USC–SIPI REPORT #356

Binary Classification of Ground Vehicles Based on the Acoustic Data Using Fuzzy Logic Rule-Based Classifiers

by

Hongwei Wu and Jerry M. Mendel

October 2002

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 400
Los Angeles, CA 90089-2564 U.S.A.

Contents

List of Tables

List of Figures

Abstract

This report summarizes our research about the binary classification of ground vehicles based on the acoustic data, which was carried out from July 2001 to July 2002.

Because the acoustic data corresponding to each run are time-variant, we segmented each run into one-second data blocks, and used the data blocks, which we called prototypes, for classification. The magnitudes of the second through 12th harmonics of each prototype were used as features. We found, by analyzing the features within each run and across runs, that the run-means and run-standard-deviations of the features vary from run to run for all kinds of vehicles. We therefore used type-2 fuzzy sets to model the uncertainties contained in these features, and then constructed type-2 fuzzy logic rule-based classifiers (FL-RBC) for three binary classification problems: tracked vs. wheeled vehicle, heavy-tracked vs. light-tracked vehicle, and heavy-wheeled vs. light-wheeled vehicle. To evaluate the performance of the type-2 FL-RBCs in a fair way, we also constructed the Bayesian classifiers and type-1 FL-RBCs and compared their performance through many experiments. The parameters of the Bayesian classifiers were estimated using the training prototypes; whereas, the parameters of both the type-1 and type-2 FL-RBCs were optimized using a steepest descent algorithm that minimized an objective function which depended upon the training prototypes. All classifiers had two working modes—non-adaptive and adaptive. When the false alarm rate (FAR) of a classifier in its non-adaptive mode is less than 0.5 then this classifier has a better performance in its adaptive mode than in its non-adaptive mode after a certain time.

We carried out the leave-one-out and leave-M-out experiments to evaluate the performance of all classifiers. In the leave-one-out experiments, only one run was used for testing, and all the other runs were used for training. In the leave-M-out experiments, one run of each kind of vehicle was used for testing, and all the other runs were used for training. Our experiments showed that for each binary classification problem, both the type-1 and type-2 FL-RBCs had significantly better performance than the Bayesian classifier, whereas the type-1 and type-2 FL-RBCs had similar performance, although most of the time, the type-2 FL-RBC had slightly better performance than the type-1 FL-RBC.

Both the type-1 and type-2 FL-RBC designs were tested for blind runs of the normal terrain. The blind test results of the type-2 FL-RBC designs were scored by our sponsor at the Army Research Laboratory. The scores were very high, which demonstrates that our type-2 FL-RBC designs for the binary classification of ground vehicles based on their acoustic emissions are successful.

Chapter 1

Introduction

This report describes research about the classification of tracked and wheeled ground vehicles based only on their acoustic signatures. More specifically, we present results on the design and performance for three binary classification problems:

- tracked versus wheeled vehicles
- heavy-tracked versus light-tracked vehicles
- heavy-wheeled versus light-wheeled vehicles

The acoustic emission of a ground vehicle contains a wealth of information, which can be used for vehicle classification, e.g. in the battlefield. The model of the acoustic emission can be simplified as an addition of periodic components and random noise. The periodic components in the low frequency band are related to the periodic movements in the engine. For example, the engine firing rate of a cylinder-based engine is usually the most predominant peak in the spectrum; whereas, the acoustic energy of a turbine engine-based vehicle is distributed relatively evenly in the low frequency band because there is not an obvious fundamental frequency for it, such as the firing rate in the cylinder-based engine. Some tracked vehicles may also have periodic components in the high frequency band that are related to the sprocket and track system. The random noise is due to the propulsion process in the engine, and the interaction between the tires (or tracks) and roads [8].

Compared to other sensor systems (e.g., radar and optical) in the battlefield, acoustic sensors are less expensive and have fewer restrictions. The acoustic emission of a ground vehicle is easy to detect and hard to disguise; furthermore, a high signal-to-noise ratio (SNR) can be obtained for the acoustic data under some circumstances [8].

Although it is attractive to classify ground vehicles based on their acoustic emissions in the battlefield, it is still very challenging to accomplish this because the acoustic emissions of a ground vehicle are subject to variations of the environmental conditions (e.g., terrain and wind) and vehicle-traveling speed, their measurements are subject to the variation of the distance between the vehicle and the sensor system, and features obtained from these measurements may also be subject to the partitioning (windowing) of the measurements during the feature-extraction process. Hence, features that are extracted from the acoustic measurements of ground vehicles will be time-varying and will contain a lot of uncertainties. Any classifier that makes use of the acoustic measurements of ground vehicles for classification must therefore account for these time-variations and uncertainties.

Note that it is impossible to establish *precise* mathematical models to describe these variations and uncertainties. Type-2 fuzzy sets, whose membership functions (MF) have a so-called *footprint of uncertainty* (FOU), provide a novel way to model the unknown time-variations and uncertainties. Type-2 fuzzy logic systems (FLSs), which use type-2 fuzzy sets to model inputs, antecedents or consequents of rules, have already demonstrated excellent performance in non-stationary or uncertain environments [5]. This has motivated us to apply type-2 FLS theories to the designs of fuzzy-logic-rule-based classifiers (FL-RBC) for ground vehicle classification.

This report summarizes our study on binary classification of ground vehicles using FL-RBCs based on the acoustic data that were collected in the normal environmental conditions. Our FL-RBC results are base-lined against those obtained from a Bayesian classifier. Some preliminary results have been presented in [10] and [11], and these are incorporated into this report for completeness.

The rest of this report is organized as follows. In Chapter 2, we analyze the data, and specify the FOUs for features. In Chapter 3, we construct classifiers, and introduce two different working modes. In Chapter 4, we present experiment results. In Chapter 5, we describe the blind test and results. Finally in Chapter 6, we draw conclusions.

Chapter 2

Footprints of Uncertainties of the Features

The starting point for designing a type-2 FL-RBC is to specify the FOU of each feature. This is the fuzzy logic counterpart to the starting point for designing a Bayesian classifier, which is to specify an appropriate (joint) probability density function (pdf) for the features. In order to specify an appropriate FOU for each feature, we must understand the acoustic measurements of ground vehicles and the variations of the features that are extracted from them. The main purpose of this chapter is to establish such FOUs for all features that are used in our type-2 FL-RBC.

2.1 Data Collection

We used the "Acoustic-Seismic Classification/Identification Data Set" (ACIDS) for training and testing classifiers. ACIDS consists of more than 230 records of acoustic data that were collected by two sensor systems for nine kinds of ground vehicles in four environmental conditions. In our study, we have focused on three binary classification problems—tracked vs. wheeled vehicles, heavy-tracked vs. light-tracked vehicles, and heavy-wheeled vs. light-wheeled vehicles—for the *normal environmental condition* for which 115 records are available.

Each sensor system is a three-element equilateral triangular microphone array with a length of 38.1 cm between each microphone. When collecting the acoustic data, each sensor system was situated so that the triangle was pointing to, and one side of the triangle was parallel to a vehicle's traveling direction. All the three microphones of a sensor system were working so that one collected record has three *channels* of acoustic data. The acoustic data were low-pass filtered at 400 Hz to prevent spectral aliasing, high-pass filtered at 25 Hz to reduce wind noise, and then digitized by an 11-bit A/D converter at the sampling rate (F_s) of 1025.641 Hz.

In this work, we distinguish between a *run* and a *record.* A *run* corresponds to a ground vehicle traveling at a constant speed toward the sensor system, passing the closest point of approach (CPA), and then moving away from the sensor system. The duration of a run varies from 56 seconds to 420 seconds, for which the CPA to a sensor system varies from 25 m to 100 m, and, a vehicle's speed varies from 5 km/hr to 40 km/hr. Sometimes, there are two *records* (collected by the two sensor systems respectively) for the same run. In such a situation, we only used the record collected by sensor system 1. The numbers of runs and records for each kind of vehicle in the normal environmental condition are summarized in Table 2.1. An important fact from Table 2.1 is that there are only 89 runs in total for the normal environmental condition.

2.2 Prototype Generation

To fulfill the real-time requirements imposed by battlefield applications, a classifier needs to make decisions using short intervals of acoustic data; hence, we segmented the runs into one-second blocks, and treated one block (rather than an entire run) as one *prototype.*

Figure 2.1 shows the measurements of the first channel of one complete run. In the beginning and ending parts of the run, the magnitude of the acoustic data is low and flat, which corresponds to the moving vehicle being very far away from the sensor system. In the middle part of the record, the magnitude of the acoustic data rapidly increases and then decreases; this corresponds to the moving vehicle being closer to and then farther away from the sensor system. Clearly, the middle part of the run has higher SNR, and contains more information about the vehicle than its beginning and ending parts; hence, we should use this part to generate prototypes.

There are two ways to locate the middle part of a run. One is to first locate the CPA, and then truncate a fixed interval of data about it. The other is not based on CPA information; instead, we first estimate the background noise level from the beginning part of a run and then use an energy test to determine which part of the run has high SNR. We call the prototypes generated from the former method *CPA-based prototypes*, and those from the latter method *non-CPA-based prototypes.* We use the CPA-based prototypes for training and testing classifiers, and the non-CPA-based prototypes only for testing, because:

- CPA information is available during the design period of a classifier since each run is completely available to us. On the other hand, CPA information is not available during the implementation period of a classifier if this classifier must make decisions *before* a ground vehicle reaches its CPA.

- We can easily control the number of CPA-based prototypes for each run by adjusting the length of the interval about the CPA.
- Non-CPA-based prototypes are more vulnerable to noise than are the CPA-based prototypes.

2.2.1 CPA-Based Prototype Generation

When we design our classifiers we want to use the same number of CPA-based prototypes for both tracked and wheeled vehicles. However, in the ACIDS (see Table 2.1) the number of runs for the tracked vehicles is approximately twice that for the wheeled vehicles. Consequently, if we used both the same interval length and percentage of window overlap for both kinds of vehicles, then the total number of CPA-based prototypes would not be the same for the tracked and wheeled vehicles. We therefore used different percentages of window overlap so that for the same interval length about the CPA the number of CPA-based prototypes would be the same for both tracked and wheeled vehicles.

The method that we used to generate the CPA-based prototypes is:

1. The time, t_0, where the acoustic data has its maximum magnitude was considered as the time that a vehicle reaches its CPA.
2. Data blocks (prototypes), which have three channels, were generated from [t_0-25 sec, t_0+25 sec] by sliding a 1024-point (about 1 sec.) rectangular window (initially centered at t_0), to the right and to the left of t_0. For a *tracked* vehicle, the adjacent blocks had 50% overlap, and only the middle 80 blocks were used for classifier designs. For a *wheeled* vehicle, the adjacent blocks had 75% overlap, and only the middle 160 blocks were used for classifier designs.

In this way, the length of the interval to generate CPA-based prototypes is approximately the same for the tracked and wheeled vehicles. The length of the data[1] used for a tracked run is: $1024 + (80-1) \times (1-50\%) \times 1024 = 41,472$ data; and, the length of the interval used for a wheeled

[1]Let L be the window length, and $L = 1024$ for our acoustic data blocks. We used the first L data to generate the first block; and, for the Q $(0 \leq Q < 1)$ window overlap, we could then generate a new block whenever we had $(1-Q) \times L$ new data. So n adjacent data blocks corresponded to $L + (n-1) \times (1-Q) \times L$ data.

run is $1024 + (160-1) \times (1-75\%) \times 1024 = 41,728$ data. The total number of tracked CPA-based prototypes is (80 blocks/run)×(61 runs)=4,880 blocks, and the total number of wheeled CPA-based prototypes is (160 blocks/run)×(28 runs)=4,480 blocks.

Because only the CPA-based prototypes were used for training classifiers, it is very important that *using different percentages of window overlap for the tracked and wheeled vehicles does not affect our analysis about the FOUs of the features.* We have demonstrated (in Appendix A) that for the wheeled vehicles, the means and standard deviations of the features for the 75%-overlapped CPA-based prototypes are approximately the same as those for the 50%-overlapped CPA-based prototypes.

2.2.2 Non-CPA-Based Prototype Generation

The method used to generate the non-CPA-based prototypes is:

1. We calculated the energy of the first data block of a run, and used it as the background noise level of this run.

2. If the energy of a succeeding data block is greater than the background noise level for a certain threshold, then this data block was called a non-CPA-based prototype, and the classifier made a decision on it; otherwise, the classifier was not activated by that data block.

The threshold that is used during the energy test was determined as follows:

1. We calculated the difference (i.e., *energy difference*) between the maximum data-block energy and the minimum data-block energy in dB for all runs.

2. We calculated the average and standard deviation of the energy difference over all runs, and then set the threshold[2] as (average energy difference−one standard deviation)/3, which turned out to be approximately 8 dB.

The number of non-CPA-based prototypes varied from run-to-run. The average number is 93.6, the minimum number is 13 for a W-a run, and the maximum number is 273 for a T-a run.

[2]The threshold was set after examining the energy distribution of all runs; hence, it depended on the whole data set. The threshold cannot be too large, otherwise some runs may not have any non-CPA-based prototypes, nor can it be too small, otherwise some runs may generate non-CPA-based prototypes from low SNR data.

2.3 Feature Extraction

In previous works [9], three feature spaces have been examined—*spectral lines*, *harmonic line association* (HLA), and *principal components*. Spectral lines and HLA use the elements in the magnitude spectrum of a data block. A spectral line vector consists of the magnitudes of all the frequency components below 200 Hz, whereas an HLA vector is a subset of a spectral line vector, and consists of the magnitudes of the 2nd through 12th harmonic frequency components[3]. Principal components are obtained through a linear transformation of a data block, where the transformation function is determined by principal component analysis [2].

The spectral line feature vector seems to preserve the most information contained in an entire data block, but its high dimensionality is not suitable for FL-RBCs. The principal component feature vector, although promising, does not directly correspond to any physical quantities, and is more appropriate for representing a data block than for discriminating a data block [2]. In our study, we only used the HLA feature vector (i.e., the magnitudes of the 2nd through 12th harmonic frequency components) for vehicle classification.

Determining the fundamental frequency, f_0, is crucial to extracting the features from a data block. Only after f_0 is estimated can the harmonic frequency components be located and their magnitudes be taken as features. The maximum-likelihood method was used in [4] to estimate f_0. This method is based on the assumptions that the acoustic emission of a ground vehicle can be well modeled by a coupled harmonic signal model, and that f_0 for each run is initially known to within a certain accuracy (e.g., roughly 0.5 Hz). The HLA algorithm was developed in [7] to estimate f_0. Compared to the maximum-likelihood method, the HLA method does not rely very strongly on a mathematical model for the acoustic emission of a ground vehicle, and it requires less a priori knowledge about f_0.

The HLA algorithm, which we have used in our study for f_0 estimation and feature extraction, is briefly reviewed as follows (note that there are three channels):

1. Take one data block, and set the channel index $i = 1$.

2. The i-th channel of the data block is normalized so that the total energy of this channel is unity.

[3] Exactly why this range of harmonics is used is unclear. Perhaps more of them, or fewer of them, are needed by a FL-RBC to give acceptable performance. In this study, we did not address this interesting issue.

3. The discrete Fourier transformation (DFT) is performed on the i-th channel, and only the magnitude spectrum of the DFT is used for feature extraction.

4. The real f_0 is assumed to be in the range (9 Hz, 18 Hz]. From the magnitude spectrum of the i-th channel, we determined the frequency component associated with the maximum magnitude, $f^i_{\max}$. This $f^i_{\max}$ is considered to be the k-th harmonic line so that $f^i_{\max}/k$ can be in the range (9 Hz, 18 Hz]. Clearly, k may take multiple integer values. For example, if $f^i_{\max} = 100$ Hz, then k can be $\{6, \ldots, 10\}$. For each possible value of k, we have a potential fundamental frequency $f^i_0(k) = f^i_{\max}/k$.

5. For each $f^i_0(k)$, the magnitudes of its 2nd through 12th harmonic lines are summed. We compare the summed-magnitudes for all potential fundamental frequencies, and then determine one particular potential fundamental frequency $f^i_0(k^i_{\max}) \equiv f^i_0$ that is associated with the maximum summed-magnitude. This f^i_0 is considered to be the fundamental frequency for the i-th channel, and its summed-magnitude is saved.

6. If i is not the last channel, set $i = i + 1$ and go to Step 2; otherwise go to Step 7.

7. Among f^i_0 and their associated summed-magnitudes for all channels, the f^i_0 associated with the maximum summed-magnitude is determined as the estimated f_0 for the specific data block.

8. After f_0 is determined, the magnitudes of the 2nd through 12th harmonic lines are averaged over all three channels to get the feature vector for one data block.

Note that the initial range, (9 Hz, 18 Hz] was adapted from [9] where the range [8 Hz, 20 Hz] was used. We chose (9 Hz, 18 Hz] to avoid one frequency in the range being an integer multiple of another frequency in the range. Suppose, for example, that 9 Hz is the actual fundamental frequency of a given data block, and from the magnitude spectrum of the i-th channel of this data block we have located $f^i_{\max}$ to be 90 Hz. If the range [8 Hz, 20 Hz] was used, then $f^i_{\max}$ could correspond to many potential fundamental frequencies, including 9 Hz and 18 Hz. Because 9 Hz is the actual fundamental frequency, the sum of the magnitudes, $\sum_{k=2}^{12}\text{Magnitude}(9k)$, at the second through 12th hamonics of 9 Hz is expected to be larger than the sum of the magnitudes at the second through 12th hamonics of any other potential fundamental frequency. However, because the harmonics of 18 Hz are simultaneously the harmonics of 9 Hz, it is very possible that $\sum_{k=2}^{12}\text{Magnitude}(9k) \leq \sum_{k=2}^{12}\text{Magnitude}(18k)$. This means that 18 Hz might be incorrectly considered to be the fundamental frequency. Such errors can be avoided by using the range (9 Hz, 18 Hz].

2.4 Footprints of Uncertainties of the Features

The *footprint of uncertainty* is associated with a type-2 fuzzy set. A type-2 fuzzy set, $\tilde{A}$, is characterized by a three dimensional MF, $\mu_{\tilde{A}}(x,u)$, as:

$$\tilde{A} = \left\{ \left((x,u), \mu_{\tilde{A}}(x,u)\right) | \forall x \in X, \forall u \in J_x \subseteq [0,1] \right\} \tag{2.1}$$

where x is a point in the universe of discourse X, u is a primary membership grade of x, $\mu_{\tilde{A}}(x,u)$ is the secondary membership grade of (x,u), and J_x is the union of all the primary membership grades of x. The union of J_x over all $x \in X$ corresponds to a bounded region in the x-u plane, and is called the *footprint of uncertainty (FOU)* of $\tilde{A}$. The FOU provides a convenient way to describe the uncertainties inherent in a type-2 fuzzy set. In practical applications of type-2 fuzzy sets, it is very important to specify an appropriate FOU for it before specifying its specific MF [5].

Because only the CPA-based prototypes are used to train classifiers, our analysis about the FOUs of the features focuses on the CPA-based prototypes. After feature extraction for each CPA-based prototype, we calculated the run-means and run-standard-deviations of the features for each run. Tables 2.2 and 2.3 summarize these statistics for one representative tracked and wheeled run. We then plotted the distributions of the features, and calculated statistics of the features *across runs* for each kind of vehicle. Figure 2.2 shows the distributions of the features for a representative tracked and wheeled vehicle, and Tables 2.4 and 2.5 summarize the statistics across runs for these two representative vehicles. Observe from Fig. 2.2 and Tables 2.4 and 2.5, that for each feature:

- The standard deviation of the run-means is not negligible when compared to the mean of the run-means.
- The standard deviation of the run-standard-deviations is also not negligible when compared to the mean of the run-standard-deviations.
- The standard deviation of the run-standard-deviations is of the same order of magnitude as the standard deviation of the run-means.

These results establish that when we construate a FL-RBC, the MFs must be properly chosen so as to account for the *simultaneous variations in the run-means and run-standard-deviations* of the features. This led us to choose the MF for each antecedent to be a Gaussian primary MF with an uncertain mean and an uncertain standard deviation, and the MF for each input to be a Gaussian primary MF whose mean is located at the measured feature and whose standard deviation is uncertain. Figure 2.3 shows the FOU of such type-2 fuzzy sets.

Table 2.1: Numbers of runs and records for each kind of vehicle in the normal environmental condition.

Tracked Vehicle			**Wheeled Vehicle**		
	No. of Runs	No. of Records		No. of Runs	No. of Records
T-a (Vehicle 1)	15	22	W-a (Vehicle 3)	8	9
T-b (Vehicle 2)	8	12	W-b (Vehicle 5)	8	12
T-c (Vehicle 4)	15	15	W-c (Vehicle 6)	8	12
T-d (Vehicle 8)	15	17	W-d (Vehicle 7)	4	4
T-e (Vehicle 9)	8	12			
Total	61	78	Total	28	37
Total No. of runs			89		

Table 2.2: Means and standard deviations of the features (x_i) for a representative tracked (T-c) run.

	x_1	x_2	x_3	x_4	x_5	x_6
mean	1.1268	1.1806	3.0424	1.3921	16.0297	1.6044
standard deviation	0.3774	0.4922	1.7033	0.4068	3.1944	0.5201

	x_7	x_8	x_9	x_{10}	x_{11}
mean	1.5764	2.4929	0.8605	0.7485	1.0897
standard deviation	1.0043	1.0172	0.2743	0.4269	0.4567

Table 2.3: Means and standard deviations of the features (x_i) for a representative wheeled (W-b) run.

	x_1	x_2	x_3	x_4	x_5	x_6
mean	0.3191	3.4738	1.5670	2.7815	16.7137	0.9684
standard deviation	0.2247	1.4316	0.7136	1.3797	2.6064	0.7316

	x_7	x_8	x_9	x_{10}	x_{11}
mean	0.5049	1.4842	0.5240	0.8240	2.2348
standard deviation	1.3374	1.0310	0.3468	0.6037	2.0614

Table 2.4: Statistics of the features (x_i) for a representative tracked (T-c) vehicle (SD represents standard deviation).

Statistics	x_1	x_2	x_3	x_4	x_5	x_6
mean of run-means	0.4063	1.0699	3.8050	1.2316	11.0492	1.2041
mean of run-SDs	0.2054	0.5817	2.1879	1.4539	3.7128	0.5020
SD for run-means	0.2714	0.7422	3.2831	0.5084	5.1150	0.2470
SD for run-SDs	0.1411	0.6966	2.1495	1.4467	1.3279	0.3111
Statistics	x_7	x_8	x_9	x_{10}	x_{11}	
mean of run-means	2.6432	3.2985	1.1897	1.0100	2.8322	
mean of run-SDs	1.4490	1.6748	0.7514	0.4725	1.8120	
SD of run-means	1.5600	1.8537	0.2787	0.2551	1.1158	
SD of run-SDs	1.1254	1.7061	0.4456	0.2763	1.3357	

Table 2.5: Statistics of the features (x_i) for a representative wheeled (W-b) vehicle (SD represents standard deviation).

Statistics	x_1	x_2	x_3	x_4	x_5	x_6
mean of run-means	0.3909	4.3311	1.0387	2.7465	15.0241	1.4667
mean of run-SDs	0.2587	2.0478	0.5006	1.4653	3.6137	1.0226
SD of run-means	0.0868	1.5129	0.3697	1.2053	3.5494	0.4948
SD of run-SDs	0.1050	0.7662	0.1382	0.6048	1.5683	0.3927
Statistics	x_7	x_8	x_9	x_{10}	x_{11}	
mean of run-means	0.5399	2.8427	0.5376	0.8914	1.2883	
mean of run-SDs	0.5455	2.1736	0.6445	0.7047	1.2246	
SD for run-means	0.1495	1.3634	0.1115	0.5275	0.7647	
SD for run-SDs	0.5144	0.8806	0.5531	0.4198	0.8272	

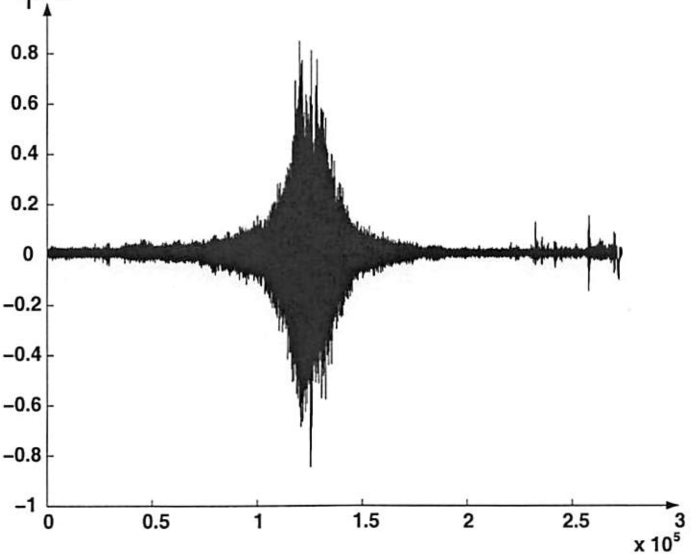
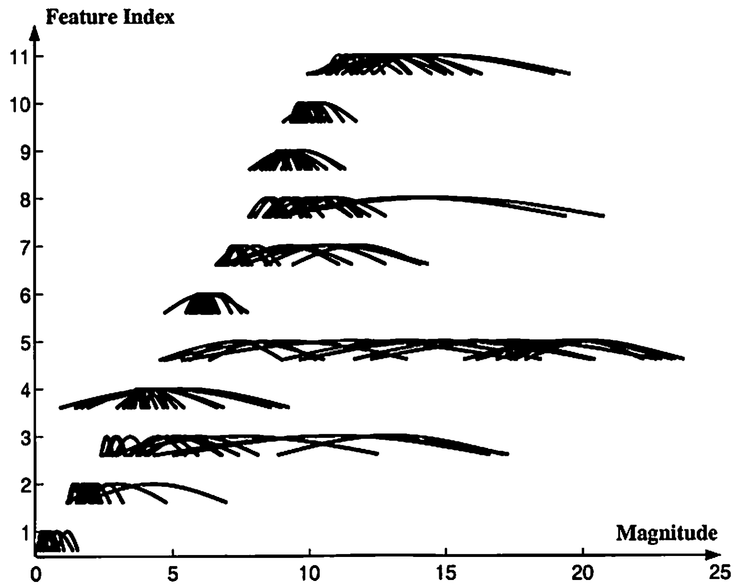
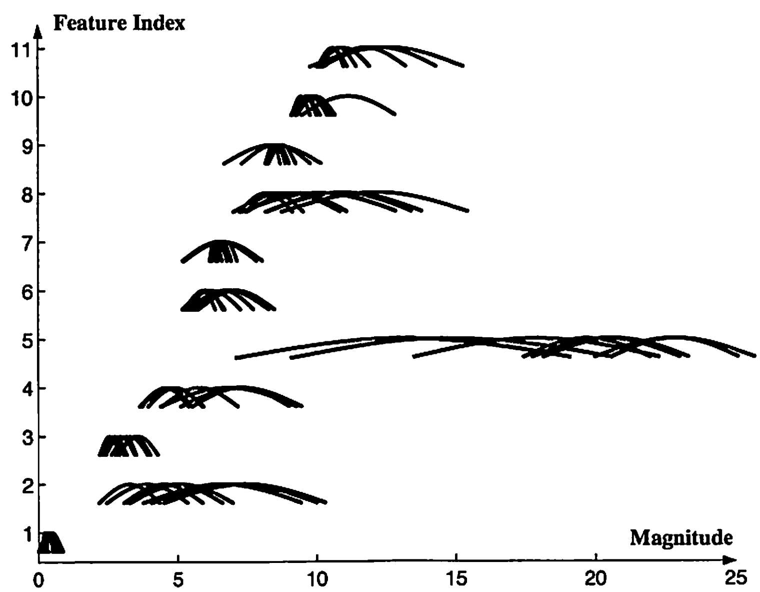


Figure 2.1: One channel of acoustic data from one run in the normal environmental condition. The horizontal axis is the sample index.



(a)



(b)

Figure 2.2: Feature distributions for a representative (a) tracked (T-c) and (b) wheeled (W-b) vehicle. Each curve represents a feature distribution within one run, which is assumed to be normal, and is plotted based on the run-mean and run-standard-deviation.

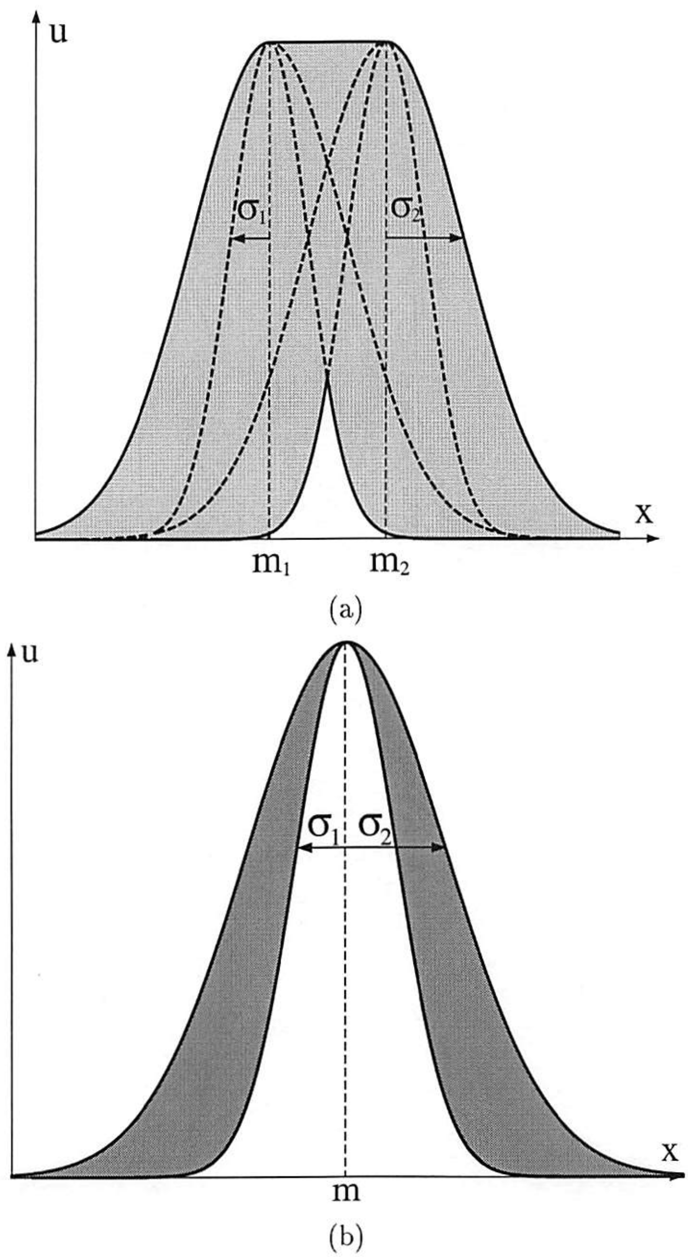


Figure 2.3: FOU of type-2 fuzzy sets: (a) Gaussian primary MF with an uncertain mean and an uncertain standard deviation; and (b) Gaussian primary MF with an uncertain standard deviation.

Chapter 3

Classifier Designs

To evaluate the type-2 FL-RBC in a fair way, we compared its performance to two other classifiers, namely a Bayesian classifier and a type-1 FL-RBC. In this chapter we provide the details about all these classifiers.

3.1 Classifier Structures

3.1.1 Bayesian Classifier

For the Bayesian classifier, the features are considered to be random vectors that are assumed to comply to some probability distribution models. Our Bayesian classifiers consisted of nine probability models for the tracked vs. wheeled vehicle classification problem, five probability models for the heavy-tracked vs. light-tracked vehicle classification problem, and four probability models for the heavy-wheeled vs. light-wheeled vehicle classification problem. Each probability model corresponds to one kind of vehicle, and is assumed to be described by a Gaussian pdf:

$$p(\mathbf{x}|V_l) = \frac{1}{\sqrt{(2\pi)^{11}\det(\Sigma_l)}}\exp\left[-(\mathbf{x}-\mathbf{m}_l)^t\Sigma_l^{-1}(\mathbf{x}-\mathbf{m}_l)\right], \quad l=1,\ldots,M \tag{3.1}$$

where V_l represents the l-th kind of vehicle, $\mathbf{x}$ is an 11×1 feature vector, $\mathbf{m}_l$ and Σ_l are the mean and covariance matrix for V_l, and M is the number of probability models.

Given an unknown feature vector $\mathbf{x}'$, the Bayesian classifier [2] first computes the log-likelihood for each model, as:

$$L(\mathbf{x}'|V_l) = \log p(\mathbf{x}'|V_l) = -\frac{11}{2}\log(2\pi) - \frac{1}{2}\log\left[\det(\Sigma_l)\right] - (\mathbf{x}'-\mathbf{m}_l)^t\Sigma_l^{-1}(\mathbf{x}'-\mathbf{m}_l), \quad l=1,\ldots,M \tag{3.2}$$

It then compares $L(\mathbf{x}'|V_l)$ for all M models to determine $V_{l^*} \equiv V_{\max}$ that is associated with the maximum log-likelihood. Finally the unknown feature vector $\mathbf{x}'$ is classified as the same kind of vehicle as $V_{\max}$, i.e., for the tracked vs. wheeled vehicle classification problem, $\mathbf{x}'$ is classified as a tracked (or wheeled) vehicle if and only if $V_{\max}$ is for a tracked (or wheeled) vehicle; for the heavy-tracked vs. light-tracked vehicle classification problem, $\mathbf{x}'$ is classified as a heavy-tracked (or light-tracked) vehicle if and only if $V_{\max}$ is for a heavy-tracked (or light-tracked) vehicle; and, for the heavy-wheeled and light-wheeled vehilce classification problem, $\mathbf{x}'$ is classified as a heavy-wheeled (or light wheeled) if and only if $V_{\max}$ is for a heavy-wheeled (or light-wheeled) vehicle.

3.1.2 Type-1 FL-RBC

A type-1 FL-RBC is a type-1 FLS that is used for classification. Its rule base consisted of nine rules for the tracked vs. wheeled vehicle classification problem, five rules for the heavy-tracked vs. light-tracked vehicle classification problem, and four rules for the heavy-wheeled vs. light-wheeled vehicle classification problem. Each rule corresponds to one kind of vehicle, and has the following form:

R_1^l: If x_1 is F_1^l and ... and x_{11} is F_{11}^l, then y is g^l

where R_1^l denotes the type-1 FL rule for the l-th kind of vehicle, $\mathbf{x} = [x_1, \ldots, x_{11}]^t$ are the features, their corresponding antecedents F_k^l $(k = 1, \ldots, 11)$ are type-1 fuzzy sets with MFs $\mu_{F_k^l}(x_k)$, and the consequent g^l is a crisp number that in the tracked vs. wheeled vehicle classification problem is positive for tracked vehicles and negative for wheeled vehicles, in the heavy-tracked vs. light-tracked vehicle classification problem is positive for heavy-tracked vehicles and negative for light-tracked vehicles, and in the heavy-wheeled vs. light-wheeled vehicle classification problem is positive for heavy-wheeled vehicles and negative for light-wheeled vehicles.

Given an unknown feature vector $\mathbf{x}' = [x_1', \ldots, x_{11}']^t$, the type-1 FL-RBC first encodes each x_k' $(k = 1, \ldots, 11)$ as a type-1 fuzzy set X_k with MF $\mu_{X_k}(x_k)$. It then computes the firing degree, $f^l(\mathbf{x}')$, for each rule [5], as:

$$f^l(\mathbf{x}') = \left[\sup_{x_1} \mu_{X_1}(x_1)\mu_{F_1^l}(x_1)\right] \times \ldots \times \left[\sup_{x_{11}} \mu_{X_{11}}(x_{11})\mu_{F_{11}^l}(x_{11})\right] \tag{3.3}$$

and combines the rules through defuzzification to obtain the output $y(\mathbf{x}')$ as:

$$y(\mathbf{x}') = \frac{\sum_{l=1}^M f^l(\mathbf{x}')g^l}{\sum_{l=1}^M f^l(\mathbf{x}')} \tag{3.4}$$

where M represents the number of rules. The decision on the unknown feature vector $\mathbf{x}'$ depends on the sign[4] of $y(\mathbf{x}')$, i.e. for the tracked vs. wheeled vehicle classification problem, $\mathbf{x}'$ is classified as a tracked (or wheeled) vehicle if and only if $y(\mathbf{x}')$ is positive (or negative); for the heavy-tracked vs. light-tracked vehicle classification problem, $\mathbf{x}'$ is classified as a heavy-tracked (or light-tracked) vehicle if and only if $y(\mathbf{x}')$ is positive (or negative); and, in the heavy-wheeled and light-wheeled vehicle classification problem, $\mathbf{x}'$ is classified as a heavy-wheeled (or light-wheeled) vehicle if and only if $y(\mathbf{x}')$ is positive (or negative).

We chose each antecedent MF, $\mu_{F_k^l}(x_k)$, as a Gaussian with mean m_k^l and standard deviation σ_k^l, and each measurement MF, $\mu_{X_k}(x_k)$, as a Gaussian centered at the measured feature x_k' with a standard deviation σ_k. Hence, our type-1 FL-RBC is characterized by the following design parameters: m_k^l (11 antecedents/rule$\times M$ rules $= 11M$ parameters), σ_k^l (11 antecedents/rule$\times M$ rules $= 11M$ parameters), g^l (1 consequent/rule$\times M$ rules $= M$ parameters), and σ_k (11 measurement parameters), for a total of $(23M + 11)$ parameters.

3.1.3 Type-2 FL-RBC

The type-2 FL-RBC we chose to use is an interval type-2 FLS that is used for classification. This is a type-2 FLS whose type-2 fuzzy sets are interval sets. As in the case of a type-1 FL-RBC, its rule base consisted of nine rules for the tracked vs. wheeled vehicle classification problem, five rules for the heavy-tracked vs. light-tracked vehicle classification problem, and four rules for the heavy-wheeled vs. light-wheeled vehicle classification problem. Each rule corresponds to one kind of vehicle, and has the following form:

R_2^l: If x_1 is $\tilde{F}_1^l$ and ... and $\tilde{F}_{11}^l$, then y is g^l

where R_2^l denotes the interval type-2 FL rule for the l-th kind of vehicle, $\mathbf{x} = [x_1, \dots, x_{11}]^t$ are the features, their corresponding antecedents $\tilde{F}_k^l$ $(k = 1, \dots, 11)$ are *interval type-2 fuzzy sets* with lower and upper MFs $\underline{\mu}_{\tilde{F}_k^l}(x_k)$ and $\overline{\mu}_{\tilde{F}_k^l}(x_k)$ respectively, and the consequent g^l is a crisp number that in the tracked vs. wheeled vehicle classification problem is positive for tracked vehicles and negative for wheeled vehicles, in the heavy-tracked vs. light-tracked vehicle classification problem is positive for heavy-tracked vehicles and negative for light-tracked vehicles, and in the heavy-wheeled vs. light-wheeled vehicle classification problem is positive for heavy-wheeled vehicles and negative

[4]We could also use the unnormalized output of the FLS for the decision, because $\sum_{l=1}^{M} f^l(\mathbf{x}')$ does not affect the sign of $y(\mathbf{x}')$.

for light-wheeled vehicles. Note that the structure of a rule is the same for type-1 and type-2 FL-RBCs. It is the way in which the antecedents and measurements are *modeled* that is different.

Given an unknown feature vector $\mathbf{x}' = [x'_1, \ldots, x'_{11}]^t$, the type-2 FL-RBC first encodes each x'_k ($k = 1, \ldots, 11$) as an interval type-2 fuzzy set $\tilde{X}_k$ with lower and upper MFs $\underline{\mu}_{\tilde{X}_k}(x_k)$ and $\overline{\mu}_{\tilde{X}_k}(x_k)$, respectively. It then computes the lower and upper firing degrees, $\underline{f}^l(\mathbf{x}')$ and $\overline{f}^l(\mathbf{x}')$, for each rule [5], as:

$$\underline{f}^l(\mathbf{x}') = \left[\sup_{x_1} \underline{\mu}_{\tilde{X}_1}(x_1)\underline{\mu}_{\tilde{F}_1^l}(x_1)\right] \times \ldots \times \left[\sup_{x_{11}} \underline{\mu}_{\tilde{X}_{11}}(x_{11})\underline{\mu}_{\tilde{F}_{11}^l}(x_{11})\right] \tag{3.5}$$

$$\overline{f}^l(\mathbf{x}') = \left[\sup_{x_1} \overline{\mu}_{\tilde{X}_1}(x_1)\overline{\mu}_{\tilde{F}_1^l}(x_1)\right] \times \ldots \times \left[\sup_{x_{11}} \overline{\mu}_{\tilde{X}_{11}}(x_{11})\overline{\mu}_{\tilde{F}_{11}^l}(x_{11})\right] \tag{3.6}$$

and combines the rules through type-reduction [5] to obtain a *type-reduced set* $[y_l(\mathbf{x}'), y_r(\mathbf{x}')]$, where

$$y_l(\mathbf{x}') = \frac{\sum_{l=1}^{L} \overline{f}^l(\mathbf{x}')g^l + \sum_{l=L+1}^{M} \underline{f}^l(\mathbf{x}')g^l}{\sum_{l=1}^{L} \overline{f}^l(\mathbf{x}') + \sum_{l=L+1}^{M} \underline{f}^l(\mathbf{x}')} \tag{3.7}$$

$$y_r(\mathbf{x}') = \frac{\sum_{l=1}^{R} \underline{f}^l(\mathbf{x}')g^l + \sum_{l=R+1}^{M} \overline{f}^l(\mathbf{x}')g^l}{\sum_{l=1}^{R} \underline{f}^l(\mathbf{x}') + \sum_{l=R+1}^{M} \overline{f}^l(\mathbf{x}')} \tag{3.8}$$

in which M represents the number of rules, R_2^l ($l = 1, \ldots, M$) are re-ordered such that $g^1 \leq \ldots \leq g^M$, and, L and R are each determined using the Karnik-Mendel iterative procedures[5] [5]. Finally, the type-2 FL-RBC defuzzifies the type-reduced set to get an output $[y_l(\mathbf{x}') + y_r(\mathbf{x}')]/2$. The decision of the type-2 FL-RBC about the unknown feature vector $\mathbf{x}'$ depends on the sign of $[y_l(\mathbf{x}') + y_r(\mathbf{x}')]/2$, i.e. for the tracked vs. wheeled vehicle classification problem, $\mathbf{x}'$ is classified as a tracked (or wheeled) vehicle if and only if $[y_l(\mathbf{x}') + y_r(\mathbf{x}')]/2$ is positive (or negative); for the heavy-tracked vs. light-tracked vehicle classification problem, $\mathbf{x}'$ is classified as a heavy-tracked (or light-tracked) vehicle if and only if $[y_l(\mathbf{x}') + y_r(\mathbf{x}')]/2$ is positive (or negative); and for the heavy-wheeled vs. light-wheeled vehicle classification problem, $\mathbf{x}'$ is classified as a heavy-tracked (or light-tracked) vehicle if and only if $[y_l(\mathbf{x}') + y_r(\mathbf{x}')]/2$ is positive (or negative).

As we stated in Section 2.4, the MFs for the type-2 FL-RBC must be properly chosen so as to capture the variations in the run-means and run-standard-deviations of the features. Consequently, we chose the MF for each $\tilde{F}_k^l$ to be a Gaussian primary MF with an uncertain mean, $m_k^l \in \left[m_{1,k}^l, m_{2,k}^l\right]$, and an uncertain standard deviation, $\sigma_k^l \in \left[\sigma_{1,k}^l, \sigma_{2,k}^l\right]$, and the MF for each $\tilde{X}_k$ to be a Gaussian primary MF where the mean is located at the measured feature x'_k, but the standard deviation is uncertain, i.e. $\sigma_k \in [\sigma_{1,k}, \sigma_{2,k}]$. Hence, our type-2 FL-RBC is characterized by the

[5]These procedures require at most M iterations.

following design parameters: $m_{1,k}^l$ and $m_{2,k}^l$ (2×11 antecedents/rule×M rules = $22M$ parameters), $\sigma_{1,k}^l$ and $\sigma_{2,k}^l$ (2×11 antecedents/rule×M rules = $22M$ parameters), g^l (1 consequent/rule×M rules = M parameters), and $\sigma_{1,k}$ and $\sigma_{2,k}$ (2×11 measurement parameters = 22 parameters), for a total of ($45M$+22) parameters.

3.2 Classifier Initializations and Optimizations

3.2.1 Bayesian Classifier

The parameters $\mathbf{m}_l$ and Σ_l, $l = 1, \ldots, M$ (M is the number of vehicles in each classification problem, i.e. $M = 9$ for tracked vs. wheeled vehicle classification, $M = 5$ for heavy-tracked vs. light-tracked vehicle classification, and $M = 4$ for heavy-wheeled vs. light-wheeled vehicle classification) for each probability model of the Bayesian classifier are estimated from the training prototypes as [2]:

$$\mathbf{m}_l = \frac{1}{N_l} \sum_{i \in V_l} \mathbf{x}_i \tag{3.9}$$

$$\Sigma_l = \frac{1}{N_l - 1} \sum_{i \in V_l} (\mathbf{x}_i - \mathbf{m}_l)(\mathbf{x}_i - \mathbf{m}_l)^t \tag{3.10}$$

where V_l denotes the l-th kind of vehicle, N_l represents the number of training prototypes for V_l (e.g., in the leave-one-out experiment, if all runs of the tracked vehicle T-a are used for training, then the N_l associated with T-a is 80 prototypes/run×15 runs = 1200 prototypes; or if all runs of the wheeled vehicle W-a are used for training, then the N_l associated with W-a is 160 prototypes/run×8 runs = 1280 prototypes), and $\mathbf{x}_i$ is the feature vector of a training prototype.

Note that each $\mathbf{m}_l$ is an 11×1 vector, each Σ_l is an 11×11 matrix, and these quantities vary because the training prototypes are different from design to design. So, we do not show their values. Those who are interested can repeat our experiments and obtain these values by following our methodology.

Once the parameters of the Bayesian classifier are estimated, they are fixed and are not further optimized using the training prototypes.

3.2.2 Type-1 FL-RBC

The MF parameters of the type-1 FL-RBC are initialized as ($l = 1, \ldots, M$):

$$m_k^l(0) = \frac{1}{N_l}\sum_{i\in V_l} \mathrm{x}_{k,i} \tag{3.11}$$

$$\left[\sigma_k^l(0)\right]^2 = \frac{1}{N_l - 1}\sum_{i\in V_l}\left[\mathrm{x}_{k,i} - m_k^l(0)\right]^2 \tag{3.12}$$

$$\sigma_k(0) = \frac{1}{M}\sum_{l=1}^{M}\sigma_k^l(0) \tag{3.13}$$

where V_l denotes the l-th kind of vehicle, N_l is the number of training prototypes for V_l, and $\mathrm{x}_{k,i}$ is the k-th element of the feature vector $\mathbf{x}_i$. For the tracked vs. wheeled vehicle classification problem, the consequent parameters are initialized as $g^l(0) = 1$ if V_l is a tracked vehicle, or as $g^l(0) = -1$ if V_l is a wheeled vehicle; for the heavy-tracked vs. light-tracked vehicle classification problem, the consequent parameters are initialized as $g^l(0) = 1$ if V_l is a heavy-tracked vehicle, or as $g^l(0) = -1$ if V_l is a light-tracked vehicle; and, for the heavy-wheeled vs. light-wheeled classification problem, the consequent parameters are initialized as $g^l(0) = 1$ if V_l is a heavy-wheeled vehicle, or as $g^l(0) = -1$ if V_l is a light-wheeled vehicle.

The classifier's $(23M + 11)$ design parameters are optimized using a steepest-descent algorithm to minimize the following objective function:

$$J_1(\theta_1) = \sum_{i=1}^{N}\left[y_d(i) - y\left(\mathrm{x}_i|\theta_1\right)\right]^2 \tag{3.14}$$

where θ_1 represents the parameters of the type-1 FL-RBC to be tuned, N is the total number of training prototypes[6], $y_d(i)$ is the desired classification result for the i-th training prototype (in the tracked vs. wheeled vehicle classification problem, $y_d(i)$ is $+1$ for a tracked prototype and -1 for a wheeled prototype; in the heavy-tracked vs. light-tracked vehicle classification problem, it is $+1$ for a heavy-tracked prototype and -1 for a light-tracked prototype; and, in the heavy-wheeled vs. light-wheeled vehicle classifiction problem, it is $+1$ for a heavy-wheeled prototype and -1 for a light-wheeled prototype), and $y\left(\mathrm{x}_i|\theta_1\right)$ is the output given by the type-1 FL-RBC. Let $\theta_1(j)$ be the parameters in the j-th epoch (one epoch corresponds to a single presentation of all training protototypes [2]), then $\theta_1(j+1)$ is determined as:

$$\theta_1(j+1) = \theta_1(j) - [\alpha p(j)]\left.\frac{\partial J_1(\theta_1)}{\partial \theta_1}\right|_{\theta_1(j)} \tag{3.15}$$

[6] $N = N_1 + \ldots + N_M$, where N_l is the number of training prototypes for V_l. If V_l is a tracked vehicle, then $N_l = 80$ prototypes/run$\times$ number of training runs of V_l; if V_l is a wheeled vehicle, then $N_l = 160$ prototypes/run$\times$ number of training runs of V_l. See Table 2.1 for the number of training runs for each kind of vehicle.

where

$$p(j) = \frac{number \quad of \quad incorrectly \quad classified \quad testing \quad prototypes}{total \quad number \quad of \quad testing \quad prototypes} \tag{3.16}$$

is the false alarm rate (FAR) calculated using both $\theta_1(j)$ and the testing prototypes, $0 < \alpha < 1$ is a constant so that $[\alpha p(j)]$ is the step-size for tuning. The specific formulas for the partial derivative $\partial J_1(\theta_1)/\partial\theta_1$ are given in (6-30)-(6-33) of [5].

3.2.3 Partially Dependent Initialization and Optimization for the Type-2 FL-RBC

The $(45M+22)$ design parameters of the type-2 FL-RBC are not directly initialized on the training prototypes. Instead, they are initialized using the type-1 FL-RBC that has just been optimized, as follows:

$$\sigma_{k1}(0) = (1-\gamma)\sigma_k \qquad \sigma_{k2}(0) = (1+\gamma)\sigma_k \tag{3.17}$$

$$\sigma_{k1}^l(0) = (1-\gamma)\sigma_k^l \qquad \sigma_{k2}^l(0) = (1+\gamma)\sigma_k^l \tag{3.18}$$

$$m_{k1}^l(0) = m_k^l - \gamma\sigma_k^l \qquad m_{k2}^l(0) = m_k^l + \gamma\sigma_k^l \tag{3.19}$$

$$g^l(0) = g^l \tag{3.20}$$

where $\gamma \in [0,1]$ (we chose $\gamma = 0.25$ in our experiments). The parameters on the left-hand sides of these equations are the initial values for the type-2 FL-RBC, whereas those on the right-hand sides are the parameters of the type-1 FL-RBC that has just been optimized.

By this kind of initialization, we are using our optimized type-1 FL-RBC parameters to start off the designs of our type-2 FL-RBC parameters, which is why we refer to it as a *partially-dependent initialization.*

The design parameters of the type-2 FL-RBC are then optimized using a steepest descent algorithm to minimize the following objective function:

$$J_2(\theta_2) = \sum_{i=1}^{N} \left\{ y_d(\mathbf{x}_i) - \frac{1}{2}\left[y_l(\mathbf{x}_i|\theta_2) + y_r(\mathbf{x}_i|\theta_2)\right] \right\}^2 \tag{3.21}$$

where θ_2 represents all the parameters of the type-2 FL-RBC that are tuned, $y_l(\mathbf{x}_i|\theta_2)$ and $y_r(\mathbf{x}|\theta_2)$ are the left and right end-points of the type-reduced set so that $[y_l(\mathbf{x}_i|\theta_2)+y_r(\mathbf{x}_i|\theta_2)]/2$ is the output given by the type-2 FL-RBC, and, N and $y_d(\mathbf{x}_i)$ are the same as in (3.14). The tuning procedure used for the type-2 FL-RBC is the same as the one used for the type-1 FL-RBC in (3.15), except that $p(j)$ now represents the FAR of the type-2 FL-RBC calculated using $\theta_2(j)$ and the testing prototypes, and $J_1(\theta_1)$ and θ_1 are replaced by $J_2(\theta_2)$ and θ_2, respectively (a general method to determine $\partial J_2(\theta_2)/\partial\theta_2$ is given on p.p. 406-408 of [5]).

3.3 Non-Adaptive and Adaptive Working Modes

So far, the working modes of all three classifiers are *non-adaptive*, i.e., the decision of one prototype is independent of the decisions of other prototypes. The objective functions (3.14) and (3.21), and the FAR $p(j)$ used in (3.15) are all for this non-adaptive working mode, which means that the type-1 and type-2 FL-RBCs are tuned to optimize their non-adaptive performance.

The classifiers can also operate in a real-time *adaptive* mode, i.e., make decisions based on all data as it *becomes available.* The adaptive working mode can be implemented in different ways. The implementation method we have chosen involves voting. The classifier first makes a non-adaptive decision for the present prototype, and then votes using the non-adaptive decisions that are available for the present and all previous prototypes. For instance, in the tracked vs. wheeled vehicle classification problem, if the number of non-adaptive decisions for the tracked vehicle is greater (or smaller) than that for the wheeled vehicle, then the present prototype is classified as a tracked (or wheeled) vehicle; if the number of non-adaptive decisions for the tracked and wheeled vehicles are equal, then the classifier does not make a final decision until the next prototype is presented. Block diagrams for both the non-adaptive and adaptive working modes are shown in Figure 3.1.

The basic idea behind the adaptive working mode of a classifier is the same as in *decision fusion* [3], namely to combine multiple decisions in order to achieve a more accurate decision. Because a fusion process may produce worse results in some cases, it is only under certain conditions that the adaptive performance of a classifier is superior to its non-adaptive performance.

Theorem 1 *Let p be the FAR of a classifier for its* non-adaptive *mode, i.e., the probability that a prototype is incorrectly classified is p. If $p < 0.5$, then the FAR of the classifier for its* adaptive *mode will be smaller than p after $n_0(p)$, which is determined from Figure B.2.*

The proof is presented in Appendix B.

According to Theorem 1, a sufficient condition for a classifier to have good performance for its adaptive mode is good performance for its non-adaptive mode. So, the type-1 and type-2 FL-RBCs are tuned during the design period to optimize their performance for the non-adaptive mode, and are then tested for both the non-adaptive and adaptive modes. We examine this kind of processing in Chapter 4.

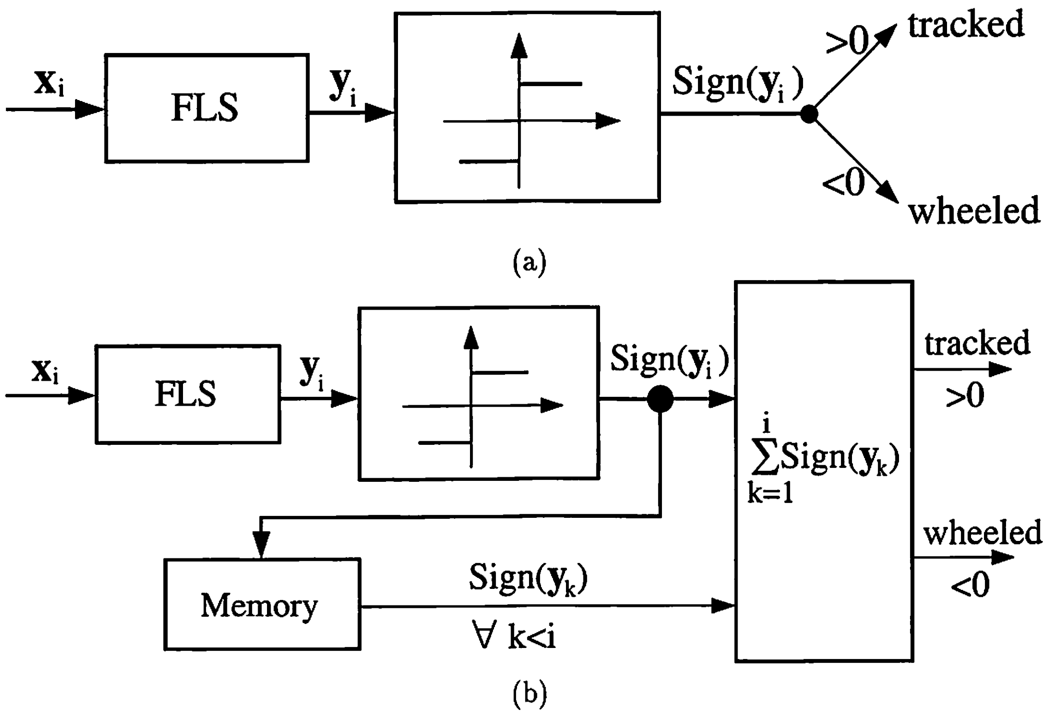


Figure 3.1: Working modes of classifiers: (a) non-adaptive and (b) adaptive.

Chapter 4

Experiments and Results

We evaluated the three classifiers using two groups of experiments, which we call *leave-one-out* [2] and *leave-M-out* (M is the number of vehicles, i.e., $M = 9$ for the tracked vs. wheeled vehicle classification, $M = 5$ for the heavy-tracked vs. light-tracked vehicle classification, and $M = 4$ for the heavy-wheeled vs. light-wheeled vehicle classification) experiments. These experiments were different in the way that the runs were divided into the *training* and *testing* runs.

4.1 Leave-One-Out Experiment

The leave-one-out experiment consisted of as many designs as the number of runs (see Table 2.1, from which it is clear that there can be 89 designs for the tracked vs. wheeled vehicle classification, 61 designs for the heavy-tracked vs. light-tracked vehicle classification, and 28 designs for the heavy-wheeled vs. light-wheeled vehicle classification). In the i-th design, the i-th run was the testing run, and all the other runs were the training runs. How the training and testing runs were used and how the classifier operated are summarized in Table 4.1. Note that:

- The experiments corresponding to the *CN-CN* configuration were completely separate from those corresponding to the *CN-NN-NA* configuration.
- For both configurations, each classifier was designed in its non-adaptive mode, and the design procedure consisted of the training and testing periods. The training runs were used to estimate or optimize parameters of classifiers; whereas, the testing runs were used for *cross-validation* [2], and to determine when to stop training.

- The additional adaptive testing in the *CN-NN-NA* configuration was not performed during the design procedure, but was carried out after the design procedure. The additional testing was performed in order to examine the adaptive mode of a classifier that had just been designed.

For the *CN-CN* configuration, the performance of a classifier in the i-th design was characterized by its FAR, p_i, for the testing; whereas, for the *CN-NN-NA* configuration, the performance of a classifier in the i-th design was characterized by its FARs, p_i and p_i^a, for both the testing and additional testing. In each design:

- The parameters of the Bayesian classifier were estimated based on the training prototypes, using (3.9) and (3.10). For the *CN-CN* configuration, this classifier was then tested for its non-adaptive mode (p_i). For the *CN-NN-NA* configuration, this classifier was tested for both its non-adaptive (p_i) and adaptive (p_i^a) modes.

- The parameters of the type-1 FL-RBC were first initialized based on the training prototypes, using (3.11)-(3.13). For the *CN-CN* configuration, this classifier was tuned and tested for 150 epochs to achieve its best performance (p_i) on the testing prototypes. For the *CN-NN-NA* configuration, this classifier was also tuned and tested for 150 epochs to achieve its best performance (p_i) on the testing prototypes, and an additional test (p_i^a) was then performed.

- The parameters of the type-2 FL-RBC were first initialized based on its just-optimized corresponding type-1 design, using (3.17)-(3.20). For the *CN-CN* configuration, this classifier was tuned and tested for 150 epochs to achieve its best performance (p_i) on the testing prototypes. For the *CN-NN-NA* configuration, this classifier was also tuned and tested for 150 epochs to achieve its best performance (p_i) on the testing prototypes, and an additional test (p_i^a) was then performed.

After all designs were completed, the average and standard deviation of p_i (and of p_i^a when the testing prototypes are non-CPA-based) were calculated. These results are shown in Tables 4.2 and 4.3. Observe that for all three binary classification problems:

- For both the CPA-based and non-CPA-based testing prototypes (Table 4.2 and the first part of Table 4.3), the *non-adaptive performance* of the type-1 and type-2 FL-RBCs are much better than that of the Bayesian classifier; and the non-adaptive performance of the type-2

FL-RBC is better than that of the type-1 FL-RBC. This means that the type-2 FL-RBC is the best of the three classifiers for the non-adaptive working mode.

- For the non-CPA-based testing prototypes (Table 4.3), the *adaptive performance* of each classifier is better than its *non-adaptive performance.* This is consistent with Theorem 1.

- For the non-CPA-based testing prototypes (the second part of Table 4.3), the *adaptive performances* of the type-1 and type-2 FL-RBCs are much better than that of the Bayesian classifier; and the adaptive performance of the type-2 FL-RBC is better than that of the type-1 FL-RBC. This means that the type-2 FL-RBC is the best of the three classifiers for the adaptive working mode.

- In most cases the performance of the type-2 FL-RBC is more than 50% better than that of the type-1 FL-RBC, but the performance of the latter is already so good that the greater than 50% improvement may not be so significant.

4.2 Leave-M-Out Experiment and Bootstrap Estimation

In each design of the leave-M-out experiment, one run of each kind of vehicle was left out as a testing run (see Table 2.1, from which it is clear that there were nine kinds of vehicles in the tracked vs. wheeled classification, five kinds of vehicles in the heavy-tracked vs. light-tracked classification, and four kinds of vehicles in the heavy-wheeled vs. light-wheeled classification), and all the other runs were the training runs. How the training and testing runs were used and how the classifier operated are summarized in Table 4.1. For the *CN-CN* configuration, the performance of a classifier in each design was characterized by its FAR, p_i, for the testing; whereas, for the *CN-NN-NA* configuration, the performance of a classifier in each design was characterized by its FARs, p_i and p_i^a, for both the testing and additional testing.

There should be as many designs as the total number of different *combinations of testing runs*, which, according to Table 2.1, is $15^3 \times 8^5 \times 4 = 442,368,000$ designs for the tracked vs. wheeled vehicle classification, $15^3 \times 8^2 = 216,000$ designs for the heavy-tracked vs. light-tracked vehicle classification, and $8^3 \times 4 = 2,048$ for the heavy-wheeled vs. light-wheeled classification. For each classification problem, we only experimented on 100 such combinations of testing runs (i.e., performed 100 designs). We then computed the average FARs of these 100 combinations, $\hat{p}_{ave}$ of $\{p_1, \ldots, p_{100}\}$ and $\hat{p}_{ave}^a$ of $\{p_1^a, \ldots, p_{100}^a\}$, so as to estimate the mean FARs of each classifier (for both the non-adaptive and adaptive working modes), and finally used the bootstrap method [6][12] to calculate the 95% confidence intervals for $\hat{p}_{ave}$ and $\hat{p}^a_{ave}$.

In each design,

- The parameters of the Bayesian classifier were estimated based on the training prototypes, using (3.9) and (3.10). For the *CN-CN* configuration, this classifier was then tested for its non-adaptive mode (p_i). For the *CN-CN-NA* configuration, this classifier was tested for both its non-adaptive (p_i) and adaptive (p^a_i) modes.

- The parameters of the type-1 FL-RBC were first initialized based on the training prototypes, using (3.11)-(3.13). For the *CN-CN* configuration, this classifier was tuned and tested for 500 epochs, so as to achieve its best performance (p_i) on the testing prototypes. For the *CN-NN-NA* configuration, this classifier was tuned and tested for 500 epochs, so as to achieve its best performance (p_i) on the testing prototypes, and an additional test (p^a_i) was then performed.

- The parameters of the type-2 FL-RBC were first initialized based on its just-optimized corresponding type-1 design, using (3.17)-(3.20). For the *CN-CN* configuration, this classifier was tuned and tested for 500 epochs, so as to achieve its best performance (p_i) on the testing prototypes. For the *CN-NN-NA* configuration, this classifier was tuned and tested for 500 epochs, so as to achieve its best performance (p_i) on the testing prototypes, and an additional test (p^a_i) was then performed.

The bootstrap method is now widely used to calculate confidence intervals for estimates of unknown parameters of a random process when standard methods cannot be applied, e.g., when the number of observations is small. When the estimator for a parameter is a linear (or almost linear) statistic, then it is well-known that the confidence interval provided by the bootstrap method is reliable if this statistic is asymptotically normal [6][12]. In our case, the non-adaptive and adaptive FARs corresponding to different combinations of testing runs, $\{p_1, p_2, \ldots\}$ and $\{p^a_1, p^a_2, \ldots\}$ can each be considered to be independent and identically distributed random variables with finite means and standard deviations (within the interval [0,1]), respectively. According to the Central Limit Theorem [1], the linear statistics $\hat{p}_{ave}$ and $\hat{p}^a_{ave}$ are asymptotically normal, which means that the bootstrap method can therefore provide reliable confidence intervals for $\hat{p}_{ave}$ and $\hat{p}^a_{ave}$.

The bootstrap method that we used to estimate the 95% confidence intervals for $\hat{p}_{ave}$ and $\hat{p}^a_{ave}$ was adapted from Table 2 of [12]. For illustrative purposes, we only describe this procedure for $\hat{p}_{ave}$ as follows (for $\hat{p}^a_{ave}$ all p_i should be replaced by the corresponding p^a_i):

1. Calculate the average $\hat{p}_{ave}$ of $\{p_1, \cdots, p_{100}\}$.

2. Estimate the standard deviation $\hat{\sigma}_p$ of $\{p_1, \cdots, p_{100}\}$ using another bootstrap procedure that is described below.

3. Draw a random sample of size 100, $\{p_1^*, \cdots, p_{100}^*\}$, with replacement from $\{p_1, \cdots, p_{100}\}$.

4. For the resampled set $\{p_1^*, \cdots, p_{100}^*\}$, calculate the sample average, $\hat{p}_{ave}^*$, and estimate its standard deviation (using the bootstrap procedure described below), $\hat{\sigma}_p^*$. Then calculate

$$\hat{\mu}_r^* = \frac{\hat{p}_{ave}^* - \hat{p}_{ave}}{\hat{\sigma}_p^*} \tag{4.1}$$

5. Repeat steps 3 and 4 2000 times, to obtain $\hat{\mu}_{r,1}^*, \cdots, \hat{\mu}_{r,2000}^*$.

6. Sort $\hat{\mu}_{r,1}^*, \cdots, \hat{\mu}_{r,2000}^*$ to obtain $\hat{\mu}_{r,(1)}^* \leq \cdots \leq \hat{\mu}_{r,(2000)}^*$, where $\hat{\mu}_{r,(k)}^*$ is the k-th smallest value of $\left\{\hat{\mu}_{r,1}^*, \cdots, \hat{\mu}_{r,2000}^*\right\}$.

7. Let $q_1 = 50$ and $q_2 = 1951$, then $\left[\hat{p}_{ave} - \hat{\sigma}_p\hat{\mu}_{r,(q1)}^*, \hat{p}_{ave} - \hat{\sigma}_p\hat{\mu}_{r,(q2)}^*\right]$ is a 95% confidence interval for p_{ave} (note that 2.5% of 2000 is 50).

The following bootstrap procedure to estimate $\hat{\sigma}_p$ and $\hat{\sigma}_p^*$ is separate from the procedure used to compute the 95% confidence interval of $\hat{p}_{ave}$, and is adapted from Table 5 of [12] (for $\hat{p}_{ave}^a$, all p_i should be replaced by the corresponding p_i^a):

1. Draw a random sample of size 100, $\{p_1', \cdots, p_{100}'\}$, with replacement from $\{p_1, \cdots, p_{100}\}$ (or from $\{p_1^*, \cdots, p_{100}^*\}$ for $\hat{\sigma}_p^*$).

2. Calculate the sample average, $\hat{p}_{ave}'$, for the resampled set $\{p_1', \cdots, p_{100}'\}$.

3. Repeat steps 1 and 2 200 times, to obtain $\hat{p}_{ave,1}', \cdots, \hat{p}_{ave,200}'$.

4. $\hat{\sigma}_p$ (or $\hat{\sigma}_p^*$) is calculated as

$$\hat{\sigma}_p = \sqrt{\frac{1}{200-1}\sum_{i=1}^{200}\left(\hat{p}_{ave,i}' - \frac{1}{200}\sum_{j=1}^{200}\hat{p}_{ave,j}'\right)^2} \tag{4.2}$$

Our bootstrap results are shown in Tables 4.4 and 4.5. Observe that for all three binary classification problems:

- For the CPA-based testing prototypes (Table 4.4), the *non-adaptive performance* of the type-2 FL-RBC is always close to and slightly better than that of the type-1 FL-RBC, and is better than that of the Bayesian classifier.

- For the non-CPA-based testing prototypes (the first part of Table 4.5), the *non-adaptive performance* of the type-2 FL-RBC is always close to that of the type-1 FL-RBC, and is better than that of the Bayesian classifier (the type-2 FL-RBC is slightly better than the type-1 FL-RBC for the tracked vs. wheeled vehicle and heavy-wheeled vs. light-wheeled vehicle classification problems, but is slightly worse than the type-1 FL-RBC for the heavy-tracked vs. light-tracked vehicle classification problem).

- For the non-CPA-based testing prototypes (Table 4.5), the *adaptive performance* of each classifier is always better than its non-adaptive performance. This is consistent with our analysis in Section 3.3, in Theorem 1.

- For the non-CPA-based testing prototypes (the second part of Table 4.5), the *adaptive performance* of the type-2 FL-RBC is always close to that of the type-1 FL-RBC, and is better than that of the Bayesian classifier (the type-2 FL-RBC is slightly better than the type-1 FL-RBC for the tracked vs. wheeled vehicle and heavy-tracked vs. light-tracked vehicle classification problems, but slightly worse than the type-1 FL-RBC for the heavy-wheeled vs. light-wheeled vehicle classification problem).

4.3 Performance Evaluation

Comparing Tables 4.2 and 4.4, and 4.3 and 4.5, we observe that the performance estimated for each classifier from the leave-one-out and leave-M-out experiments are quite different. Because the adaptive working mode is related to the non-adaptive working mode, and it is very difficult to establish a probability model for the adaptive working mode, we only explain the difference for the non-adaptive working mode. The performance differences can be explained from two viewpoints, the number of testing prototypes and the number of training prototypes.

4.3.1 The Number of Testing Prototypes and the Upper Bound of FAR

In both the leave-one-out and leave-M-out experiments, the non-adaptive performance of each classifier is characterized by its FAR, which is the ratio of the number of incorrectly classified testing prototypes to the total number of testing prototypes. Note that the FAR calculated in this way is only a statistic of the testing prototypes and is therefore random, and is related to (but not equal to) the real FAR of a classifier.

Let p be the real but unknown FAR of a classifier in the non-adaptive working mode, and assume that it is uniformly distributed over $[0, 1]$. Assume, also, that the testing prototypes are independently and identically distributed, with probability p to be incorrectly classified, and probability $1 - p$ to be correctly classified. Let n be the total number of testing prototypes, and k be the number of incorrectly classified testing prototypes. Then the probability that k out of n testing prototypes are incorrectly classified complies to the Binomial distribution, i.e.

$$Pr(k|p) = \begin{pmatrix} n \\ k \end{pmatrix} p^k(1-p)^{(n-k)} \tag{4.3}$$

We obtained n and k from experiments, and wanted to estimate p from them. It is well-known that the maximum-likelihood estimate, $\hat{p}$, of p is [1]:

$$\hat{p} = \frac{k}{n} \tag{4.4}$$

and that this maximum-likelihood estimate is consistent, which means that $\hat{p}$ is more reliable as n becomes larger. Since the number of testing prototypes is different[7] for the leave-one-out and leave-M-out experiment, the estimate of p from these two experiments should be different. It is unfair to just compare $\hat{p}$ of these two experiments to see whether they are consistent, and which estimate to use. We need also consider the effect of n on $\hat{p}$. So, we used the 95%-confidence upper bound of p, which is a function of both n and $\hat{p}$, to compare the results of the leave-one-out and leave-M-out experiments.

Our goal is to find a small amount, δ, which is a function of n and $\hat{p}$, so that $\hat{p} + \delta$ is an upper bound of the real p with 95% confidence, i.e.

$$Pr\left(p \leq \hat{p} + \delta | k\right) \geq 0.95 \tag{4.5}$$

where 0.95 represents the 95% confidence. The method that we used to derive the upper bound

[7] For example, for the tracked vs. wheeled vehicle classification problem, for the *CN-CN* configuration where the testing prototypes are CPA-based (see Section 2.2.1), the number of testing prototypes for the leave-one-out experiment is either 80 (when a tracked run is left out) or 160 (when a wheeled run is left-out), and the number of testing prototypes for the leave-M-out experiment is $80 \times 5 + 160 \times 4 = 1040$ (80 for each left-out tracked run and 160 for each left-out wheeled run)

$\hat{p}+\delta$ is adapted from p.p. 484-485 of [2].

$$
\begin{aligned}
& Pr\,(p \le \hat{p}+\delta | k) \ge 0.95 \\
\Longrightarrow\ & \int_0^{\hat{p}+\delta} Pr\,(p|k)\,dp \ge 0.95 \\
\Longrightarrow\ & \int_0^{\hat{p}+\delta} \frac{Pr\,(k|p)\,Pr(p)}{Pr(k)} dp \ge 0.95 \\
\Longrightarrow\ & \int_0^{\hat{p}+\delta} \begin{pmatrix} n \\ k \end{pmatrix} p^k (1-p)^{(n-k)} dp \ge 0.95 \int_0^1 \begin{pmatrix} n \\ k \end{pmatrix} p^k (1-p)^{(n-k)} dp \\
\Longrightarrow\ & \int_0^{\hat{p}+\delta} p^k (1-p)^{n-k} dp \ge 0.95 B(k+1, n-k+1) \qquad (4.6)
\end{aligned}
$$

where $B(\cdot,\cdot)$ is the beta function [1], and we have assumed p is uniformly distributed so that the probability density function of p is 1 in the interval [0,1]. We need to solve (4.6) for δ, but there is no closed-form solution of (4.6) for δ. Numerical solutions of $\hat{p}+\delta$ are shown in Fig. 4.1 as a function of $\hat{p}$, for $n = 80, 160, 1040$ and $t = 0.05$. Observe that for a given $\hat{p}$ the upper bound $\hat{p}+\delta$ decreases as n increases, which means that the upper bound for the real FAR is tighter when the number of testing prototypes is larger.

The upper bounds of the non-adaptive FARs for the leave-one-out and leave-M-out experiments (which are determined from Fig. 4.1), based on $\hat{p}$ and the number of testing prototypes for the *CN-CN* configuration (or the average number of testing prototypes for the *CN-NN-NA* configuration), are shown in Tables 4.6 and 4.7. For each configuration, we compared the two $\hat{p}$ of a classifier (one from the leave-one-out expriment, and the other from the leave-M-out experiment), and also its two $\hat{p}+\delta$. For example, in Table 4.6, focusing on the tracked vs. wheeled vehicle classification, the two $\hat{p}$ of the type-1 FL-RBC are 0.7614% and 9.8596%, which means that the $\hat{p}$ of the leave-M-out experiment is approximately 13 times the $\hat{p}$ of the leave-one-out experiment; whereas, the two $\hat{p}+\delta$ of this type-1 FL-RBC are 5.1845% (or 3.2536%) and 11.5904% which means that the $\hat{p}+\delta$ of the leave-M-out experiment is approximately three (or four) times the $\hat{p}+\delta$ of the leave-one-out experiment. We find, therefore, that for each configuration and each classifier, the difference between the upper bounds, $\hat{p}+\delta$, of the leave-one-out and leave-M-out experiments is smaller than the difference between their two $\hat{p}$. This means that it is better to use the upper bound of the FAR to estimate the performance of a classifier.

4.3.2 The Number of Training Prototypes

In order to tune the parameters of a classifier well, we always require that the number of training prototypes be larger than the number of tuned classifier parameters. For both the leave-one-out and leave-M-out experiments, when the prototypes generated from *different* runs are assumed to be independent and identically distributed, the number of training prototypes is always much larger than the number of classifier parameters, and it is always an over-determined problem to specify the parameters of a classifier. Consequently, the classifiers obtained from these two experiments should have similar performances. However, the prototypes generated from one run cannot be assumed to comply exactly to the same distribution of the prototypes generated from another run (see Section 2.4 that the run-mean and run-standard-deviation vary from run to run, even for the same kind of vehicle). Given two training prototypes, if they are generated from the same run, then their impact on the classifier parameters are similar; whereas, if they are generated from different runs, then their impact on the classifier parameters can be quite different, so that the generalization property of the classifier for the latter case may be better than for the former case. This means that it is the number of training runs, rather than the number of training prototypes, that has the greater impact on the training of classifiers. The percentage of runs used for training is shown in Table 4.8, where, for instance, for the tracked vs. wheeled classification, 88 runs of 89 runs were used for training in the leave-one-out experiment, and 80 runs of 89 runs were used for training in the leave-M-out experiment. Observe that the difference between the two percentages is about 10%. When the total number of runs is not large, each training run has great influence on the parameter optimization, and the difference between training runs may affect our evaluation of classifiers greatly.

We therefore advocate the leave-one-out method in our study about the binary classification of ground vehicles, because in this way we can make use of most of the data for optimizing the classifier parameters, as well as have data for cross-validation.

Table 4.1: Summary of training, testing and working modes used in the leave-one-out and leave-M-out experiments.

Configuration[8]	Training	Testing	Additional Testing
CN-CN	CPA-based & non-adaptive	CPA-based & non-adaptive (p_i)	
CN-NN-NA	CPA-based & non-adaptive	non-CPA-based & non-adaptive (p_i)	non-CPA-based & adaptive (p_i^a)

[8]The first two letters represent the prototypes (C for CPA-based, and N for non-CPA-based) and the working mode (N for non-adaptive, and A for adaptive) used during training, the second two letters represent the prototypes and the working mode used during testing, and the last two letters (if they exist) represent the prototypes and the working mode during the additional testing.

Table 4.2: The average and standard deviation (SD) of p_i for the *CN-CN* configuration of the leave-one-out experiment.

Classification Problem	Measure	Bayesian Classifier	Type-1 FL-RBC	Type-2 FL-RBC
tracked vs. wheeled	Average	12.5421%	0.7614%	0.3252%
	SD	0.184408	0.020574	0.011420
heavy-tracked vs. light-tracked	Average	6.8648%	1.2008%	0.2725%
	SD	0.132967	0.056824	0.019903
heavy-wheeled vs. light-wheeled	Average	7.4554%	0.4018%	0.0759%
	SD	0.138181	0.009659	0.005359

Table 4.3: The average and standard deviation (SD) of p_i and p_i^a for the *CN-NN-NA* configuration of the leave-one-out experiment.

Non-Adaptive Working Mode (p_i)

Classification Problem	**Measure**	**Bayesian Classifier**	**Type-1 FL-RBC**	**Type-2 FL-RBC**
tracked vs.	Average	23.1104%	1.0721%	0.6961%
wheeled	SD	0.128409	0.023463	0.019730
heavy-tracked vs.	Average	15.0932%	0.8056%	0.3757%
light-tracked	SD	0.185691	0.028269	0.016577
heavy-wheeled vs.	Average	22.8095%	1.6276%	0.2782%
light-wheeled	SD	0.187344	0.033729	0.015892

Adaptive Working Mode (p_i^a)

Classification Problem	**Measure**	**Bayesian Classifier**	**Type-1 FL-RBC**	**Type-2 FL-RBC**
tracked vs.	Average	9.1027%	0.0645%	0.0586%
wheeled	SD	0.165858	0.005090	0.004513
heavy-tracked vs.	Average	7.9063%	0.2727%	0.0671%
light-tracked	SD	0.250965	0.021967	0.010300
heavy-wheeled vs.	Average	14.1777%	0.5617%	0.1322%
light-wheeled	SD	0.311986	0.036715	0.017290

Table 4.4: The estimated mean FAR, $\hat{p}_{ave}$, and its 95% confidence interval for the *CN-CN* configuration of the leave-M-out experiment.

Classification Problem	Measure	Bayesian Classifier	Type-1 FL-RBC	Type-2 FL-RBC
tracked	$\hat{p}_{ave}$	13.0567%	9.8596%	9.8308%
vs.	Confidence	[12.0707%,	[9.3165%,	[9.2789%,
wheeled	Interval	14.1358%]	10.4797%]	10.4212%]
heavy-tracked	$\hat{p}_{ave}$	7.8450%	3.8550%	3.8125%
vs.	Confidence	[6.7277%,	[3.2142%,	[3.1797%,
light-tracked	Interval	8.9307%]	4.6167%]	4.7113%]
heavy-wheeled	$\hat{p}_{ave}$	8.8654%	7.4609%	7.3297%
vs.	Confidence	[7.4589%,	[6.9012%,	[6.7675%
light-wheeled	Interval	10.0333%]	8.0447%]	7.8947%]

Table 4.5: The estimated mean FARs, $\hat{p}_{ave}$ and $\hat{p}^a_{ave}$, and their 95% confidence intervals for the *CN-NN-NA* configurations of the leave-M-out experiment.

Non-Adaptive Working Mode (p_{ave})

Classification Problem	Measure	Bayesian Classifier	Type-1 FL-RBC	Type-2 FL-RBC
tracked	$\hat{p}_{ave}$	22.8399%	11.6147%	11.5752%
vs.	Confidence	[22.1030%,	[11.0081%,	[10.9096%,
wheeled	Interval	23.5663%]	12.2232%]	12.2198%]
heavy-tracked	$\hat{p}_{ave}$	15.4553%	6.3250%	6.3713%
vs.	Confidence	[14.5454%,	[5.7580%,	[5.6901%,
light-tracked	Interval	16.4035%]	6.9621%]	7.1727%]
heavy-wheeled	$\hat{p}_{ave}$	18.9737%	7.7949%	7.6813%
vs.	Confidence	[17.4422%,	[7.2152%,	[7.0500%
light-wheeled	Interval	20.6121%]	8.3878%]	8.2714%]

Adaptive Working Mode (p^a_{ave})

Classification Problem	Measure	Bayesian Classifier	Type-1 FL-RBC	Type-2 FL-RBC
tracked	$\hat{p}^a_{ave}$	9.3122%	3.9645%	3.7425%
vs.	Confidence	[8.3335%,	[3.1659%,	[2.8513%
wheeled	Interval	10.3089%]	4.7350%]	4.5861%]
heavy-tracked	$\hat{p}^a_{ave}$	7.5659%	1.8770%	1.8588%
vs.	Confidence	[6.1229%,	[1.3294%,	[1.1669%,
light-tracked	Interval	8.6986%]	2.2792%]	2.6834%]
heavy-wheeled	$\hat{p}^a_{ave}$	11.4211%	1.5687%	1.7216%
vs.	Confidence	[9.0436%,	[1.2293%,	[1.2756%,
light-wheeled	Interval	14.0484%]	1.8174%]	2.1166%]

Table 4.6: The upper bound of the FAR for the *CN-CN* configuration of each classifier.

Tracked vs. Wheeled Vehicle Classification

Experiment	Measure	Bayesian Classifier	Type-1 FL-RBC	Type-2 FL-RBC
	$\hat{p}$	12.5421%	0.7614%	0.3252%
Leave-one-out	$\hat{p}+\delta$, $n=80$	20.1383%	5.1845%	4.4966%
	$\hat{p}+\delta$, $n=160$	17.6383%	3.2536%	2.5735%
Leave-M-out	$\hat{p}$	13.0567%	9.8596%	9.8308%
	$\hat{p}+\delta$, $n=1040$	14.9596%	11.5904%	11.5616%

Heavy-tracked vs. Light-tracked Vehicle Classification

Experiment	Measure	Bayesian Classifier	Type-1 FL-RBC	Type-2 FL-RB
	$\hat{p}$	6.8648%	1.2008%	0.2725%
Leave-one-out	$\hat{p}+\delta$, $n=80$	13.4273%	5.7321%	4.4112%
Leave-M-out	$\hat{p}$	7.8450%	3.8550%	3.8125%
	$\hat{p}+\delta$, $n=400$	10.5012%	5.8862%	5.8438%

Heavy-wheeled vs. Light-wheeled Vehicle Classification

Experiment	Measure	Bayesian Classifier	Type-1 FL-RBC	Type-2 FL-RBC
	$\hat{p}$	7.4554%	0.4018%	0.0759%
Leave-one-out	$\hat{p}+\delta$, $n=160$	11.8304%	2.7456%	2.1027%
Leave-M-out	$\hat{p}$	8.6547%	7.4600%	7.3297%
	$\hat{p}+\delta$, $n=640$	10.7469%	9.4921%	9.3609%

Table 4.7: The upper bound of the FAR (non-adaptive) for the *CN-NN-NA* configuration of each classifier (n is the average number of testing prototypes).

Tracked vs. Wheeled Vehicle Classification

Experiment	Measure	Bayesian Classifier	Type-1 FL-RBC	Type-2 FL-RBC
	$\hat{p}$	23.1104%	1.0721%	0.6961%
Leave-one-out	$\hat{p}+\delta$, $n=94$	31.1792%	4.9359%	4.5321%
Leave-M-out	$\hat{p}$	22.8399%	11.6147%	11.5752%
	$\hat{p}+\delta$, $n=756$	25.4854%	13.7311%	13.6916%

Heavy-tracked vs. Light-tracked Vehicle Classification

Experiment	Measure	Bayesian Classifier	Type-1 FL-RBC	Type-2 FL-RBC
	$\hat{p}$	15.0932%	0.8056%	0.3757%
Leave-one-out	$\hat{p}+\delta$, $n=108$	21.8111%	4.2605%	3.6305%
Leave-M-out	$\hat{p}$	15.4553%	6.3250%	6.3713%
	$\hat{p}+\delta$, $n=521$	18.3344%	8.4363%	8.4826%

Heavy-wheeled vs. Light-wheeled Vehicle Classification

Experiment	Measure	Bayesian Classifier	Type-1 FL-RBC	Type-2 FL-RBC
	$\hat{p}$	22.8095%	1.6276%	0.2782%
Leave-one-out	$\hat{p}+\delta, = 63$	33.0659%	7.6105%	5.5355%
Leave-M-out	$\hat{p}$	18.9737%	7.7949%	7.6813%
	$\hat{p}+\delta$, $n=234$	23.6746%	11.6411%	11.5164%

Table 4.8: The percentage of runs that are used for training.

Classification Problem	Leave-one-out	Leave-M-out
tracked vs. wheeled	98.88%	89.89%
heavy-tracked vs. light-tracked	98.36%	91.80%
heavy-wheeled vs. light-wheeled	96.43%	85.71%

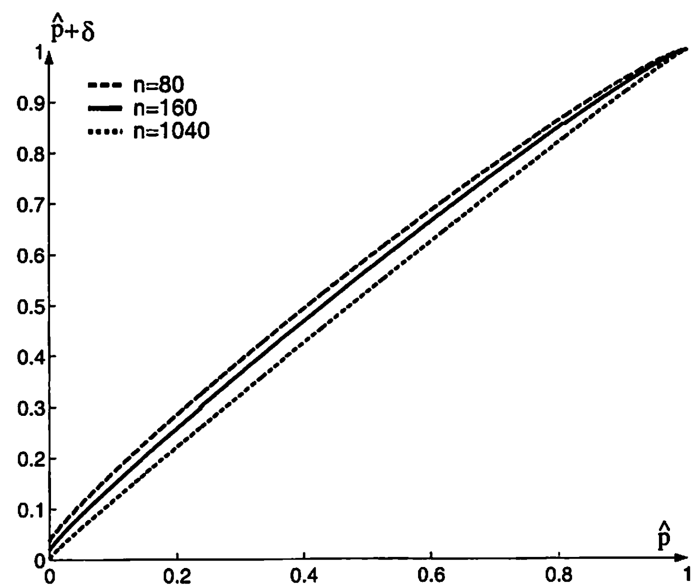


Figure 4.1: 95% confidence upper bound of the real FAR, for three values of n, computed so that the real FAR has 95% probability to be between 0 and its upper bound.

Chapter 5

Blind Tests

Besides the acoustic data in ACIDS, we also had 51 blind (unlabeled) runs. The only information that was provided to us about the blind data was that they were collected in the normal terrain. The goal of our blind tests was to classify these blind runs as either *tracked* or *wheeled* vehicles. In this chapter, we describe these blind tests, give the classification results for the blind runs, and provide the evaluation results of the blind tests.

5.1 CPA-Based Prototypes and Non-CPA-Based Prototypes

For each blind run, we first located its CPA and then truncated a fixed number of data around the CPA to generate 80 CPA-based prototypes. Each prototype contained one second of data, and adjacent prototypes had 50% of overlap.

The non-CPA-based prototypes were generated by using an energy test[9]: the classifier was activated once the energy of a data-block (one second) was 0.1dB above the background noise level (the first data-block of each run was assumed to be full of noise).

[9] Again, the threshold of 0.1dB was chosen based on the energy distribution of *all* blind runs. The energy distribution of these blind runs was slightly different than that of the design runs in ACIDS. We observed that if we had used an 8dB threshold, which was the threshold used for generating non-CPA-based prototypes from the ACIDS design runs, then for some blind runs the classifier would never be activated; hence, we lowered the threshold for the blind runs.

5.2 Non-Adaptive and Adaptive Working Modes

The type-1 and type-2 FL-RBCs were designed using the design data (89 labeled runs of the normal terrain in ACIDS) and the leave-one-out method; hence, each classifier was actually a classifier cluster, and consisted of 89 designs. The classifier cluster made a decision for one blind prototype by voting on the classification results given by its 89 component designs—if the number of designs that classified this blind prototype as a tracked vehicle was larger (smaller) than the number of designs that classified this blind prototype as a wheeled vehicle, then the classifier cluster classified this blind prototype as a tracked (wheeled) vehicle.

In the non-adaptive working mode, the decision made for one blind prototype was independent of the decisions made for the others, and for each blind run if the number of prototypes classified as a tracked vehicle was larger (smaller) than the number of prototypes classified as a wheeled vehicle, then this blind run was classified as a tracked (wheeled) vehicle; if these two numbers were the same, then this run was undetermined.

In the adaptive working mode, the classifier cluster first made a non-adaptive decision on the current prototype and then voted on the non-adaptive decisions of all the previous and current prototypes to reach an adaptive decision for the current prototype. We applied the following two voting strategies:

- The non-adaptive decisions of the previous and current prototypes were linearly combined with *equal* weights. The sign of this combined result was the adaptive decision of the current prototype.

- The non-adaptive decisions of the previous and current prototypes were linearly combined with *unequal* weights. The weight for each non-adaptive decision was proportional to the SNR of the corresponding prototype, i.e.,

$$SNR\ of\ the\ n^{th}\ prototype = 10\log_{10}\left(\frac{energy\ of\ the\ n^{th}\ prototype}{energy\ of\ the\ 1^{st}\ prototype}\right) \tag{5.1}$$

$$w_n = \frac{SNR\ of\ the\ n^{th}\ prototype}{\sum_j SNR\ of\ the\ j^{th}\ prototype} \tag{5.2}$$

where energy of the first data block (prototype) of each run is considered to be background noise, w_n represents the weight of the n-th prototype, and the summation is over all previous and current prototypes. The sign of this combined result was the adaptive decision of the current prototype.

Because the adaptive decision for the last prototype of each run combined the non-adaptive decisions of all prototypes of this run, we used it as the decision for the entire run.

Note that the decision for each run in the CN experiment (CPA-based blind prototypes and the non-adaptive working mode of classifiers) is the same as the decision of this run in the CAE experiment (CPA-based blind prototypes and the adaptive working mode of classifiers with equal weights), because in the CN experiment the decision for one run depends on the non-adaptive decisions of all CPA-based prototypes in that run; in the CAE experiment, the adaptive decision for the last prototype of each run also depends on the non-adaptive decisions of all CPA-based prototypes in that run; and the dependency ways for the two experiments are the same.

5.3 Results

We carried out five experiments, where the prototypes of blind runs and the working modes of two FL-RBC clusters are summarized in Table 5.1.

Table 5.1: Summary of blind tests.

	Experiment[10]	Prototype	Working Mode
CN:	Table 5.2	CPA-based	non-adaptive
CAE:	Table 5.3	CPA-based	adaptive with equal weights
CAU:	Table 5.4	CPA-based	adaptive with unequal weights
NAE:	Table 5.5	non-CPA-based	adaptive with equal weights
NAU:	Table 5.6	non-CPA-based	adaptive with unequal weights

Details of each experiment for both type-1 and type-2 FL-RBC clusters are summarized in Tables 5.2-5.6. The final decisions for each blind run, provided by the type-2 FL-RBC cluster of these five experiments, are summarized in Table 5.7 (note that the CN and CAE decisions for each run are the same), where the vote decision of each run was obtained by voting on the four decisions for this run provided by the CN/CAE, CAU, NAE and NAU experiments. These results associated with the type-2 FL-RBC cluster were sent to our sponsor at the Army Research Laboratory, who

[10]The blind experiment is named according to the following rule: the first letter represents which kind of prototypes (*C* for CPA-based and *N* for non-CPA-based) of blind runs are used; the second letter represents which working mode (*N* for non-adaptive and *A* for adaptive) of the classifier is used; and the last letter (if it exists) represents which voting strategy (*E* for equally weighted and *U* for unequally weighted) of the adaptive mode is used.

then scored them. The scores are summarized in Table 5.8. Observe that the scores for the NAE experiment is the highest, although the scores for all four experiments are close; and all scores are quite high, which demonstrates that our type-2 FL-RBC is providing excellent results.

Table 5.2: The percentage of prototypes that were classified as a tracked (T), a wheeled (W), or an undetermined (O) vehicle, and the final decision for each blind run for the CN experiments.

	Type-1			Type-2		
Run	Tracked	Wheeled	Decision	Tracked	Wheeled	Decision
run-f005	100%	0%	T	100%	0%	T
run-f006	96.25%	3.75%	T	97.5%	2.5%	T
run-f007	98.75%	1.25%	T	98.75%	1.25%	T
run-f008	92.5%	7.5%	T	93.75%	6.25%	T
run-f010	97.5%	2.5%	T	98.75%	1.25%	T
run-f019	100%	0%	T	100%	0%	T
run-f020	93.75%	6.25%	T	95%	5%	T
run-f021	100%	0%	T	100%	0%	T
run-f022	93.75%	6.25%	T	96.25%	3.75%	T
run-f077	100%	0%	T	100%	0%	T
run-f078	100%	0%	T	100%	0%	T
run-f079	100%	0%	T	100%	0%	T
run-f080	100%	0%	T	100%	0%	T
run-f085	100%	0%	T	100%	0%	T
run-f086	100%	0%	T	100%	0%	T
run-f087	100%	0%	T	100%	0%	T
run-f088	100%	0%	T	100%	0%	T
run-f093	1.25%	98.75%	W	1.25%	98.75 %	W
run-f094	17.5%	82.5%	W	15%	85%	W
run-f095	50%	50%	O	33.75%	66.25%	W
run-f096	42.5%	57.5%	W	41.25%	58.75%	W
run-f101	37.5%	62.5%	W	41.25%	58.75%	W
run-f102	60%	40%	T	68.75%	31.25%	T
run-f103	43.75%	56.25%	W	52.5%	47.5%	T
run-f104	70%	30%	T	80%	20%	T
run-f109	13.75%	86.25%	W	16.25%	83.75%	W
run-f110	58.75%	41.25%	T	50%	50%	O
run-f111	23.75%	76.25%	W	23.75%	76.25%	W
run-f112	8.75%	91.25%	W	20%	80%	W
run-f117	10%	90%	W	10%	90%	W
run-f118	6.25%	93.75%	W	10%	90%	W
run-f119	15%	85%	W	16.25%	83.75%	W
run-f120	25%	75%	W	31.25%	68.75%	W
run-f173	98.75%	1.25%	T	98.75%	1.25%	T
run-f174	96.25%	3.75%	T	98.75%	1.25%	T
run-f175	98.75%	1.25%	T	100%	0%	T
run-f176	98.75%	1.25%	T	98.75%	1.25%	T
run-f181	88.75%	11.25%	T	97.5%	2.5%	T
run-f182	95%	5%	T	98.75%	1.25%	T
run-f183	91.25%	8.75%	T	97.5%	2.5%	T
run-f184	91.25%	8.75%	T	97.5%	2.5%	T
run-f185	100%	0%	T	100%	0%	T
run-f186	97.5%	2.5%	T	100%	0%	T
run-f187	100%	0%	T	100%	0%	T
run-f188	98.75%	1.25%	T	100%	0%	T
run-f189	98.75%	1.25%	T	100%	0%	T
run-f190	96.25%	3.75%	T	97.5%	2.5%	T
run-f197	53.75%	46.25%	T	63.75%	36.25%	T
run-f198	56.25%	43.75%	T	77.5%	22.5%	T
run-f199	68.75%	31.25%	T	70%	30%	T
run-f200	67.5%	32.5%	T	78.75%	21.25%	T

Table 5.3: The percentage of prototypes that were classified as a tracked (T), a wheeled (W), or an undertermined (O) vehicle, and the final decision for each blind run for the CAE experiments.

Run	Type-1 Tracked	Type-1 Wheeled	Type-1 Decision	Type-2 Tracked	Type-2 Wheeled	Type-2 Decision
run-f005	100%	0%	T	100%	0%	T
run-f006	100%	0%	T	100%	0%	T
run-f007	100%	0%	T	100%	0%	T
run-f008	100%	0%	T	100%	0%	T
run-f010	100%	0%	T	100%	0%	T
run-f019	100%	0%	T	100%	0%	T
run-f020	100%	0%	T	100%	0%	T
run-f021	100%	0%	T	100%	0%	T
run-f022	100%	0%	T	100%	0%	T
run-f077	100%	0%	T	100%	0%	T
run-f078	100%	0%	T	100%	0%	T
run-f079	100%	0%	T	100%	0%	T
run-f080	100%	0%	T	100%	0%	T
run-f085	100%	0%	T	100%	0%	T
run-f086	100%	0%	T	100%	0%	T
run-f087	100%	0%	T	100%	0%	T
run-f088	100%	0%	T	100%	0%	T
run-f093	0%	100%	W	0%	100 %	W
run-f094	7.5%	85%	W	0%	98.75%	W
run-f095	71.25%	10%	O	8.75%	86.25%	W
run-f096	1.25%	93.75%	W	0%	93.75%	W
run-f101	2.5%	95%	W	13.75%	80%	W
run-f102	100%	0%	T	100%	0%	T
run-f103	15%	78.75%	W	76.25%	20%	T
run-f104	96.25%	1.25%	T	100%	0%	T
run-f109	21.25%	77.5%	W	23.75%	75%	W
run-f110	86.25%	3.75%	T	46.25%	46.25%	O
run-f111	30%	63.75%	W	20%	75%	W
run-f112	0%	100%	W	0%	100%	W
run-f117	3.75%	91.25%	W	3.75%	91.25%	W
run-f118	0%	98.75%	W	0%	98.75%	W
run-f119	0%	100%	W	0%	100%	W
run-f120	22.5%	75%	W	26.25%	72.5%	W
run-f173	100%	0%	T	100%	0%	T
run-f174	100%	0%	T	100%	0%	T
run-f175	100%	0%	T	100%	0%	T
run-f176	100%	0%	T	100%	0%	T
run-f181	100%	0%	T	100%	0%	T
run-f182	100%	0%	T	100%	0%	T
run-f183	100%	0%	T	100%	0%	T
run-f184	97.5%	1.25%	T	97.5%	1.25%	T
run-f185	100%	0%	T	100%	0%	T
run-f186	100%	0%	T	100%	0%	T
run-f187	100%	0%	T	100%	0%	T
run-f188	100%	0%	T	100%	0%	T
run-f189	100%	0%	T	100%	0%	T
run-f190	100%	0%	T	100%	0%	T
run-f197	58.75%	33.75%	T	76.25%	17.5%	T
run-f198	47.5%	50%	T	65%	31.25%	T
run-f199	50%	48.75%	T	50%	48.75%	T
run-f200	37.5%	61.25%	T	58.75%	36.25%	T

Table 5.4: The percentage of prototypes that were classified as a tracked (T), a wheeled (W), or an undetermined (O) vehicle, and the decision for each blind run for the CAU experiments.

	Type-1			Type-2		
Run	Tracked	Wheeled	Decision	Tracked	Wheeled	Decision
run-f005	100%	0%	T	100%	0%	T
run-f006	100%	0%	T	100%	0%	T
run-f007	100%	0%	T	100%	0%	T
run-f008	100%	0%	T	100%	0%	T
run-f010	100%	0%	T	100%	0%	T
run-f019	100%	0%	T	100%	0%	T
run-f020	100%	0%	T	100%	0%	T
run-f021	100%	0%	T	100%	0%	T
run-f022	100%	0%	T	100%	0%	T
run-f077	100%	0%	T	100%	0%	T
run-f078	100%	0%	T	100%	0%	T
run-f079	100%	0%	T	100%	0%	T
run-f080	100%	0%	T	100%	0%	T
run-f085	100%	0%	T	100%	0%	T
run-f086	100%	0%	T	100%	0%	T
run-f087	100%	0%	T	100%	0%	T
run-f088	100%	0%	T	100%	0%	T
run-f093	0%	100%	W	0%	100 %	W
run-f094	18.75%	81.25%	W	6.25%	93.75%	W
run-f095	96.25%	3.75%	T	70%	30%	W
run-f096	6.25%	93.75%	W	3.75%	96.25%	W
run-f101	2.5%	97.5%	W	12.5%	87.5%	W
run-f102	72.5%	27.5%	T	80%	20%	T
run-f103	23.75%	76.25%	W	45%	55%	W
run-f104	77.5%	22.5%	T	98.75%	1.25%	T
run-f109	22.5%	77.5%	W	23.75%	76.25%	W
run-f110	85%	15%	W	3.75%	96.25%	W
run-f111	36.25%	63.75%	W	20%	80%	W
run-f112	0%	100%	W	0%	100%	W
run-f117	3.75%	96.25%	W	3.75%	96.25%	W
run-f118	1.25%	98.75%	W	1.25%	98.75%	W
run-f119	2.5%	97.5%	T	2.5%	97.5%	T
run-f120	25%	75%	W	27.5%	72.5%	W
run-f173	100%	0%	T	100%	0%	T
run-f174	100%	0%	T	100%	0%	T
run-f175	100%	0%	T	100%	0%	T
run-f176	100%	0%	T	100%	0%	T
run-f181	100%	0%	T	100%	0%	T
run-f182	100%	0%	T	100%	0%	T
run-f183	100%	0%	T	100%	0%	T
run-f184	98.75%	1.25%	T	98.75%	1.25%	T
run-f185	100%	0%	T	100%	0%	T
run-f186	100%	0%	T	100%	0%	T
run-f187	100%	0%	T	100%	0%	T
run-f188	100%	0%	T	100%	0%	T
run-f189	100%	0%	T	100%	0%	T
run-f190	100%	0%	T	100%	0%	T
run-f197	70%	30%	T	82.5%	17.5%	T
run-f198	55%	45%	T	71.25%	28.75%	T
run-f199	56.25%	43.75%	T	56.25%	43.75%	T
run-f200	47.5%	52.5%	T	66.25%	33.75%	T

Table 5.5: The percentage of prototypes that were classified as a tracked (T), a wheeled (W), or an undetermined (O) vehicle, and the decision for each blind run (the number in the bracket represents the number of non-CPA-based prototypes in that run) for the NAE experiments.

	Type-1			Type-2		
Run	Tracked	Wheeled	Decision	Tracked	Wheeled	Decision
run-f005 (683)	97.6574%	1.9034%	T	98.0966%	0.7321%	T
run-f006 (146)	100%	0%	T	100%	0%	T
run-f007 (356)	100%	0%	T	100%	0%	T
run-f008 (470)	79.1489%	20.6383%	T	79.1489%	20.6383%	T
run-f010 (798)	79.4486%	20.4261%	T	78.9474%	20.9273%	T
run-f019 (230)	97.8261%	0.4348%	T	97.8261%	0.4348%	T
run-f020 (234)	62.3932%	37.1795%	T	66.2393%	32.9060%	T
run-f021 (277)	86.2816%	12.6354%	T	83.0325%	15.5235%	T
run-f022 (294)	57.1429%	42.5170%	T	55.7823%	43.5374%	T
run-f077 (363)	95.5923%	3.8567%	T	95.5923%	3.8567%	T
run-f078 (440)	91.1364%	7.5%	T	91.1364%	7.5%	T
run-f079 (374)	100%	0%	T	100%	0%	T
run-f080 (434)	100%	0%	T	100%	0%	T
run-f085 (158)	98.7342%	0.6329%	T	98.7342%	0.6329%	T
run-f086 (221)	100%	0%	T	100%	0%	T
run-f087 (252)	98.4127%	0.3968%	T	98.4127%	0.3968%	T
run-f088 (250)	92%	6.4%	T	92%	6.4%	T
run-f093 (408)	0.2451%	99.5098%	W	0.9804%	97.7941%	W
run-f094 (396)	0%	100%	W	0%	100%	W
run-f095 (175)	0.5714%	98.8571%	W	0.5714%	98.8571%	W
run-f096 (459)	0.2179%	99.3464%	W	0.2179%	99.3464%	W
run-f101 (132)	3.0303%	93.9394%	W	9.8485%	86.3636%	W
run-f102 (66)	22.7273%	71.2121%	W	22.7273%	74.2424%	W
run-f103 (224)	0.4464%	99.1071%	W	0.4464%	99.1071%	W
run-f104 (270)	6.2963%	88.8889%	W	0.3704%	97.7778%	W
run-f109 (210)	0%	100%	W	0%	100%	W
run-f110 (208)	4.3269%	94.7115%	W	4.3269%	94.7115%	W
run-f111 (170)	0%	99.4118%	W	0%	99.4118%	W
run-f112 (403)	0.2481%	99.5037%	W	0.2481%	99.5037%	W
run-f117 (125)	0.8%	98.4%	W	3.2%	92%	W
run-f118 (83)	0%	100%	W	0%	100%	W
run-f119 (32)	0%	100%	W	0%	100%	W
run-f120 (138)	0%	100%	W	0%	100%	W
run-f173 (426)	84.2723%	13.3803%	T	84.7418%	12.6761%	T
run-f174 (472)	76.2712%	23.5169%	T	81.3559%	18.4322%	T
run-f175 (455)	99.3407%	0.2198%	T	99.3407%	0.2198%	T
run-f176 (501)	73.6527%	26.1477%	T	74.4511%	25.3493%	T
run-f181 (113)	100%	0%	T	100%	0%	T
run-f182 (261)	90.0383%	8.4291%	T	91.5709%	5.7471%	T
run-f183 (263)	91.2548%	7.9848%	T	88.9374%	9.8859%	T
run-f184 (271)	88.9299%	8.8561%	T	88.9299%	8.8561%	T
run-f185 (363)	100%	0%	T	100%	0%	T
run-f186 (308)	86.3636%	13.3117%	T	86.3636%	13.3117%	T
run-f187 (260)	98.4615%	0.7692%	T	98.4615%	0.7692%	T
run-f188 (387)	100%	0%	T	99.4832%	0.2584%	T
run-f189 (483)	83.8509%	15.5280%	T	82.6087%	16.7702%	T
run-f190 (498)	50.6024%	49.1968%	T	50.6024%	49.1968%	T
run-f197 (308)	0%	100%	W	0%	100%	W
run-f198 (231)	33.3333%	66.2338%	T	25.9740%	73.1602%	T
run-f199 (262)	0%	99.6183%	W	0%	100%	W
run-f200 (234)	39.3162%	60.2564%	T	30.7692%	68.8034%	T

Table 5.6: The percentage of prototypes that were classified as a tracked (T), a wheeled (W), or an undetermined (O) vehicle, and the decision for each blind run (the number in the bracket represents the number of non-CPA-based prototypes in that run) for the NAU experiments.

	Type-1			Type-2		
Run	Tracked	Wheeled	Decision	Tracked	Wheeled	Decision
run-f005 (683)	55.9297%	44.0703%	T	46.4129%	53.5871%	W
run-f006 (146)	100%	0%	T	100%	0%	T
run-f007 (356)	100%	0%	T	100%	0%	T
run-f008 (470)	80.2128%	19.7872 %	T	80.2128 %	19.7872%	T
run-f010 (798)	76.9424%	23.0576%	T	76.9424%	23.0576%	T
run-f019 (230)	97.3913%	2.6087%	T	97.3913%	2.6087%	T
run-f020 (234)	67.0940%	32.9060%	T	69.2308%	30.7692%	T
run-f021 (277)	89.5307%	10.4693%	T	88.8087%	11.1913%	T
run-f022 (294)	66.3265%	33.6735%	T	65.9864%	34.0136%	T
run-f077 (363)	93.9394%	6.0606%	T	93.9394%	6.0606%	T
run-f078 (440)	96.8182%	3.1818%	T	96.8182%	3.1818%	T
run-f079 (374)	99.7326%	0.2674%	T	99.7326%	0.2674%	T
run-f080 (434)	100%	0%	T	100%	0%	T
run-f085 (158)	99.3671%	0.6329%	T	99.3671%	0.6329%	T
run-f086 (221)	99.5475%	0.4525%	T	99.5475%	0.4525%	T
run-f087 (252)	98.4127%	1.5873%	T	98.4127%	1.5873%	T
run-f088 (250)	94.4%	5.6%	T	94.4%	5.6%	T
run-f093 (408)	0.7353%	99.2647%	W	1.4706%	98.5294%	W
run-f094 (396)	0%	100%	W	0%	100%	W
run-f095 (175)	0.5714%	99.4286%	W	0.5714%	99.4286%	W
run-f096 (459)	0.4357%	99.5643%	W	0.4357%	99.5643%	W
run-f101 (132)	65.1515%	34.8485%	T	65.1515%	34.8485%	T
run-f102 (66)	56.0606%	43.9394%	T	56.0606%	43.9394%	T
run-f103 (224)	53.5714%	46.4286%	T	53.5714%	46.4286%	T
run-f104 (270)	45.5556%	54.4444%	T	43.3333%	56.6667%	T
run-f109 (210)	0%	100%	W	0%	100%	W
run-f110 (208)	6.7308%	93.2692%	W	4.3269%	95.6731%	W
run-f111 (170)	1.1765%	98.8235%	W	1.1765%	98.8235%	W
run-f112 (403)	0.2481%	99.7519%	W	0.2481%	99.7519%	W
run-f117 (125)	0.8%	99.2%	W	2.4%	97.6%	W
run-f118 (83)	0%	100%	W	0%	100%	W
run-f119 (32)	12.5%	87.5%	W	12.5%	87.5%	W
run-f120 (138)	0%	100%	W	0%	100%	W
run-f173 (426)	88.0282%	11.9718%	T	88.7324%	11.2676%	T
run-f174 (472)	73.0932%	26.9068%	T	76.6949%	23.3051%	T
run-f175 (455)	99.3407%	0.6593%	T	99.3407%	0.6593%	T
run-f176 (501)	75.0499%	24.9501%	T	75.6487%	24.3513%	T
run-f181 (113)	100%	0%	T	97.3451%	2.6549%	T
run-f182 (261)	80.0766%	19.9234%	T	80.0766%	19.9234%	T
run-f183 (263)	94.2966%	5.7034%	T	93.9163%	6.0837%	T
run-f184 (271)	92.6199%	7.3801%	T	92.6199%	7.3801%	T
run-f185 (363)	100%	0%	T	100%	0%	T
run-f186 (308)	88.3117%	11.6883%	T	88.3117%	11.6883%	T
run-f187 (260)	92.3077%	7.6923%	T	90.3046%	9.6154%	T
run-f188 (387)	98.9664%	1.0336%	T	98.1912%	1.8088%	T
run-f189 (483)	91.0973%	8.9027%	T	87.7847%	12.2153%	T
run-f190 (498)	52.0080%	47.9920%	T	52.4096%	47.5904%	T
run-f197 (308)	38.63640%	61.3636%	T	37.3377%	62.6623%	T
run-f198 (231)	45.0216%	54.9784%	T	42.4242%	57.5758%	T
run-f199 (262)	73.2824%	26.7176%	T	58.0153%	41.9847%	T
run-f200 (234)	54.2735%	45.7265%	T	46.1538%	53.8462%	T

Table 5.7: Summary of classification results for each blind run provided by the type-2 FL-RBC cluster, where T represents a tracked vehicle, W represents a wheeled vehicle, and O represents an undetermined vehicle.

Run	CN/CAE	CAU	NAE	NAU	Vote
run-f005	T	T	T	W	T
run-f006	T	T	T	T	T
run-f007	T	T	T	T	T
run-f008	T	T	T	T	T
run-f010	T	T	T	T	T
run-f019	T	T	T	T	T
run-f020	T	T	T	T	T
run-f021	T	T	T	T	T
run-f022	T	T	T	T	T
run-f077	T	T	T	T	T
run-f078	T	T	T	T	T
run-f079	T	T	T	T	T
run-f080	T	T	T	T	T
run-f085	T	T	T	T	T
run-f086	T	T	T	T	T
run-f087	T	T	T	T	T
run-f088	T	T	T	T	T
run-f093	W	W	W	W	W
run-f094	W	W	W	W	W
run-f095	W	W	W	W	W
run-f096	W	W	W	W	W
run-f101	W	W	W	T	W
run-f102	T	T	W	T	T
run-f103	T	W	W	T	O
run-f104	T	T	W	T	T
run-f109	W	W	W	W	W
run-f110	O	W	W	W	W
run-f111	W	W	W	W	W
run-f112	W	W	W	W	W
run-f117	W	W	W	W	W
run-f118	W	W	W	W	W
run-f119	W	T	W	W	W
run-f120	W	W	W	W	W
run-f173	T	T	T	T	T
run-f174	T	T	T	T	T
run-f175	T	T	T	T	T
run-f176	T	T	T	T	T
run-f181	T	T	T	T	T
run-f182	T	T	T	T	T
run-f183	T	T	T	T	T
run-f184	T	T	T	T	T
run-f185	T	T	T	T	T
run-f186	T	T	T	T	T
run-f187	T	T	T	T	T
run-f188	T	T	T	T	T
run-f189	T	T	T	T	T
run-f190	T	T	T	T	T
run-f197	T	T	W	T	T
run-f198	T	T	T	T	T
run-f199	T	T	W	T	T
run-f200	T	T	T	T	T

Table 5.8: Score for the classification results of the type-2 FL-RBC cluster.

Experiment	CN/CAE	CAU	NAE	NAU	Vote
Score	47/50	58/51	49/51	47/51	47/50

Chapter 6

Conclusions

In this report we have summarized our extensive study into binary classification of ground vehicles from acoustic data. We have described every aspect of our study, including: feature extraction, FOUs of the features, classifier construction and experimental evaluation of the classifiers.

Because acoustic data are subject to variations of the environmental conditions, vehicle-traveling speed, and the distance between the vehicle and the sensor system, the statistics of the features (the magnitudes of the second through 12-th harmonic frequencies) for the same kind of vehicle varies from run to run. We therefore used type-2 fuzzy sets to model the uncertainties contained in these features, and then constructed type-2 FL-RBCs for three binary classification problems: tracked vs. wheeled vehicle, heavy-tracked vs. light tracked vehicle, and heavy-wheeled vs. light-wheeled vehicle. To evaluate the type-2 FL-RBC in a fair way, we also constructed a Bayesian classifier and type-1 FL-RBC and compared their performance through many experiments. The parameters of the Bayesian classifier were estimated based on the training prototypes, whereas, the parameters of both the type-1 and type-2 FL-RBCs were optimized using a steepest descent algorithm that minimized an objective function which depended upon the training prototypes. Each classifier had two working modes—non-adaptive and adaptive. We proved that when the non-adaptive FAR of a classifier is less than 0.5, then after a certain time its adaptive performance is always better than its non-adaptive performance, and we established exactly what that time is.

We carried out two groups of experiments to evaluate the performance of all classifiers. In the first group of experiments (leave-one-out), only one run was used as the testing run, and all the other runs were used as the training runs. In the second group of experiments (leave-M-out), one run of each vehicle (M runs where there are M kinds of vehicles) was used for testing and all the other runs were used for training. In both groups of experiments, the CPA-based prototypes of the training runs were used for training. For the *CN-CN* configuration (see Table 4.1 for the meanings of these acronyms), the CPA-based prototypes of the testing runs were used to examine the non-adaptive performance of each classifier; and, for the *CN-NN-NA* configuration, the non-CPA-based prototypes of the testing runs were used to examine both the non-adaptive and adaptive performances of each classifier.

Our experimental results showed that for each binary classification problem, both the type-1 and type-2 FL-RBCs had significantly better performance than the Bayesian classifier, whereas the type-1 and type-2 FL-RBCs had similar performance, although most of time, the type-2 FL-RBC had slightly better performance than the type-1 FL-RBC. Our experimental result also confirmed our theoretical result that the adaptive performance is always better than the non-adaptive performance.

We also carried out blind tests for both the type-1 and type-2 FL-RBC designs using blind runs of the normal terrain. The classification results provided by the type-2 FL-RBC design were scored by our sponsor at the Army Research Laboratory. The scores werve very high, which demonstrates that our type-2 FL-RBC designs for the binary classification of ground vehicles are successful.

In the next phase of our study we shall examine the multiple-category classification problem using FL-RBCs.

Acknowledgment

The effort reported on was sponsored by The Department of Army Research Office, Grant DAAD19-01-1-0666. The content of the information does not necessarily reflect the position or policy of the federal government, and no official endorsement should be inferred.

Appendix A

Demonstration that the Vehicle Features are Stationary

In this appendix, we demonstrate that for wheeled vehicles, the means and standard deviations of the features for 75%-overlapped CPA-based prototypes are approximately the same as those for 50%-overlapped CPA-based prototypes.

Let $\mathbf{b}_i$ $(i = 1, \ldots, 160)$ be the 160 CPA-based prototypes from a wheeled run, which are generated by sliding a rectangular window with 75% of window overlap, and $f(\mathbf{b}_i)$ be one of the features extracted from $\mathbf{b}_i$. The mean of $f(\mathbf{b}_i)$, m, is

$$m = \frac{1}{160}\sum_{i=1}^{160} f(\mathbf{b}_i) = \frac{1}{2}\left(\frac{1}{80}\sum_{j=1}^{80} f(\mathbf{b}_{2j-1}) + \frac{1}{80}\sum_{j=1}^{80} f(\mathbf{b}_{2j})\right) = \frac{1}{2}(m_{odd} + m_{even}) \tag{A-1}$$

where m_{odd} and m_{even} are the means of those $f(\mathbf{b}_i)$ with odd and even indices, respectively. The variance of $f(\mathbf{b}_i)$, σ^2, is

$$\begin{aligned}\sigma^2 &= \frac{1}{159}\sum_{i=1}^{160}[f(\mathbf{b}_i) - m]^2 = \frac{1}{159}\sum_{i=1}^{160}\left[f(\mathbf{b}_i) - \frac{1}{2}(m_{odd} + m_{even})\right]^2 \\ &= \frac{1}{159}\sum_{j=1}^{80}\left\{[f(\mathbf{b}_{2j-1}) - m_{odd}] + \frac{1}{2}(m_{odd} - m_{even})\right\}^2 \\ &+ \frac{1}{159}\sum_{j=1}^{80}\left\{[f(\mathbf{b}_{2j}) - m_{even}] + \frac{1}{2}(m_{even} - m_{odd})\right\}^2\end{aligned} \tag{A-2}$$

If the features that are extracted from the interval about the CPA are stationary in the mean square sense, then it is reasonable to assume that the statistics of $f(\mathbf{b}_i)$ with odd indices are almost the same as those with even indices, i.e., $m_{odd} \approx m_{even}$ and $\sigma^2_{odd} \approx \sigma^2_{even}$, where σ^2_{odd} and σ^2_{even} are defined as

$$\sigma_{odd}^2 \equiv \frac{1}{79}\sum_{j=1}^{80}\left[f\left(\mathbf{b}_{2j-1}\right) - m_{odd}\right]^2 \tag{A-3}$$

$$\sigma_{even}^2 \equiv \frac{1}{79}\sum_{j=1}^{80}\left[f\left(\mathbf{b}_{2j}\right) - m_{even}\right]^2 \tag{A-4}$$

Hence, (A-1) and (A-2) can be rewritten as

$$m = \frac{1}{2}\left(m_{odd} + m_{evem}\right) \approx m_{odd} \approx m_{even} \tag{A-5}$$

and (according to (A-2)-(A-5))

$$\begin{aligned} \sigma^2 &\approx \frac{1}{159}\sum_{j=1}^{80}\left[f(\mathbf{b}_{2j-1}) - m_{odd}\right]^2 + \frac{1}{159}\sum_{j=1}^{80}\left[f(\mathbf{b}_{2j}) - m_{even}\right]^2 \\ &= \frac{79}{159}\sigma_{odd}^2 + \frac{79}{159}\sigma_{even}^2 \approx \frac{158}{159}\sigma_{odd}^2 \approx \frac{158}{159}\sigma_{even}^2 \end{aligned} \tag{A-6}$$

Because the blocks with odd (or even) indices have 50% overlap, m_{odd} and σ_{odd} (m_{even} and σ_{even}) are the statistics for those CPA-based prototypes that are generated by sliding a rectangular window with 50% of window overlap. Hence, (A-5) and (A-6) demonstrate that the means and standard deviations of the features for the 75%-overlapped CPA-based prototypes are approximately the same as those for the 50%-overlapped CPA-based prototypes, if the features extracted from the interval about the CPA are stationary in the mean square sense.

Table A.1 shows the means (m, m_{odd} and m_{even}) and standard deviations (σ, σ_{odd} and σ_{even}) of the features (feature extraction is described in Section 2.3) for one wheeled run (the statistics for the other wheeled runs, although not shown, have similar results), where the CPA-based prototypes are generated by using 75% window overlap. Observe that:

1. $(m_{odd} + m_{even})/2 = m$, which is consistent with our analysis in (A-1).
2. m_{odd} and m_{even} are each close to m, and $\sqrt{158/159}\sigma_{odd}$ and $\sqrt{158/159}\sigma_{even}$ are each close to σ, respectively, consistent with our analyses in (A-5) and (A-6).
3. $\sqrt{79\left(\sigma_{odd}^2 + \sigma_{even}^2\right)/159} \approx \sigma$ which is consistent with our analysis in (A-6).

These results support our assumption about the stationary property of the features for wheeled vehicles.

Table A.1: Means (m, m_{odd} and m_{even}) and standard deviations (σ, σ_{odd} and σ_{even}) of the features for one W-a run, where the CPA-based prototypes are generated by using the 75% window overlap [f_0 represents the fundamental frequency, and x_i ($i = 1, \ldots, 11$) represents the i-th feature].

	f_0	x_1	x_2	x_3	x_4	x_5
m	10.0893	0.4744	0.5360	0.4163	0.8993	0.5512
m_{odd}	10.1054	0.4760	0.5375	0.4139	0.9085	0.5517
m_{even}	10.0732	0.4729	0.5346	0.4186	0.8900	0.5508
$(m_{odd} + m_{even}/2$	10.0893	0.4744	0.5360	0.4163	0.8993	0.5512
σ	0.6439	0.2296	0.1950	0.1995	0.2229	0.2958
$\sqrt{158/159}\sigma_{odd}$	0.6376	0.2227	0.1911	0.1924	0.2117	0.2937
$\sqrt{158/159}\sigma_{even}$	0.6496	0.2364	0.1989	0.2062	0.2332	0.2981
$\sqrt{79\ (\sigma_{odd}^2 + \sigma_{even}^2)/\ 159}$	0.6437	0.2296	0.1950	0.1994	0.2227	0.2958

	x_6	x_7	x_8	x_9	x_{10}	x_{11}
m	0.2203	0.3359	0.1715	0.2820	0.1393	0.1261
m_{odd}	0.2037	0.3413	0.1726	0.2904	0.1399	0.1195
m_{even}	0.2369	0.3305	0.1704	0.2736	0.1386	0.1327
$(m_{odd} + m_{even})/2$	0.2203	0.3359	0.1715	0.2820	0.1393	0.1261
σ	0.2063	0.1747	0.0618	0.1395	0.0544	0.1001
$\sqrt{158/159}\sigma_{odd}$	0.1727	0.1796	0.0645	0.1424	0.0555	0.0914
$\sqrt{158/159}\sigma_{even}$	0.2340	0.1694	0.0590	0.1360	0.0532	0.1077
$\sqrt{79\ (\sigma_{odd}^2 + \sigma_{even}^2)/\ 159}$	0.2056	0.1746	0.0618	0.1393	0.0544	0.0999

Appendix B

Proof of Theorem 1

When our binary classifier operates in its non-adaptive mode, the probability of the decision made by this classifier for an arbitrary prototype $\mathbf{x}$ being incorrect is p, and the probability of this decision being correct is $1 - p$, where p represents the FAR of this classifier in its non-adaptive mode. Let s be the decision for $\mathbf{x}$ made by a classifier in its non-adaptive working mode, where $+1$ represents s being correct, and -1 represents s being incorrect. Then, s can be considered to be a Bernoulli random variable with distribution:

$$Pr\,(s = -1) = p, \quad Pr\,(s = +1) = 1 - p \tag{B-1}$$

Let $\mathbf{x}_1, \mathbf{x}_2, \ldots$ be the prototypes generated from a run, and s_n be the binary decision for $\mathbf{x}_n$ made by a classifier in its non-adaptive working mode. Because $s_1, s_2, \ldots$ are independently made, each of them is a Bernoulli random variable as in (B-1). When the classifier operates in its adaptive mode, the adaptive decision for $\mathbf{x}_n$ depends on the non-adaptive decisions $s_1, \ldots, s_n$ as follows:

- When $\sum_{i=1}^{n} s_i > 0$, the adaptive decision for $\mathbf{x}_n$ is correct
- When $\sum_{i=1}^{n} s_i < 0$, the adaptive decision for $\mathbf{x}_n$ is incorrect
- When $\sum_{i=1}^{n} s_i = 0$, the adaptive decision for $\mathbf{x}_n$ is unspecified (and is also counted as incorrect)

Note that $\sum_{i=1}^{n} s_i$ can range from $-n$ to n.

The procedure that we use to prove Theorem 1 is:

1. Calculate the probability $Pr\,(\sum_{i=1}^{n} s_i = k)$ for general k values, and the probability $Pr\,(\sum_{i=1}^{n} s_i \leq 0)$, i.e., the probability of the adaptive decision for $\mathbf{x}_n$ being incorrect.

2. Prove the probability of the adaptive decision for $\mathbf{x}_{n+1}$ being incorrect is larger than the probability of the adaptive decision for $\mathbf{x}_n$ being incorrect when n is odd; and, the probability of the adaptive decision for $\mathbf{x}_{n+1}$ being incorrect is smaller than the probability of the adaptive decision for $\mathbf{x}_n$ being incorrect when n is even. This means that the probability of the adaptive decision for $\mathbf{x}_n$ being incorrect is an oscillating function of n.

3. Prove the probability of the adaptive decision for $\mathbf{x}_{n+2}$ being incorrect is smaller than the probability of the adaptive decision of $\mathbf{x}_n$ being incorrect for both n odd and n even. This means that the probability that the adaptive decision for $\mathbf{x}_n$ is incorrect has a decreasing tendency.

1. Because $\sum_{i=1}^{n} s_i$ is the sum of n independently and identically distributed Bernoulli random variables, it is a Binomial random variable [1]. Note[1] that $\sum_{i=1}^{n} s_i = k$ implies that $(n+k)/2$ of $\{s_1, \ldots, s_n\}$ are $+1$, and the remaining $(n-k)/2$ of $\{s_1, \ldots, s_n\}$ are -1; hence,

$$Pr\left(\sum_{i=1}^{n} s_i = k\right) = \binom{n}{(n+k)/2} (1-p)^{(n+k)/2} p^{(n-k)/2} \tag{B-2}$$

where k can only take values as $\{-n, -n+2, \ldots, n-2, n\}$ so that both $(n+k)/2$ and $(n-k)/2$ are integers.

For an integer $m \in [-n, n]$, the probability of $\sum_{i=1}^{n} s_i$ being no greater than m is

$$Pr\left(\sum_{i=1}^{n} s_i \leq m\right) = \sum_{k \leq m} Pr\left(\sum_{i=1}^{n} s_i = k\right) = \sum_{k \leq m} \binom{n}{(n+k)/2} (1-p)^{(n+k)/2} p^{(n-k)/2} \tag{B-3}$$

To simplify (B-3), we can let $t = (n+k)/2$, so that $k \leq m$ implies $0 \leq t \leq (n+m)/2$.

Note that when $m = 0$, (B-3) is the probability that the adaptive decision for $\mathbf{x}_n$ is incorrect,

$$Pr\left(\sum_{i=1}^{n} s_i \leq 0\right) = \sum_{k \leq 0} Pr\left(\sum_{i=1}^{n} s_i = k\right) = \sum_{k \leq 0} \binom{n}{(n+k)/2} (1-p)^{(n+k)/2} p^{(n-k)/2} \tag{B-4}$$

Next, we consider (B-4) for n odd and n even. When $m = 0$ and n is odd, $k \leq 0$ means that t $(0 \leq t \leq n/2)$ takes values as $\{0, 1, \ldots, (n-3)/2, (n-1)/2\}$; hence, (B-4) becomes

$$Pr\left(\sum_{i=1}^{n} s_i \leq 0\right) = Pr\left(\sum_{i=1}^{n} s_i \leq -1\right) = \sum_{t=0}^{(n-1)/2} \binom{n}{t} (1-p)^t p^{n-t}, \quad n \ odd \tag{B-5}$$

---

[1]Suppose k_1 of $\{s_1, \ldots, s_n\}$ are $+1$ and k_2 of $\{s_1, \ldots, s_n\}$ are -1. Then $k_1 - k_2 = k$ and $k_1 + k_2 = n$ simultaneously hold, from which we obtain that $k_1 = (n+k)/2$ and $k_2 = (n-k)/2$. Because both k_1 and k_2 must be integers, both $n + k = 2k_1$ and $n - k = 2k_2$ must be even; hence, k is even when n is even, and k is odd when n is odd.

When $m = 0$ and n is even, $k \leq 0$ means that t $(0 \leq t \leq n/2)$ takes values as $\{0, 1, \ldots, (n-2)/2, n/2\}$; hence, (B-4) becomes

$$Pr\left(\sum_{i=1}^{n} s_i \leq 0\right) = \sum_{t=0}^{n/2} \binom{n}{t} (1-p)^t p^{n-t}, \quad n \ \ even \tag{B-6}$$

Figure B.1 shows $Pr\left(\sum_{i=1}^{n} s_i \leq 0\right)$ as a function of n for different values of $p < 0.5$. Each curve corresponds to one value of p, and is obtained by using (B-5) and (B-6) alternatively for odd and even values of n. Observe from Figure B.1 that each curve is oscillating with a decreasing tendency, and the decreasing slope is steeper for smaller p. Hence, it is possible that the classifier achieves better performance than its non-adaptive mode by just using a simple adaptive voting strategy, and the performance improvement is more dominant as the non-adaptive FAR of the classifier becomes smaller.

2. When n is odd, the probability of the adaptive decision for $\mathbf{x}_n$ being incorrect is given by (B-5). For $\mathbf{x}_{n+1}$, if its non-adaptive decision $s_{n+1} = 1$, then $\sum_{i=1}^{n} s_i$ (which must be odd) should be no greater than -1 to guarantee $\sum_{i=1}^{n+1} s_i \leq 0$. On the other hand, if $s_{n+1} = -1$, then $\sum_{i=1}^{n} s_i$ should be no greater than 1 to guarantee $\sum_{i=1}^{n+1} s_i \leq 0$. Consequently, the probability of the adaptive decision for $\mathbf{x}_{n+1}$ being incorrect, $Pr\left(\sum_{i=1}^{n+1} s_i \leq 0\right)$, can be expressed as a conditional probability, i.e.

$$\begin{aligned} & Pr\left(\sum_{i=1}^{n+1} s_i \leq 0\right) \qquad \text{(B-7)} \\ = \ & Pr\left(\sum_{i=1}^{n} s_i \leq -1 \,\middle|\, s_{n+1} = 1\right) Pr\left(s_{n+1} = 1\right) + Pr\left(\sum_{i=1}^{n} s_i \leq 1 \,\middle|\, s_{n+1} = -1\right) Pr\left(s_{n+1} = -1\right) \end{aligned}$$

Because the non-adaptive decisions, $s_1, \ldots, s_{n+1}$, are independent of each other,

$$Pr\left(\sum_{i=1}^{n} s_i \,\middle|\, s_{n+1}\right) = Pr\left(\sum_{i=1}^{n} s_i\right)$$

and (B-7) becomes

$$\begin{aligned} Pr\left(\sum_{i=1}^{n+1} s_i \leq 0\right) &= Pr\left(\sum_{i=1}^{n} s_i \leq -1\right) Pr\left(s_{n+1} = 1\right) + Pr\left(\sum_{i=1}^{n} s_i \leq 1\right) Pr\left(s_{n+1} = -1\right) \\ &= (1-p) Pr\left(\sum_{i=1}^{n} s_i \leq -1\right) + pPr\left(\sum_{i=1}^{n} s_i \leq 1\right) \qquad \text{(B-8)} \end{aligned}$$

Upon the substitution of (B-3) with $m = -1$ or 1 into (B-8), and letting $t = (n+k)/2$ [hence, $k \leq m$ implies that $0 \leq t \leq (n+m)/2$], we obtain

$$Pr\left(\sum_{i=1}^{n+1} s_i \leq 0\right) = (1-p) \sum_{t=0}^{(n-1)/2} \binom{n}{t} (1-p)^t p^{n-t} + p \sum_{t=0}^{(n+1)/2} \binom{n}{t} (1-p)^t p^{n-t} \tag{B-9}$$

The difference between (B-9) and (B-5) is:

$$
\begin{aligned}
& Pr\left(\sum_{i=1}^{n+1} s_i \le 0\right) - Pr\left(\sum_{i=1}^{n} s_i \le 0\right) \\
&= (1-p)\sum_{t=0}^{(n-1)/2}\binom{n}{t}(1-p)^t p^{n-t} + p\sum_{t=0}^{(n+1)/2}\binom{n}{t}(1-p)^t p^{n-t} - \sum_{t=0}^{(n-1)/2}\binom{n}{t}(1-p)^t p^{n-t} \\
&= -p\sum_{t=0}^{(n-1)/2}\binom{n}{t}(1-p)^t p^{n-t} + p\sum_{t=0}^{(n+1)/2}\binom{n}{t}(1-p)^t p^{n-t} \\
&= p\binom{n}{(n+1)/2}(1-p)^{(n+1)/2}p^{(n-1)/2} > 0 \qquad \text{(B-10)}
\end{aligned}
$$

This means that the probability that the adaptive decision for $\mathbf{x}_{n+1}$ is incorrect is larger than the probability that the adaptive decision for $\mathbf{x}_n$ is incorrect when n is odd.

When n is even, the probability of the adaptive decision for $\mathbf{x}_n$ being incorrect is given by (B-6). For $\mathbf{x}_{n+1}$, if $s_{n+1} = 1$, then $\sum_{i=1}^{n} s_i$ (which must be even) should be no greater than -2 to guarantee $\sum_{i=1}^{n+1} s_i \le 0$. On the other hand, if $s_{n+1} = -1$, then $\sum_{i=1}^{n} s_i$ should be no greater than 0 to guarantee $\sum_{i=1}^{n+1} s_i \le 0$. Consequently, the probability that the adaptive decision for $\mathbf{x}_{n+1}$ is incorrect, $Pr\left(\sum_{i=1}^{n+1} s_i \le 0\right)$, can be expressed as a conditional probability, i.e.

$$
\begin{aligned}
& Pr\left(\sum_{i=1}^{n+1} s_i \le 0\right) \\
&= Pr\left(\sum_{i=1}^{n} s_i \le -2 \,\middle|\, s_{n+1} = 1\right) Pr\left(s_{n+1} = 1\right) + Pr\left(\sum_{i=1}^{n} s_i \le 0 \,\middle|\, s_{n+1} = -1\right) Pr\left(s_{n+1} = -1\right) \\
&= Pr\left(\sum_{i=1}^{n} s_i \le -2\right) Pr\left(s_{n+1} = 1\right) + Pr\left(\sum_{i=1}^{n} s_i \le 0\right) Pr\left(s_{n+1} = -1\right) \\
&= (1-p)Pr\left(\sum_{i=1}^{n} s_i \le -2\right) + pPr\left(\sum_{i=1}^{n} s_i \le 0\right) \qquad \text{(B-11)}
\end{aligned}
$$

Upon the substitution of (B-3) with $m = -2$ or 0 into (B-11), and again letting $t = (n+k)/2$ [hence, $k \le m$ implies $0 \le t \le (n+m)/2$], we obtain

$$
Pr\left(\sum_{i=1}^{n+1} s_i \le 0\right) = (1-p)\sum_{t=0}^{(n-2)/2}\binom{n}{t}(1-p)^t p^{n-t} + p\sum_{t=0}^{n/2}\binom{n}{t}(1-p)^t p^{n-t} \qquad \text{(B-12)}
$$

The difference between (B-12) and (B-6) is:

$$Pr\left(\sum_{i=1}^{n+1} s_i \leq 0\right) - Pr\left(\sum_{i=1}^{n} s_i \leq 0\right)$$

$$= (1-p)\sum_{t=0}^{(n-2)/2}\begin{pmatrix} n \\ t \end{pmatrix}(1-p)^t p^{n-t} + p\sum_{t=0}^{n/2}\begin{pmatrix} n \\ t \end{pmatrix}(1-p)^t p^{n-t} - \sum_{t=0}^{n/2}\begin{pmatrix} n \\ t \end{pmatrix}(1-p)^t p^{n-t}$$

$$= (1-p)\sum_{t=0}^{(n-2)/2}\begin{pmatrix} n \\ t \end{pmatrix}(1-p)^t p^{n-t} - (1-p)\sum_{t=0}^{n/2}\begin{pmatrix} n \\ t \end{pmatrix}(1-p)^t p^{n-t}$$

$$= -(1-p)\begin{pmatrix} n \\ n/2 \end{pmatrix}(1-p)^{n/2} p^{n/2} < 0 \qquad \text{(B-13)}$$

This means that the probability that the adaptive decision for $\mathbf{x}_{n+1}$ is incorrect is smaller than the probability that the adaptive decision for $\mathbf{x}_n$ is incorrect when n is even.

3. When n is odd, if $s_{n+1} + s_{n+2} = 2$ (corresponding to $s_{n+1} = 1$ and $s_{n+2} = 1$), then $\sum_{i=1}^{n} s_i$ (which must be odd) should be no greater than -3 to guarantee $\sum_{i=1}^{n+2} s_i \leq 0$; if $s_{n+1} + s_{n+2} = 0$ (corresponding to $s_{n+1} = 1$ and $s_{n+2} = -1$, or $s_{n+1} = -1$ and $s_{n+2} = 1$), then $\sum_{i=1}^{n} s_i$ should be no greater than -1 to guarantee $\sum_{i=1}^{n+2} s_i \leq 0$; and, if $s_{n+1} + s_{n+2} = -2$ (corresponding to $s_{n+1} = -1$ and $s_{n+2} = -1$), then $\sum_{i=1}^{n} s_i$ should be no greater than 1 to guarantee $\sum_{i=1}^{n+2} s_i \leq 0$. Consequently, the probability that the adaptive decision for $\mathbf{x}_{n+2}$ is incorrect, $Pr\left(\sum_{i=1}^{n+2} s_i \leq 0\right)$, can be expressed as a conditional probability, i.e.

$$Pr\left(\sum_{i=1}^{n+2} s_i \leq 0\right)$$

$$= Pr\left(\sum_{i=1}^{n} s_i \leq -3 \,\middle|\, s_{n+1} + s_{n+2} = 2\right) Pr\left(s_{n+1} + s_{n+2} = 2\right)$$

$$+ Pr\left(\sum_{i=1}^{n} s_i \leq -1 \,\middle|\, s_{n+1} + s_{n+2} = 0\right) Pr\left(s_{n+1} + s_{n+2} = 0\right)$$

$$+ Pr\left(\sum_{i=1}^{n} s_i \leq 1 \,\middle|\, s_{n+1} + s_{n+2} = -2\right) Pr\left(s_{n+1} + s_{n+2} = -2\right)$$

$$= Pr\left(\sum_{i=1}^{n} s_i \leq -3\right) Pr\left(s_{n+1} + s_{n+2} = 2\right) + Pr\left(\sum_{i=1}^{n} s_i \leq -1\right) Pr\left(s_{n+1} + s_{n+2} = 0\right)$$

$$+ Pr\left(\sum_{i=1}^{n} s_i \leq 1\right) Pr\left(s_{n+1} + s_{n+2} = -2\right)$$

$$= Pr\left(s_{n+1} = 1\right) Pr\left(s_{n+2} = 1\right) Pr\left(\sum_{i=1}^{n} s_i \leq -3\right)$$

$$+ \left[Pr\left(s_{n+1} = 1\right) Pr\left(s_{n+2} = -1\right) + Pr\left(s_{n+1} = -1\right) Pr\left(s_{n+2} = 1\right)\right] Pr\left(\sum_{i=1}^{n} s_i \leq -1\right)$$

$$+ \ Pr\left(s_{n+1}=-1\right) Pr\left(s_{n+2}=-1\right) Pr\left(\sum_{i=1}^{n} s_i \le 1\right)$$

$$= (1-p)^2 Pr\left(\sum_{i=1}^{n} s_i \le -3\right) + 2p(1-p) Pr\left(\sum_{i=1}^{n} s_i \le -1\right) + p^2 Pr\left(\sum_{i=1}^{n} s_i \le 1\right) \qquad \text{(B-14)}$$

Upon the substitution of (B-3) with $m = -3$, -1 or 1 into (B-14), and letting $t = (n+k)/2$ [hence, $k \le m$ implies $0 \le t \le (n+m)/2$], we obtain

$$Pr\left(\sum_{i=1}^{n+2} s_i \le 0\right) = (1-p)^2 \sum_{t=0}^{(n-3)/2} \binom{n}{t} (1-p)^t p^{n-t} + 2p(1-p) \sum_{t=0}^{(n-1)/2} \binom{n}{t} (1-p)^t p^{n-t}$$

$$+ \ p^2 \sum_{t=0}^{(n+1)/2} \binom{n}{t} (1-p)^t p^{n-t} \qquad \text{(B-15)}$$

The difference between (B-15) and (B-5) is:

$$Pr\left(\sum_{i=1}^{n+2} s_i \le 0\right) - Pr\left(\sum_{i=1}^{n} s_i \le 0\right)$$

$$= (1-p)^2 \sum_{t=0}^{(n-3)/2} \binom{n}{t} (1-p)^t p^{n-t} + 2p(1-p) \sum_{t=0}^{(n-1)/2} \binom{n}{t} (1-p)^t p^{n-t}$$

$$+ \ p^2 \sum_{t=0}^{(n+1)/2} \binom{n}{t} (1-p)^t p^{n-t} - \sum_{t=0}^{(n-1)/2} \binom{n}{t} (1-p)^t p^{n-t}$$

$$= (1-p)^2 \sum_{t=0}^{(n-3)/2} \binom{n}{t} (1-p)^t p^{n-t} + (2p-2p^2-1) \sum_{t=0}^{(n-1)/2} \binom{n}{t} (1-p)^t p^{n-t}$$

$$+ \ p^2 \sum_{t=0}^{(n+1)/2} \binom{n}{t} (1-p)^t p^{n-t}$$

$$= (1-p)^2 \sum_{t=0}^{(n-3)/2} \binom{n}{t} (1-p)^t p^{n-t} - \left[(1-p)^2 + p^2\right] \sum_{t=0}^{(n-1)/2} \binom{n}{t} (1-p)^t p^{n-t}$$

$$+ \ p^2 \sum_{t=0}^{(n+1)/2} \binom{n}{t} (1-p)^t p^{n-t}$$

$$= -(1-p)^2 \binom{n}{(n-1)/2} (1-p)^{(n-1)/2} p^{(n+1)/2} + p^2 \binom{n}{(n+1)/2} (1-p)^{(n+1)/2} p^{(n-1)/2}$$

$$= -(1-p) \binom{n}{(n-1)/2} (1-p)^{(n+1)/2} p^{(n+1)/2} + p \binom{n}{(n+1)/2} (1-p)^{(n+1)/2} p^{(n+1)/2}$$

$$= (2p-1)\begin{pmatrix} n \\ (n-1)/2 \end{pmatrix}(1-p)^{(n+1)/2}p^{(n+1)/2} \tag{B-16}$$

In the above derivation, we have used the fact

$$\begin{pmatrix} n \\ (n-1)/2 \end{pmatrix} = \begin{pmatrix} n \\ (n+1)/2 \end{pmatrix}$$

Note that (B-16) is negative when $p < 0.5$. This means that the probability of the adaptive decision for $\mathbf{x}_{n+2}$ being incorrect is smaller than the probability of the adaptive decision for $\mathbf{x}_n$ being incorrect when n is odd.

When n is even, if $s_{n+1} + s_{n+2} = 2$ (corresponding to $s_{n+1} = 1$ and $s_{n+2} = 1$), then $\sum_{i=1}^{n} s_i$ (which must be even) should be no greater than -2 to guarantee $\sum_{i=1}^{n+2} s_i \le 0$; if $s_{n+1} + s_{n+2} = 0$ (corresponding to $s_{n+1} = 1$ and $s_{n+2} = -1$, or $s_{n+1} = -1$ and $s_{n+2} = 1$), then $\sum_{i=1}^{n} s_i$ should be no greater than 0 to guarantee $\sum_{i=1}^{n+2} s_i \le 0$; and, if $s_{n+1} + s_{n+2} = -2$ (corresponding to $s_{n+1} = -1$ and $s_{n+2} = -1$), then $\sum_{i=1}^{n} s_i$ should be no greater than 2 to guarantee $\sum_{i=1}^{n+2} s_i \le 0$. Consequently, the probability that the adaptive decision for $\mathbf{x}_{n+2}$ is incorrect, $Pr\left(\sum_{i=1}^{n+2} s_i \le 0\right)$, can be expressed as a conditional probability, i.e.

$$\begin{aligned}
&Pr\left(\sum_{i=1}^{n+2} s_i \le 0\right) \\
=\;& Pr\left(\sum_{i=1}^{n} s_i \le -2 \,\middle|\, s_{n+1} + s_{n+2} = 2\right) Pr\left(s_{n+1} + s_{n+2} = 2\right) \\
+\;& Pr\left(\sum_{i=1}^{n} s_i \le 0 \,\middle|\, s_{n+1} + s_{n+2} = 0\right) Pr\left(s_{n+1} + s_{n+2} = 0\right) \\
+\;& Pr\left(\sum_{i=1}^{n} s_i \le 2 \,\middle|\, s_{n+1} + s_{n+2} = -2\right) Pr\left(s_{n+1} + s_{n+2} = 2\right) \\
=\;& Pr\left(\sum_{i=1}^{n} s_i \le -2\right) Pr\left(s_{n+1} + s_{n+2} = 2\right) + Pr\left(\sum_{i=1}^{n} s_i \le 0\right) Pr\left(s_{n+1} + s_{n+2} = 0\right) \\
+\;& Pr\left(\sum_{i=1}^{n} s_i \le 2\right) Pr\left(s_{n+1} + s_{n+2} = 2\right) \\
=\;& Pr\left(s_{n+1} = 1\right) Pr\left(s_{n+2} = 1\right) Pr\left(\sum_{i=1}^{n} s_i \le -2\right) \\
+\;& \left[Pr\left(s_{n+1} = 1\right) Pr\left(s_{n+2} = -1\right) + Pr\left(s_{n+1} = -1\right) Pr\left(s_{n+2} = 1\right)\right] Pr\left(\sum_{i=1}^{n} s_i \le 0\right) \\
+\;& Pr\left(s_{n+1} = -1\right) Pr\left(s_{n+2} = -1\right) Pr\left(\sum_{i=1}^{n} s_i \le 2\right)
\end{aligned}$$

$$= (1-p)^2 Pr\left(\sum_{i=1}^{n} s_i \le -2\right) + 2p(1-p)Pr\left(\sum_{i=1}^{n} s_i \le 0\right) + p^2 Pr\left(\sum_{i=1}^{n} s_i \le 2\right) \quad \text{(B-17)}$$

Upon the substitution of (B-3) with $m = -2$, 0, or 2 into (B-17), and letting $t = (n+k)/2$ [hence, $k \le m$ implies $0 \le t \le (n+m)/2$], we obtain:

$$Pr\left(\sum_{i=1}^{n+2} s_i \le 0\right) = (1-p)^2 \sum_{t=0}^{(n-2)/2} \binom{n}{t} (1-p)^t p^{n-t} + 2p(1-p) \sum_{t=0}^{n/2} \binom{n}{t} (1-p)^t p^{n-t}$$
$$+ p^2 \sum_{t=0}^{(n+2)/2} \binom{n}{t} (1-p)^t p^{n-t} \quad \text{(B-18)}$$

The difference between (B-18) and (B-6) is:

$$Pr\left(\sum_{i=1}^{n+2} s_i \le 0\right) - Pr\left(\sum_{i=1}^{n} s_i \le 0\right)$$
$$= (1-p)^2 \sum_{t=0}^{(n-2)/2} \binom{n}{t} (1-p)^t p^{n-t} + 2p(1-p) \sum_{t=0}^{n/2} \binom{n}{t} (1-p)^t p^{n-t}$$
$$+ p^2 \sum_{t=0}^{(n+2)/2} \binom{n}{t} (1-p)^t p^{n-t} - \sum_{t=0}^{n/2} \binom{n}{t} (1-p)^t p^{n-t}$$
$$= (1-p)^2 \sum_{t=0}^{(n-2)/2} \binom{n}{t} (1-p)^t p^{n-t} + (2p - 2p^2 - 1) \sum_{t=0}^{n/2} \binom{n}{t} (1-p)^t p^{n-t}$$
$$+ p^2 \sum_{t=0}^{(n+2)/2} \binom{n}{t} (1-p)^t p^{n-t}$$
$$= (1-p)^2 \sum_{t=0}^{(n-2)/2} \binom{n}{t} (1-p)^t p^{n-t} - \left[(1-p)^2 + p^2\right] \sum_{t=0}^{n/2} \binom{n}{t} (1-p)^t p^{n-t}$$
$$+ p^2 \sum_{t=0}^{(n+2)/2} \binom{n}{t} (1-p)^t p^{n-t}$$
$$= -(1-p)^2 \binom{n}{n/2} (1-p)^{n/2} p^{n/2} + p^2 \binom{n}{(n+2)/2} (1-p)^{(n+2)/2} p^{(n-2)/2}$$
$$= -\binom{n}{n/2} (1-p)^{n/2+2} p^{n/2} + \binom{n}{(n+2)/2} (1-p)^{(n+2)/2} p^{(n+2)/2}$$
$$= \frac{n!}{(n/2)!(n/2-1)!} (1-p)^{n/2+1} p^{n/2} \left[-\frac{1-p}{n/2} + \frac{p}{n/2+1}\right] \quad \text{(B-19)}$$

Observe that (B-19) is negative when $p < 0.5$, because $(1-p) > p$ and $n/2 < n/2 + 1$. This means that the probability that the adaptive decision for $\mathbf{x}_{n+2}$ is incorrect is smaller than the probability that the adaptive decision for $\mathbf{x}_n$ is incorrect when n is even.

According to the above proof, the probability of the adaptive decision for $\mathbf{x}_n$ being incorrect has a decreasing tendency, and when n is even, the probability of the adaptive decision for $\mathbf{x}_{n+1}$ or $\mathbf{x}_{n+2}$ being incorrect is smaller than the probability of the adaptive decision for $\mathbf{x}_n$ being incorrect. We can therefore always find an even value of $n_0(p)$ for each $p < 0.5$, such that the probability of the adaptive decision for $\mathbf{x}_n$ being incorrect is smaller than p when $n \geq n_0(p)$. For each $p < 0.5$, we search for n_0 by calculating $Pr\left(\sum_{i=1}^{n} s_i \leq 0\right)$ for even values of n [using (B-6)] and comparing it with p. Figure B.2 shows $n_0(p)$ as function of $p < 0.5$. Observe that $n_0(p)$ is non-decreasing with respect to p. This means that when the non-adaptive performance of a classifier is worse (i.e. p is larger) it takes more time [i.e., a larger $n_0(p)$] to guarantee that the adaptive performance of this classifier is better than its non-adaptive performance.

We conclude that when the FAR of a classifier for its non-adaptive mode is less than 0.5, then its performance for the adaptive mode should be better than its non-adaptive mode after a certain time $n_0(p)$, which can be determined from Figure B.2.

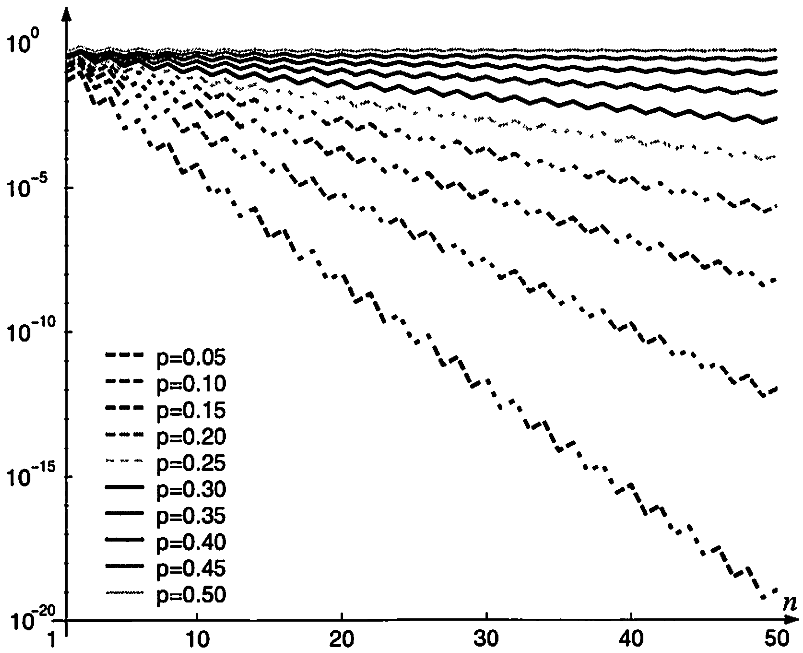


Figure B.1: The probability that the adaptive decision for a prototype is incorrect as a function of its index n and for different values of p, which is the FAR of the classifier for its non-adaptive mode.

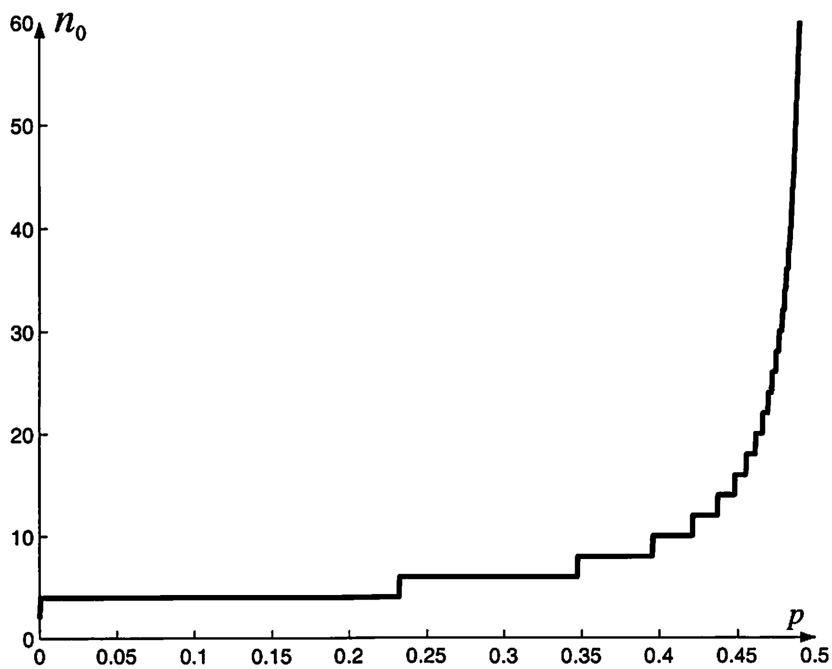


Figure B.2: $n_0(p)$ as function of $p < 0.5$. The probability that the adaptive decision of the n-th prototype is incorrect is always smaller than that of its non-adaptive decision as long as $n \geq n_0(p)$.

Bibliography

[1] G. Casella and R. L. Berger, *Statistical Inference, Second Edition,* Duxbury Press, Pacific Grove, CA, 2002.

[2] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification,* John Wiley & Sons, Inc., New York, NJ, 2001.

[3] D. L. Hall and J. LLinas, "An Introduction to Multisensor Data Fusion," *Proceedings of the IEEE,* pp. 6-23, vol. 85, no.1, January 1997.

[4] D. Lake, "Efficient Maximum Likelihood Estimate for Multiple and Coupled Harmonics," Army Research Laboratory, ARL-TR-2014, December, 1999.

[5] J. M. Mendel *Uncertainty Rule-Based Fuzzy Logic Systems: Introduction and New Directions,* Prentice Hall, Upper Saddle River, NJ, 2001.

[6] D. N. Politis, "Computer-Intensive Methods in Statistical Analysis," *IEEE Signal Processing Magazine,* vol. 15, No. 1, pp. 39-55, Jan. 1998

[7] J. A. Robertson and B. Weber, "Artificial Neural Networks for Acoustic Target Recognition," Joint Report between ARL and ITT Research, 1993.

[8] N. Srour and J. Robertson, "Remote Netted Acoustic Detection System: Final Report," Army Research Laboratory, ARL-TR-706, May, 1995.

[9] M. C. Wellman, N. Srour, and D. B. Hills, "Acoustic Feature Extraction for a Neural Network Classifier," Army Research Laboratory, ARL-TR-1166, January, 1997.

[10] H. Wu and J. M. Mendel, "Data Analysis and Feature Extraction for Ground Vehicle Identification Using Acoustic Data," *Proceedings of 2001 MSS Specialty Group Meeting on Battlefield*

Acoustic and Seismic Sensing, Magnetic and Electric Field Sensors, Applied Physics Lab/Johns Hopkins University, Laurel, MD, October, 2001.

[11] H. Wu and J. M. Mendel, "Classification of Ground Vehicles from Acoustic Data Using Fuzzy Logic Rule-Based Classifiers: Early Results," *Unattended Ground Sensor Technologies and Applications IV, SPIE AeroSense 2002,* Orlando, FL, 2002.

[12] A. M. Zoubir and B. Boashash "The Bootstrap and its Application in Signal Processing," *IEEE Signal Processing Magazine,* vol. 15, No. 1, pp. 56-76, Jan. 1998