, \tau_1 \ldots, \tau_Q\}$. We use the convention that $\tau_0 = -\infty$ and $\tau_Q = +\infty$. The new quantized random variable $x(k)$ is given by:

$$x(k) = q, \text{ when } \tau_q < w(k) \le \tau_{q+1} . \quad \textbf{(12)}$$

To determine the quantization thresholds [26], we assume we have event training data and find the thresholds that maximize:

$$\max_{\tau} [E\{\boldsymbol{l} \mid H_1) - E\{\boldsymbol{l} \mid H_0\}] \quad \textbf{(13)}$$

where $\boldsymbol{l}$ is a random variable determined as follows:

$$l_q = \log\left[\frac{\Pr(x = q \mid H_1)}{\Pr(x = q \mid H_0)}\right], q = 0,\ldots,Q-1 . \quad \textbf{(14)}$$

This approach finds the thresholds that maximize the expected log-likelihood ratio difference between H_1 and H_0, and as a result we maximize the SPRT's ability to distinguish the two hypotheses. It assumes we are equally interested in H_1 and H_0. For Q quantiles, we can write (13) as:

$$\max_{\tau}\left\{\sum_{q=0}^{Q-1}[\Pr(l_q \mid H_1) - \Pr(l_q \mid H_0)]l_q\right\}. \tag{15}$$

6 Markov SPRT

Using (4) and (11) the SPRT likelihood ratio for Markov dependence [7] [10] and a sequence of quantized GOF scores $x(i)$ $(i = 1\ldots n)$ becomes:

$$\Lambda(n) = \prod_{i=1}^{n}\frac{f\left(x(i) \mid H_1\right)}{f\left(x(i) \mid H_0\right)} = \frac{\Pr_1\left(x(1)\right)}{\Pr_0\left(x(1)\right)}\prod_{i=2}^{n}\frac{\Pr_1\left(x(i) \mid x(i-1)\right)}{\Pr_0\left(x(i) \mid x(i-1)\right)} = \prod_{i=1}^{n}\lambda(i). \tag{16}$$

The log likelihood ratio for the Markov model is:

$$Z(n) = \log(\Lambda(n)) = \log\left[\frac{\Pr_1\left(x(1)\right)}{\Pr_0\left(x(1)\right)}\right] + \sum_{i=2}^{n}\log\left[\frac{\Pr_1\left(x(i) \mid x(i-1)\right)}{\Pr_0\left(x(i) \mid x(i-1)\right)}\right] = \sum_{i=1}^{n}\log(\lambda(i)) = \sum_{i=1}^{n} z(i). \tag{17}$$

From equations (8), (9), and (17) we can define the weights of evidence vector and matrix. The weight of evidence vector e_0 comes from the term:

$$\log\left[\frac{\Pr_1\left(x(1)\right)}{\Pr_0\left(x(1)\right)}\right] \tag{18}$$

in equation (17). The following equation gives the i^{th} element of e_0:

$$e_0(i) = \log\left[\frac{p_1(i)}{p_0(i)}\right]. \tag{19}$$

The weight of evidence matrix E_1 comes from the term:

$$\log\left[\frac{\Pr_1\left(x(i) \mid x(i-1)\right)}{\Pr_0\left(x(i) \mid x(i-1)\right)}\right] \tag{20}$$

in equation (17). The following equation gives the (i,j) element of E_1:

$$E_1(i, j) = \log\left[\frac{T_1(i, j)}{T_0(i, j)}\right]. \tag{21}$$

Given an initial state x_1 or a transition from state x_i to x_j we can respectively use e_0 or E_1 as a table lookup for the amount of evidence to accumulate in favor of a target.

7 Modeling the Nontarget Class

In most pattern recognition problems in an unconstrained environment, we typically have a lot of information on the target class, but very little information on the nontarget class. Here, the target has a simple hypothesis, but the nontarget class usually requires a composite hypothesis.

One approach for handling the composite hypothesis models every object that will be sensed by the sensor. Although such an approach would, in theory, produce an optimal classifier, this approach of modeling the "whole world" is often untenable. Our approach for modeling the nontarget class determines the *worse case nontarget* distribution $f^*(w|\overline{T})$. The approach has some similarities to that taken by [14] for modeling composite hypotheses by determining the *least favorable choice*. We can then use the worse case nontarget to determine the quantization thresholds using (13) and ultimately the evidence vector (19) and matrix (21).

If we have a pool of close nontargets we can determine the least favorable choice by finding the nontarget distribution that is closest to the target distribution and use that nontarget as $f^*(w|\overline{T})$. Otherwise we use *statistical power analysis* [8], [22] to find $f^*(w|\overline{T})$. We now discuss each method in turn.

7.1 Worse Case Nontarget from a Pool of Nontargets

Here, for each target we have a pool of nontargets and we would like to have a similarity metric to determine which nontarget is closest to the target. Rabiner [23] suggests the following for the distance between two Markov models:

$$D_{01} = \frac{E\{\boldsymbol{z}_1\} - E\{\boldsymbol{z}_0\}}{2}. \qquad \textbf{(22)}$$

We can compute $E\{\mathbf{z}_k\}$ directly using the following equation:

$$E\{\mathbf{z}_k\} = \sum_{i=1}^{Q}\sum_{j=1}^{Q} E_1(i,j)T_k(i,j)p_k(j). \qquad \textbf{(23)}$$

For the target classifier, this distance D_{01} is essentially the difference between the average weight of evidence given a target and the average weight of evidence given a nontarget. The smaller this difference the closer the target and nontarget become. The target that minimizes (22) is the worst case nontarget.

7.2 Worse Case Nontarget using Power Analysis

Power analysis considers a hypothesis test whose null hypothesis of a procedure having *no-effect* has a known distribution. Usually, we do not know the distribution of the alternative hypothesis of the procedure having some effect. Here, we cannot compute the Type II error or the probability of accepting the no-effect hypothesis when no-effect hypothesis is false.

Power analysis assumes the tested effect is linear and the measured effect size (small, medium or large) is known. Typically, power analysis allows the statistician to determine if enough samples were collected to give the test a high power $1-\beta$.

Let $f(w|T) = g_{\mathbf{w}}(w)$ represent the distribution of our GOF random variable $\mathbf{w}$ given a target hypothesis T. This is similar to the no-effect hypothesis in power analysis. Assuming a linear model with parameters (a_0, a_1) the GOF for the nontarget is $a_0 + a_1\mathbf{w}$ and the distribution for the nontarget hypothesis $\overline{T}$ is $f(w|\overline{T}) = \frac{1}{a_1} g_{\mathbf{w}}\left(\frac{w-a_0}{a_1}\right)$. We assume $a_0 \geq 0$ and $a_1 > 0$, since small $\mathbf{w}'s$ point to a target hypothesis. We define the worse case nontarget as the one that maximizes the expected number of observations to a decision given a nontarget.

Theorem 1. *Let* $f(w|T) = g_{\mathbf{w}}(w)$ *represent the target distribution and*

$$f(w|\overline{T}) = \frac{1}{a_1} g_{\mathbf{w}}\left(\frac{w - a_0}{a_1}\right)$$ *represent the nontarget distribution using power analysis.*

When $a_1 = 1$ *and* $g_{\mathbf{w}}(w) \sim N(0,1)$ *we have the worse case nontarget, since it that maximizes the expected number of observations to a decision given a nontarget.*

Proof. See Appendix B in [18].

With $a_1 = 1$ and a normal distribution the worse case nontarget $f^*(w|\overline{T})$ is $N(a_0,1)$ where the location parameter a_0 gives the effect size. For the nontarget hypothesis, the location parameter $a_0 = \mu_N$ represents the smallest acceptable effective difference between the target and nontarget. For a signature with $\mu < \mu_N$ we accept that the target and signature are so close that the errors we make have no practical consequence and this preference increases with decreasing μ. For a signature with $\mu > \mu_N$ we call this a nontarget and this preference increases with increasing μ.

Using $f^*(w|\overline{T})$ we can determine τ and p_0, but we still need T_0 for the Markov model of H_0. We define a worst case nontarget T_0 as the Markov model with the smallest D_{01} (22). From (21), (22), and (23) we get the following objective function:

$$J = \sum_{i=1}^{Q}\sum_{j=1}^{Q} \log\left(\frac{T_1(i,j)}{T_0(i,j)}\right) T_1(i,j) p_1(j) - \sum_{i=1}^{Q}\sum_{j=1}^{Q} \log\left(\frac{T_1(i,j)}{T_0(i,j)}\right) T_0(i,j) p_0(j). \quad \textbf{(24)}$$

We also have constraints on T_0:

$$0 \le T_0(i,j) \le 1, \quad \sum_{j=1}^{Q} T_0(i,j) = 1, \quad \lim_{n\to\infty} T_0^n = P_0, \quad \textbf{(25)}$$

where $P_0 = [1,1,\ldots,1]' p_0$ (10). To enforce the limit constraint between T_0 and p_0, we use the expansion of T_k in terms of *spectral* matrices [3]:

$$T_k = \sum_{i=1}^{Q} \gamma_k(i) S_k(i) \tag{26}$$

Here $\gamma_k(i)$ represents the eigenvalues ordered largest to smallest of T_k and the spectral matrices $S_k(i) = v_k(i) u_k'(i)$ consist of the right eigenvectors $v_k(i)$ and the left eigenvectors $u_k'(i)$. Spectral expansions have the following properties [3]:

$$\gamma_1 = 1,\ |\gamma_k| < 1 \text{ for } k > 1,\ S_i S_j = \begin{cases} 0 & i \neq j \\ S_i & i = j \end{cases}, \text{ and } \sum_{i=1}^{Q} S_i = I \tag{27}$$

Using these properties and the spectral expansion of T_0, the limit constraint becomes:

$$\begin{aligned} &\lim_{n\to\infty} T_0^n = P_0 \\ &\lim_{n\to\infty} \sum_{i=1}^{Q} \gamma_0^n(i) S_0(i) = P_0 \\ &S_0(1) = P_0 \end{aligned} \tag{28}$$

Thus, we need to make P_0 a spectral matrix of T_0. Using the spectral expansion properties we can enforce this by introducing the following constraint:

$$P_0 T_0 = P_0 . \tag{29}$$

The constraints become:

$$0 \le T_0(i,j) \le 1,\ \sum_{j=1}^{Q} T_0(i,j) = 1,\ \ P_0 T_0 = P_0 . \tag{30}$$

We solve this optimization problem (24) with the constraints (30) numerically using Matlab's nonlinear optimization routine for equality and inequality constraints.

As an example, consider $Q = 2$ with

$$T_k = \begin{bmatrix} \psi_k & 1-\psi_k \\ 1-\zeta_k & \zeta_k \end{bmatrix}. \tag{31}$$

To satisfy the first two constraints on T_0 we require $0 \le \psi_k, \zeta_k \le 1$. We assume we know T_1 and we want to determine T_0 or find the ψ_0 and ζ_0 that minimizes (24). To enforce the third constrain on T_0 we first perform a spectral decomposition for T_k:

$$T_k = \frac{1}{\psi_k + \zeta_k - 2}\begin{bmatrix} \zeta_k - 1 & \psi_k - 1 \\ \zeta_k - 1 & \psi_k - 1 \end{bmatrix} + \frac{\psi_k + \zeta_k - 1}{\psi_k + \zeta_k - 2}\begin{bmatrix} \psi_k - 1 & 1-\psi_k \\ 1-\zeta_k & \zeta_k - 1 \end{bmatrix}. \tag{32}$$

For stationary processes the first term of the spectral decomposition gives p_k, and thus to satisfy the third constraint on T_0 we require:

$$p_k = \frac{1}{\psi_k + \zeta_k - 2}\begin{bmatrix} \zeta_k - 1 & \psi_k - 1 \end{bmatrix}. \tag{33}$$

Assuming a target distribution of $f(w|T) = N(0,1)$, power analysis gives $f(w|\overline{T}) = N(\mu_N, 1)$. For $Q = 2$ and using (15), we get $\tau = \left\{-\infty, \frac{1}{2}\mu_N, \infty\right\}$. The Gaussian distributions combined with the optimal quantization thresholds τ gives $p_0(1) = p_0(2)$. Thus using (33):

$$\frac{\zeta_0 - 1}{\psi_0 + \zeta_0 - 2} = \frac{\psi_1 - 1}{\psi_1 + \zeta_1 - 2}, \tag{34}$$

and solving for ψ_0:

$$\psi_0 = \frac{\psi_1 - \zeta_0 - \zeta_1 + \zeta_0\zeta_1}{\psi_1 - 1}. \tag{35}$$

Since we assume we know the target transition matrix ψ_1 and ζ_1, we just need to find the ζ_0 that minimizes *J* (24). Using (33) and (32) the following equation gives *J* as:

$$J(\psi_0, \zeta_0, \psi_1, \zeta_1) = \frac{1}{\psi_1 + \zeta_1 - 2}[j_{11} + j_{12} + j_{21} + j_{22}]. \tag{36}$$

where

$$
\begin{aligned}
j_{11} &= [\psi_1(\zeta_0 - 1) - \zeta_0 + 2\zeta_1 - \zeta_1^2][\log(1-\zeta_1) - \log(1-\zeta_0)] \\
j_{12} &= [\psi_0 + \psi_1(\psi_1 - 2) + \zeta_1 - \psi_0\zeta_1]\left[\log\left(\frac{(\zeta_0 - 1)(\zeta_1 - 1)}{1-\psi_1}\right) - \log(1-\psi_1)\right] \\
j_{21} &= [\zeta_0(\zeta_1 - 1) + \zeta_1 - \psi_1\zeta_1][\log(\zeta_0) - \log(\zeta_1)] \\
j_{22} &= [\psi_0 - \psi_0\psi_1 + \psi_1(\zeta_1 - 1)]\left[\log(\psi_1) - \log\left(\frac{\psi_1 - \zeta_0 + \zeta_1(\zeta_0 - 1)}{\psi_1 - 1}\right)\right]
\end{aligned}
\qquad \textbf{(37)}
$$

We can minimize (36) numerically for ζ_0 given ψ_1 and ζ_1 and using (35) for ψ_0.

8 Estimating Markov Parameters

The weights of evidence and thereby a classifier for target T are defined by the quintuple $(\tau, p_1, T_1, p_0, T_0)$. The parameters p_1 and T_1 are estimated from sequences of GOF scores from the target T using the quantization thresholds from the set τ. The parameter p_0 is estimated using the same quantization thresholds, except now we use GOF scores from the worst case nontarget $\overline{T}$. For T_0, we can use the sequences from the worst case nontarget, if available, or minimize the objective function (24). We estimate the a priori probability vector p_k in the usual way; we count the numbers of each state in training sequences for hypothesis *k* and then normalize p_k, so that its row sum equals one. The transition matrix T_k is estimate similarly, but instead we count numbers of each possible transition and again apply the constraint that the rows should sum to one.

One problem that occurs in the estimation process is insufficient training data. Here, we have an insufficient number of states occurrences to get good estimates of model parameters. Increasing numbers of states or large distances between the target and worst case nontarget exasperates this problem. The problem becomes intolerable when there is no occurrence of a state and the weights of evidence (19) (21) go to $\pm\infty$. Since it is often impractical to increase the

training set size, we add extra constraints to the model parameters to insure that no model parameter falls below a specified level p_ε. Thus we require estimates of $\Pr_k(Q_i)$ and $\Pr_k(Q_j \mid Q_i)$ to have values greater than p_ε. If this constraint is violated then we set the offending values to p_ε and renormalize so that the row sums equal one. For our results we use $p_\varepsilon = 5 \times 10^{-3}$. Such post-processing techniques have been applied to problems in speech recognition [23].

9 Effective Number of Independent Observations: Handling Feature Dependence

Unfortunately the Markov model doesn't always account for all the dependencies present in the data. For a Markov model, the state occupancy duration δ has a probability function of geometric: $\Pr(\delta = n) = p_\delta^{n-1}(1 - p_\delta)$ where p_δ is the probability of remaining in the same state. If the system stays in a state, on average, longer than expected then the Markov model will not completely account for all the dependencies and the number of errors will be larger than expected. Approaches to overcome this limitation model the true $\Pr(\delta = n)$, for example semi-Markov models [24] explicitly characterize the state occupancy probabilities. Determining the form of these models and their parameters is difficult without a lot of training data. Instead, we estimate the remaining dependency and use that to modify the SPRT decision process.

Here we build an SPRT assuming Markov dependence and then adjust the design, so the relationship between the error rates and the decision boundaries (7) is maintained. Let Ω_t represent a set of actual training data and Ω_s represent a set of Monte Carlo simulation data using the appropriate Markov parameters. If the Markov model does not adequately describe the

dependence, then we expect $\mathrm{Var}\{Z(n)|\Omega_t, H_j\}$ to be larger than the $\mathrm{Var}\{Z(n)|\Omega_s, H_j\}$, and we use the ratio

$$\kappa_{\mathrm{d}} = \max\left(1, \frac{\mathrm{Var}\{Z(n)|\Omega_t, H_1\}}{\mathrm{Var}\{Z(n)|\Omega_s, H_1\}}\right) \quad \textbf{(38)}$$

to modify the SPRT decision process. The ratio is constant if we assume the unexplained dependence results from a serially correlated, and weakly stationary process [5]. The modification to the SPRT is presented in the following theorem:

Theorem 2. *To handle dependency not completely describe by the Markov model we modify the SPRT decision boundaries. The new decision boundaries* a' *and* b' *become* $a' = \kappa_d\, a$ *and* $b' = \kappa_d\, b$.

Proof. See Appendix A in [18].

10 Multiclass SPRT

There are many nonoptimal approaches for handling sequential testing of multiple hypotheses [12], [16]. Using a Bayesian sequential decision procedure, a multiclass SPRT (MSPRT) has been shown to be optimal in the average number of observations [4] [11], but requires knowledge of classes' a priori probabilities. Unfortunately, we could not directly translate any of the preceding approaches to use the unknown class. Similar to the MSPRT approach we define a multiclass SPRT using the SPRT one-class classifier formulation, but cannot make any claims to optimality. For each known class θ_i $(i = 1, \ldots, m)$ we have a SPRT one-class classifier and a corresponding cumulative log-likelihood ratio $Z_i(n)$, with $Z_i(n)$ measuring a pattern's log-likelihood of belonging to θ_i and $\overline{\theta}_i$ (not class θ_i). For classifier θ_i, we define

$\alpha_i = \Pr(\text{Decide } \theta_i \mid \overline{\theta}_i)$ and $\beta_i = \Pr(\text{Decide } \overline{\theta}_i \mid \theta_i)$, and we also use θ_0 to represent the unknown class. We propose the following multiclass decision:

$$\begin{array}{ll} \text{Decide class } \theta_i & \begin{cases} \text{For the first } Z_i(n) \geq a_i \\ \text{If } Z_i(n) \leq b_i \quad \forall i = 1,\ldots,m. \\ \text{Otherwise} \end{cases} \\ \text{Decide class } \theta_0 & \\ \text{Get more data} & \end{array} \tag{39}$$

The decision boundaries a_i and b_i for class θ_i are determined by α_i and β_i (7).

Theorem 3. *For class θ_i the resulting errors α_i' and β_i' from using the multiclass decision (39) are bounded by the original SPRT one-class error rates α_i and β_i.*

Proof. See Appendix C in [18].

11 Computation of Confidences: Terminating the Test

Due to the sequential nature of the test we cannot guarantee we will have enough data to make a decision. Instead of using forced termination techniques [7],[13], [21], [28] we can define the confidence of the system's largest absolute response. Also, signal/image analysts would like to have a single number representing the *confidence* of the system's decision. Intuitively, we can imagine shrinking the a and b decision boundaries towards zero until a decision is made. This shrinkage can be accomplished by making the error rates α and β equal to γ. Adjusting γ from its initially small value to 0.5 can shrink the boundaries toward zero (7). Mathematically, we define confidence as $1-\xi$ where ξ represents the probability of error. Here, the errors associated with the final decision boundary are inversely related to the confidence. For the SPRT:

$$\xi = P(error) = P(error \mid H_0)P(H_0) + P(error \mid H_1)P(H_1) = \alpha P(H_0) + \beta P(H_1). \tag{40}$$

Assuming equal a priori probabilities and equal error rates $\alpha = \beta = \gamma$, the confidence ϕ is

$$\phi = 1 - \gamma. \tag{41}$$

We define 5 levels of confidence at 0.99, 0.95, 0.85, 0.75, and 0.5. The minimal 0.5 confidence level forces the SPRT to make a decision (a=b=0) and has high associated errors. As our confidence increases the associated errors decrease.

12 Data, Testing, and Results

To test our algorithm, we use data collected over many different experiments and in realistic conditions of three sound sources moving by an acoustic sensor. For this testing, the role of the tracker is minimized by testing with relatively clean data; we use data with only single sound sources and more than 30 feature vectors in a track. This data provides a difficult test for our approach. We distinguish the targets with three symbols: T_1, T_2, and T_3. The T_1 and T_2 targets have similar acoustic signatures, making them a difficult test case for the identification algorithm. The T_1 target has a larger feature variance causing larger errors. The T_3 target is fairly distinct and requires power analysis to determine its worse case nontarget. We use the tracking algorithm to give a list of feature vectors for each target event. We also have experiments with nuisance sound sources gathered by the sensor. The letter U denotes these sources. This data is assigned to the unknown class and is used to test the unknown class rejection capabilities of the SPRT classifiers.

12.1 SPRT Results

Now we discuss the results of T_1, T_2, and T_3 target identification based on the SPRT. All the results are based on using four quantiles (Q=4) to model the GOF outputs as a Markov process. The κ_d for T_1, T_2, and T_3 has been estimated as 4, 6, and 5 respectively. The worse case nontarget for T_1 is T_2 and vice versa. The worse case nontarget for T_3 is determined based on power analysis using μ_n=3. First we take the classifiers for each target separately and perform

an analysis for each assuming a GOF classifier. Then we put the three together and analyze them in a multiclass system.

12.1.1 Model Verification

To check the models used for the SPRT, we use *verification plots* by selecting a theoretical error, γ, and computing the (A,B) decision boundaries using $\alpha = \beta = \gamma$ and equation (7). Using these decision boundaries we estimate the actual error and then plot the theoretical vs. the actual error for a range of γ values. While this doesn't cover all the possible combinations of possible (α,β) errors, it does cover the small errors since $A \approx 1/\alpha$ and $B \approx 1/\beta$ for $\alpha,\beta << 1$.

In Figure 3 we show the verification plots for the three GOF classifiers and the multiclass classifier. It's important to note that the verification plots are based on all the possible thresholds and thus errors for each target event. Figure 3a shows the analysis of the T_1 GOF classifier. The diagonal line shows the theoretical error matching the actual error. The four curves show the results of putting the data streams from the three targets and the unknown into the T_1 GOF classifier. The dotted bars show the 2σ confidence interval for the error estimates. In this plot, the theoretical error approximates or bounds the actual error. This gives an indication that we have reasonable models for approximating the target and nontarget distributions. Figures 3b and 3c show the same results for the T_2 and T_3 classifiers respectively. In both cases the theoretical error clearly bounds the actual measured errors. Figure 3d shows the verification plot for the multiclass system. Again the actual errors are bounded by the SPRT theoretical errors.

12.1.2 Performance Characteristics

Figure 4 shows the performance plot for each classifier. Each plot graphs the probability of making an incorrect p_i call vs. the probability of making a correct call p_c at different levels of confidence for each target and the unknown. The error rates (p_i and $1-p_c$) shown in Figure 4

are different than the error rates shown in Figure 3. The error rates in Figure 4 are based on only those target events in which the classifier was able to make a call.

At any point on the performance curve the probability of no-call p_{nc} is $1 - p_c - p_i$. Here, the no-call rates represent the percentage of targets or nontargets that have final and intermediate SPRT values between the thresholds *A* and *B*. Recall, we can only make a call of target if the likelihood ratio goes above *A* or a call of nontarget if the likelihood ratio goes below *B*. The diagonal line shows when $p_{nc} = 0$ or when the system is forced to make a call. Here the confidence is 0.5. The markers on the curve indicate different levels of confidence. There are five markers, one for each level of confidence given in Section 11. In general, we can decrease the no-call rate by decreasing our desired confidence, but the price we pay is an increased error p_i. Thus as we travel up the curves p_c increases, but the confidence decreases.

Figure 4a shows the performance results for the T_1 GOF Classifier. The large no-call rates at high levels of confidence for all the sound sources indicate the high variability in of the T_1*'s* features. Figure 4b shows the results for the T_2 GOF Classifier. The high no-call rates for the T_1 target at high confidences shows the difficulty of distinguishing the T_1 and T_2 targets. Figure 4c shows the results for the T_3 GOF Classifier. The fast drop off of no-call rates and the small error rates show that the T_3 target has a significantly different signature from the T_1 and T_2 targets. For the T_1 or T_2 GOF classifier the operating condition must be selected carefully. Requiring too large of a confidence would result in too many no-calls, and a small confidence may produce too many errors. Depending on the operation scenario it may be possible to combine the two classes. Improvements in the sensor, feature extraction algorithms, or the incorporation of another modality should improve the no-call performance at the higher confidence levels. Figure 4d shows the result of combining the GOF classifiers into a multiclass

classifier. In general, adding multiple classifiers improves the performance over the one-class classifier. For example, suppose the T_1 classifier alarms on a T_2 event. The T_2 classifier might call the event a target before the T_1 classifier can false alarm on it.

13 Conclusion

We have applied the Markov SPRT and its extensions to a difficult problem of identifying acoustic signatures. The SPRT can take a stream of observations and classify it as a target or a nontarget. The Markov property allows the SPRT to handle dependent observations, where the state occupancy probability is geometric. For a non-geometric process we show how to use the effective amount of independent information to modify the decision process, so that we can account for the remaining dependencies.

The desired error rates determine the SPRT's upper and lower decision boundaries. From this property we develop a method of computing the confidence of a decision. We also use power analysis to develop statistical models of the worst-case nontarget class. This approach does not require training with every possible nontarget that will move by the sensor. Results show a viable system with statistical analysis allowing a user to understand the tradeoffs in determining the system's operational concept.

Acknowledgements

For M. Koch's early work in evidence accrual, he would like to acknowledge the support of Eddie Hoover at Sandia National Laboratories. M. Koch would also like to acknowledge José Salazar and David Harmony at Sandia National Laboratories for their many hours of discussions on the SPRT.

References

[1] G. E. Albert, “On the Computation of the Sampling Characteristics of a General Class of Sequential Decision Problems,” Ann. Math. Stats., vol. 25, pp. 340-356, 1954.

[2] M. M. Al-Ibrahim and P. K. Varshney, “A Simple Multi-Sensor Sequential Detection Procedure,” Proceedings of the 27th Conference on Decision and Control, Austin Texas, pp. 2479-2483, December 1988.

[3] M. S. Bartlett, *An Introduction to Stochastic Processes with Special Reference to methods and Applications*, New York: Cambridge University Press, pp. 24-28, 1978.

[4] C. W. Baum and V. V. Veeravalli, “A Sequential Procedure for Multihypothesis Testing,” *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 1994-2007, 1994.

[5] G. V. Bayley and J. M. Hammersley, "The Effective Number of Independent Observations in an Autocorrelated Time Series," *Supplement to the Journal of the Royal Statistical Society*, vol. 8, pp. 184-197, 1946.

[6] R. S. Blum, S. A. Kassam, and H. V. Poor, “Distribution Detection with Multiple Sensors: Part II – Advanced Topics,” Proceedings of IEEE, vol. 85, no. 1, pp. 64-79, 1997.

[7] R. Chandramouli and R. Ranganathan, “A Generalized Sequential Sign Detector for Binary Hypothesis Testing,” *IEEE Signal Processing Letters*, vol. 5, no. 11, 1998.

[7] Y. T. Chien and K. S. Fu, “A Modified Sequential Recognition Machine using Time-Varying Stopping Boundaries,” *IEEE Transactions on Information Theory*, vol. 12, no. 2., pp. 206-214, 1996.

[8] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. New Jersey: Lawrence Erlbaum Associates, 1988.

[9] G. Corsini, E. D. Mese, G. Marchetti, L. Verrazzani, “Design of the SPRT for RADAR target detection,” IEE Proceedings, vol. 132, pt. F., no. 3, pp. 139-148, 1985.

[10] B. Dimitriadis, D. Kazakos, “A nonparametric sequential test for data with Markov dependence,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 19, no. 3, pp. 338-346, 1983.

[11] V. P. Dragalin, A. G. Tartakovsky, and V. V. Veeravalli, “Multihypothesis sequential probability ratio tests-part I: asymptotic optimality,” *IEEE Transactions on Information Theory*, Vol. 45, no. 7, 1999, pp. 2448-2461.

[12] B. Eisenberg, “Multihypothesis Problems,” in *Handbook of Sequential Analysis*, B. K. Ghosh and P. K. Sens, Eds. New York: Marcel Dekker, 1991.

[13] K. S. Fu, *Sequential Pattern Recognition*. New York: Academic Press, 1968.

[14] J. D. Gibson and J. L. Melsa, *Introduction to Nonparametric Detection with Applications*, New York: IEEE Press, pp. 25, 1996.

[15] A. M. Hussian, “Multisensor Distributed Sequential Detection,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 3, pp. 698-708, 1994.

[16] I. Jouny and F. D. Garber, “M-ary Sequential Hypothesis Tests for Automatic Target Recognition,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 2, pp. 473-483, 1992.

[17] N. L. Johnson, S. Kotz, and B. Balakrishnan, *Continuous Univariate Distribution*s, Volume I, New York: John Wiley and Sons, Inc., pp. 426, 1994.

[18] M. W. Koch, G. B. Haschke, and K. T. Malone, “Classifying Acoustic Signatures using the Sequential Probability Ratio Test,” SAND2004-0705J, Sandia National Laboratories, Albuquerque NM, March 2004.

[19] D. Krazakos and J. Meadors, “Sequential Distributed Detection for Multisensor Fusion,” Proceedings of the 27th Conference on Decision and Control, Austin Texas, pp. 2507-2508, December 1988.
[20] C. H. Ku and W. H. Tsai, “Smooth Vision-Based Autonomous Land Vehicle Navigation in Indoor Environments by Person Following Using Sequential Pattern Recognition,” Journal of Robotic Systems, vol. 16, no. 5, pp. 249-262, 1999.
[21] C. C. Lee and J. B. Thomas, “ A Modified Sequential Detection Procedure,” *IEEE Transactions on Information Theory*, vol. 30, no. 1, pp. 16-23, 1984.
[22] K. Murphy, and B. Myors, *Statistical Power Analysis: A Simple and General Model for Traditional and Modern Hypothesis Tests*. New Jersey: Lawrence Erlbaum Associates, 1998.
[23] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proceedings of the IEEE, Vol. 77, No. 2., pp. 257-286, 1989.
[24] N. Ratnayake, M. Savic, J. Sorensen, “Use of Semi-Markov Models for Speaker-Independent Phoneme Recognition,” *IEEE International Conference on Acoustics, Speech and Signal Processing, San Francisco, CA.*, pp. 565-567, 1992.
[25] B. A. Rozanov, “Distribution of Truncated Sequential Tests for Rapidly Fading Radar Targets,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 4, no. 10, 1972, pp. 1667-1673.
[26] S. H. Steiner, P. L. Geyer, and G. O. Wesolowsky, “Grouped Data-Sequential Probability Ratio Tests and Cumulative Sum Control Charts,” *Technometrics*, vol. 38, no. 3, pp. 230-237, 1996.
[27] Wald, *Sequential Analysis*, New York: John Wiley & Sons Inc., 1947.
[28] N. D. Wallace, “Design of Truncated Sequential Tests for Rapidly Fading Radar Targets,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 4, no. 3, pp. 433-442, 1968.
[29] E. B. Wilson and M. M. Hilferty, The distribution of Chi-Square, *Proceedings of National Academy of Sciences*, vol. 17, 684-688, 1931.

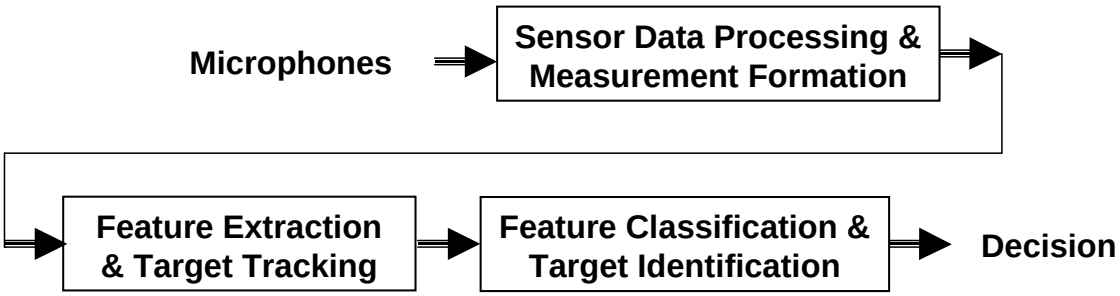

Figure 1. Block diagram of approach.

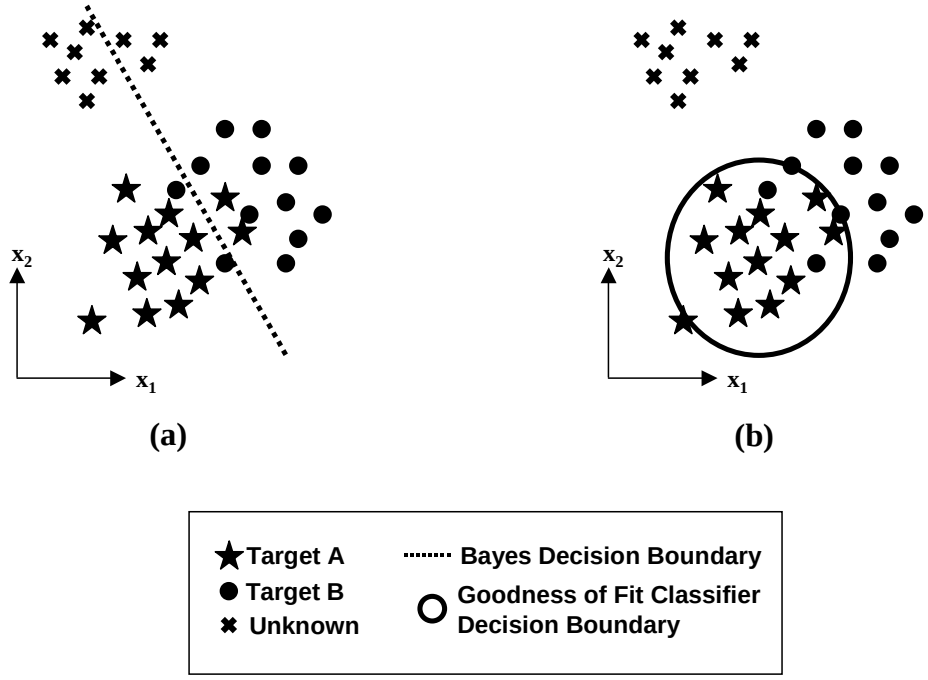

Figure 2. Comparison of Bayes and goodness of fit (GOF) classifiers. (a) Bayes classifier. (b) GOF classifier.

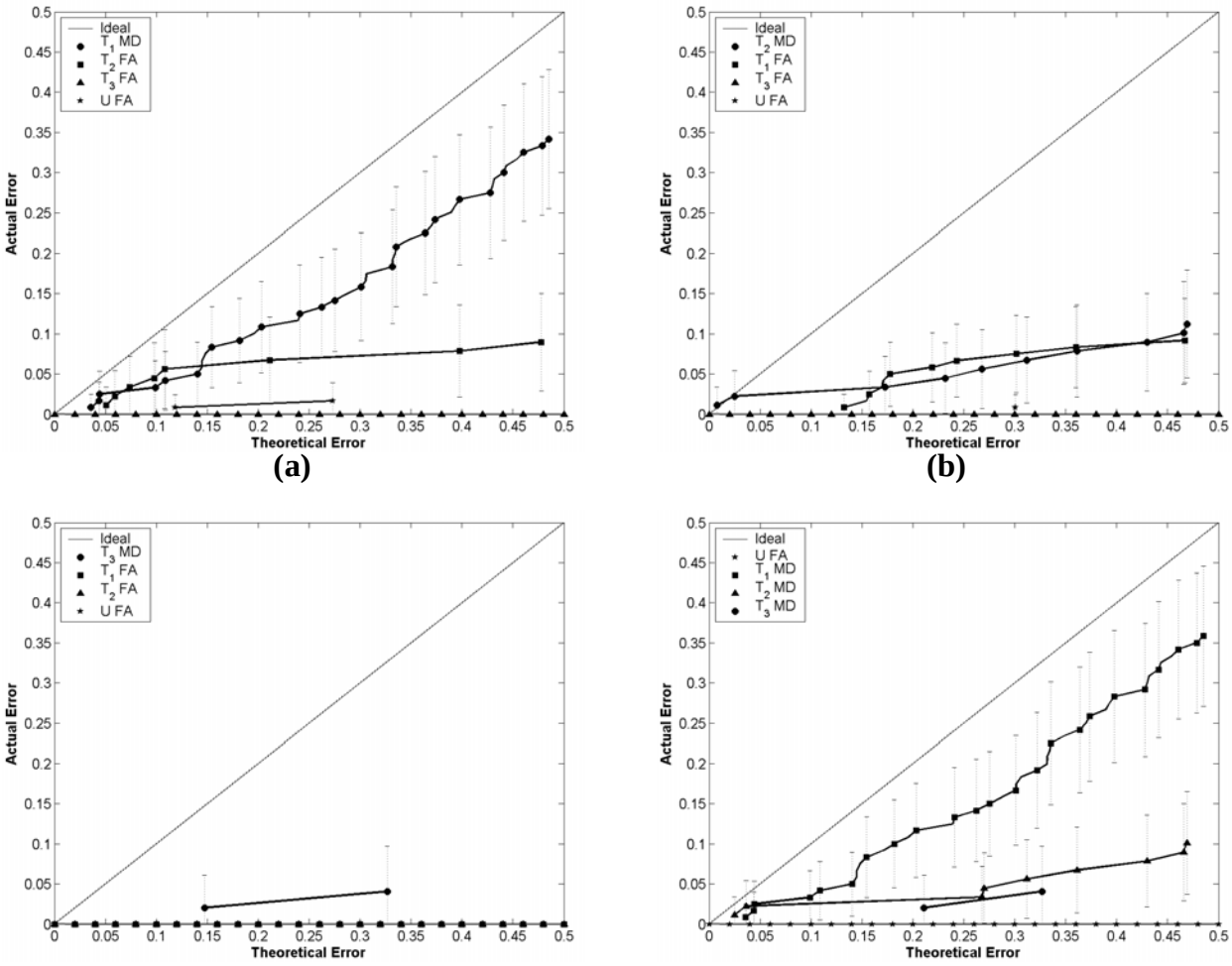

Figure 3. Verification plots for target identification using acoustic signatures. (a) GOF SPRT classifier for the T1 target. (b) GOF SPRT classifier for the T2 target. (c) GOF SPRT classifier for the T3 target. (d) Multiclass SPRT classifier.

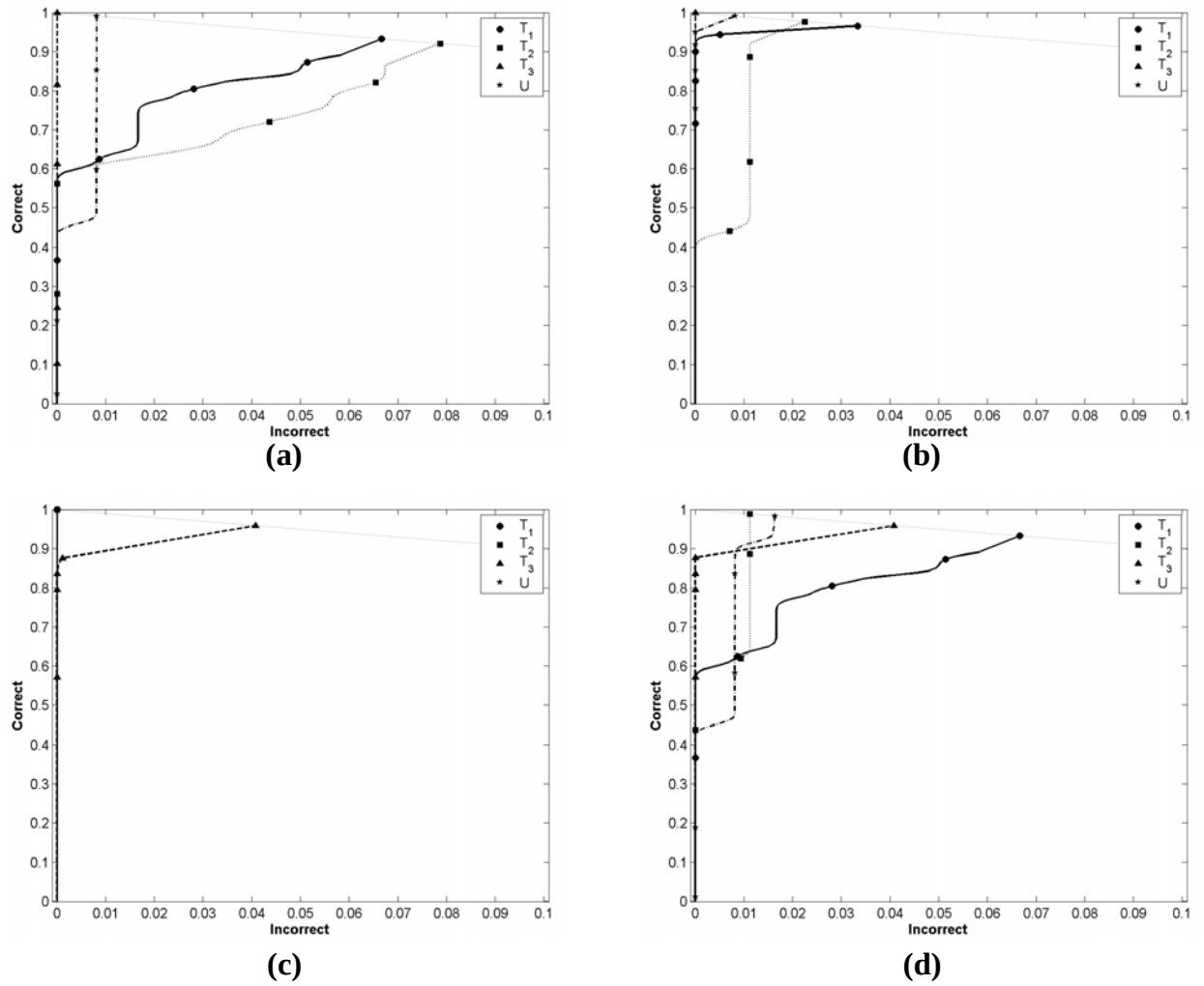

Figure 4. Operating characteristics for target identification of acoustic targets. (A) GOF T_1 classifier. (B) GOF T_2 classifier. (C) GOF T_3 classifier. (D) Multiclass classifier.

Distribution:

1	MS 0519	Greg B. Haschke, 02344
1	MS 0978	Larry G. Stotts, 05934
1	MS 0978	Kevin T. Malone, 05934
3	MS 1163	Mark W. Koch, 15433
1	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 9616