ISSN: 2994-6433

Research Article

Journal of Sensor Networks and Data Communications

Distributed Target Recognition with Correlated Features

P. William Kelsey*

Valdosta, GA, United States of America

*Corresponding Author
P. William Kelsey, Valdosta, GA, United States of America.

Submitted: 2024, Dec 06; **Accepted:** 2025, Jan 03; **Published:** 2025, Jan 07

Citation: Kelsey, P. W. (2025). Distributed Target Recognition with Correlated Features. *J Sen Net Data Comm, 5*(1), 01-14.

Abstract

This short paper describes the distributed recognition of targets. The sensor data fusion of features arising from multiple sensors is considered for the purpose of target recognition/classification. This is performed in a scenario wherein the underlying distributions are not Gaussian (i.e., the distributions do not obey Normality). Furthermore, there is 'correlation' between the separate sensor features. The separate (sensor) features are not statistically independent. The data fusion procedure pursued here does not find itself in the object identification sensor data fusion paradigm. It is an intermediate step between the two levels of data fusion for target recognition. In a departure from the (class) conditional assumptions typically made, another factorization of the joint conditional distribution is evaluated. This factorization requires the conditioning on previous feature vectors. A novel adaptive procedure is suggested to address that alternate factorization. A non-standard nonparametric classification procedure is detailed in providing the classification results. The classification/recognition results are for multiple classes. Results are compared against the centralized method and the statistically independent method.

Keywords: Classification, Sensor Data Fusion, Target Recognition

1. Introduction

The field of Automatic Target Recognition (ATR) in real world applications is an exercise in navigating rough terrain. Even in the instance of utilizing one sensor, the underlying distributions of the features (a feature vector) exhibit a strong departure from known parametric forms. I.e., the distributions are not Gaussian (they do not obey Normality) nor do they possess any other known parametric forms. Indeed, the underlying distributions (of the sensor feature vector) can exhibit multi-modality. This, by itself, can cause havoc on single sensor classification systems.

In the instance that multiple sensors are employed to classify (or recognize, the terms are synonymous) a target, yet another problem occurs. The features from the separate sensors can exhibit 'correlation'. In the context of this application the 'correlation' is the (class) conditional dependence of the features across the multiple sensors. This is apparently an issue of some angst. Certainly, it can be said that correlation among separate sensors is not a good thing in general. Procedures deployed to remove the 'correlation' include editing or principal components analysis to arrive at a refurbished set of features.

Further complicating the problem at hand is the development of a distributed procedure to provide a fused target declaration from underlying distributions that are ill-behaved.

The field of sensor data fusion most likely had its nascence from the seminal efforts of [1-4]. The initial application was in the field of multi-sensor detection fusion. Sensor data fusion then grew into the areas of (multiple) target tracking and target recognition.

The field of ATR (and classification using statistical pattern recognition techniques) is similar to target detection in that it can be cast as a problem in hypothesis testing. Indeed, viewed in this way, ATR is nothing more than a generalization of detection. Instead of a binary target/no target decision, the classification procedure provides an answer for an M-ary hypothesis testing problem, where M is the number of classes of interest. Hence, many of the earlier efforts can be brought to bear for this problem.

This fact was recognized in perhaps the 1st paper representing this field of investigation. In [5], a structural procedure is presented to integrate the classification responses from multiple sensors and provide a final (consolidated) classification response.

In [6], a diagram is provided (Figure 12) that lists the data fusion techniques of ATR/classification. The 1st level is termed "data level fusion" which combines the 'data' from multiple sensors into a product from which a multisensor feature vector can be constructed for classification. The research in this area is apparently absent.

The 2nd level is termed “feature level fusion” wherein the separate sensor feature vectors are obtained and placed into a stacked vector for subsequent classification. Research focus in this area also seems to absent.

The 3rd level is termed “decision (or ID) fusion”. This involves the combination of the separate sensor decisions into a single final decision.

The 3rd level has been of interest since that point with active research commencing in [5] and continuing to recent activity [7-9]. The area has been so active that there is not available space to properly cite the efforts in this area. Interestingly, in [7,9], the development tackled the problem of ‘correlation’ albeit at the (classification) decision level. In [7], there is a notable inclusion of a Probability space which is a 3-tuple space of (S,B,P), where B is the Borel σ- field for the sample space considered.

What is curious is that the three levels presented in [6] do not consider the fusion of intermediate information. That information would be the separate sensor likelihoods obtained from each sensor feature vector. The usage of likelihoods (or their posterior counterparts) is rather fundamental to both detection and statistical pattern recognition procedures.

Procedures developed along these ‘intermediate’ lines appear to be scant as well. In [10-12], these procedures are pursued wherein the posterior is constructed from the separate sensor likelihoods for all the classes. For the lack of a better term, procedures along these lines shall be termed “likelihood fusion”.

What is an underlying theme in almost all the efforts is the assumption that the feature vectors are (class) conditionally independent (or that the classification decisions also exhibit conditional independence). For the efforts at the 3rd level, the separate sensor decisions are deemed to be class conditionally independent, allowing factorization of either likelihood ratio tests or separate sensor performance probabilities. In the so-called ‘intermediate’ efforts, the assumption allows an immediate factorization of the joint conditional likelihood.

In this note, the investigation of the ‘intermediate level’ of sensor data fusion for classification is pursued. The procedure must deal with general nonparametric statistical distributions. It must also accommodate a relaxation of the assumption of conditional independence.

In Section II, a short segue is presented regarding the subject of ‘correlation’. In Section III, the design is discussed. In Section IV, results are presented for four cases of interest. Section V contains some concluding remarks.

2. A Note on ‘Correlation’

Clearly there is a concern of ‘correlation’ among either the separate sensor feature vectors (at the “intermediate level”) or the sensor classifications (“ID fusion” level).

Two simple examples are pursued that expose questions that continue to plague ATR procedures.

The first example involves two sensors that have scalar features. The plot of the feature scatter is depicted in Figure 1. As can be seen visually, the features are highly correlated. Since the features are drawn from Normal populations the features are also conditionally dependent. What can also be seen is that while the populations are somewhat close to each other, they do seem to display separation.

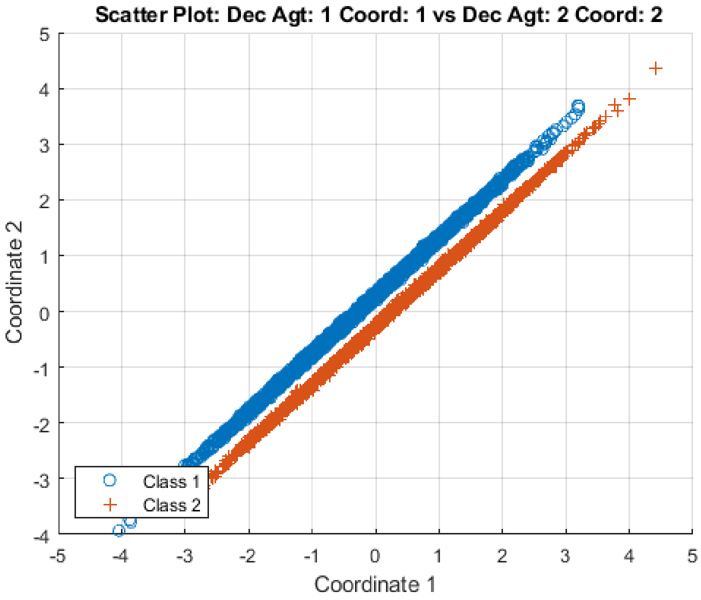

Figure 1: Feature Scatter Diagram for Example 1

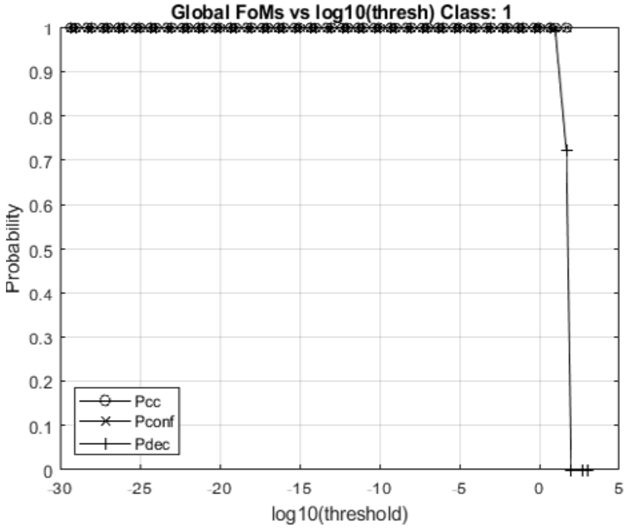

Figure 2: Classifier Performance Class #1for Example #1

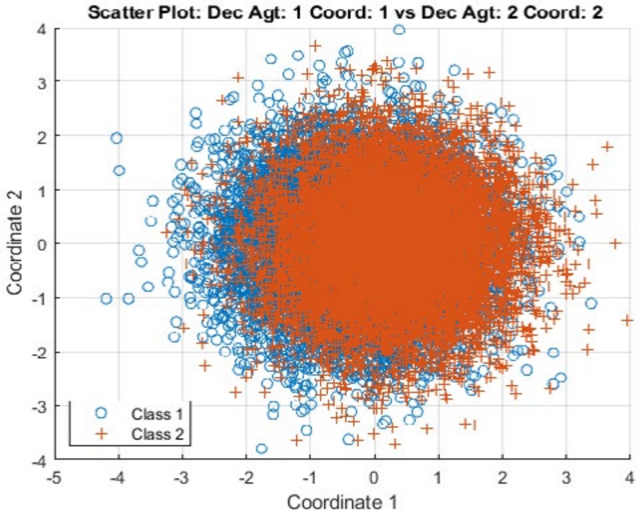

Figure 3: Feature Scatter Diagram for Example 2

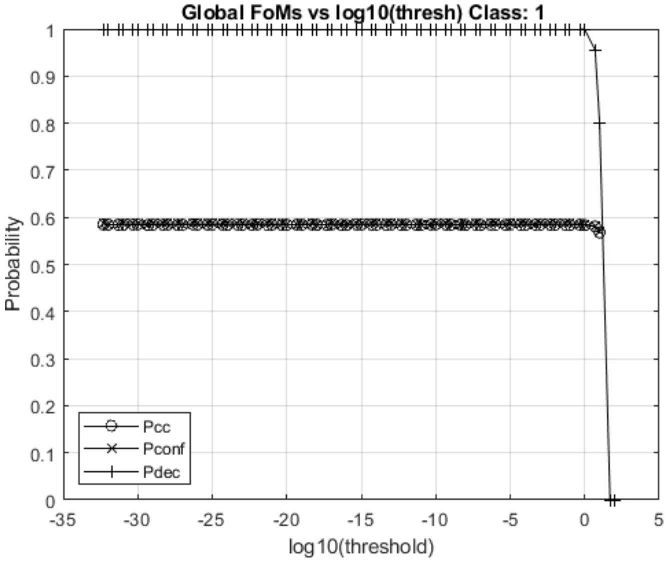

Figure 4: Classifier Performance Class #1 for Example #2

The classification performance is shown in Figure 2 for class #1 (class #2 was the same) where both features are used. The performance is practically perfect confirming the separability (the traces are explained later in Section IV).

But these features are highly correlated and conditionally dependent. A situation like this is to be avoided if at all possible.

Another example is shown in Figure 3. The populations are also drawn from Normal distributions and are conditionally independent. Conditionally independent ('uncorrelated") features are typically desirable. The scatter diagram shows the two populations and it shows a high amount of overlap between the two classes. The classification performance is shown in Figure 4 (for class #1, class #2 was similar). Since this is a two-class problem, the classifier struggles with a performance that closely approximates a coin flip experiment.

Based on these two examples, some empirical observations can be made.

It seems that feature vectors that are (class) conditionally dependent ('correlated') is not a sufficient condition that classifier performance will be poor.

As well, it seems that the situation of conditionally independent ('uncorrelated') features is not a necessary condition for good classifier performance.

What seems to be important is the separability of the class conditional distributions. Indeed, had a similarity metric such as [13] been employed, the results would indicate that Example #1 should have superior performance over that of Example #2. The problem is that metrics such as [13] provide extremely misleading results when the underlying conditional distributions are nonparametric.

Finally, in [14] the following phrase is encountered ("Dirty Secrets in Data Fusion"): "there is *no* substitute for a good sensor". For Example #1, a graph of the classifier performance of the 'best' sensor is shown in Figure 5 (sensor #2 was poorer for both classes). The performance is substantially poorer than the fused performance shown in Figure 2.

And so, the two sensors provide very poor single sensor performance when evaluated by themselves. However, when both sensors are combined, the joint performance is virtually without error. Feature vector separability changed going from a single sensor to both sensors combined. (This is discussed again in Case #4 in Section IV.) And this is in light that the 'correlation' (the class conditional dependence) is very high. The statement in [14] does not seem to apply in this instance.

Upon reflection, the statement in [14] has much more applicability to the problem of sensor data fusion for the purposes of state estimation and target tracking. In this instance, poor sensors (with poor measurement error covariance matrices) cannot provide an accurate state estimate since the state estimate error covariance matrix fails to converge to an accurate level with poor quality measurements.

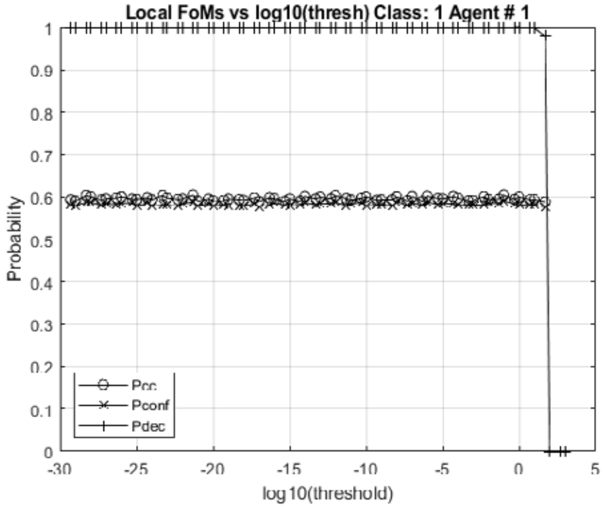

Figure 5: Sensor #1 Performance, Class #1 for Example 1

But in this application of ATR, it seems that there is more involved in the determination of classifier performance. It may not be that 'correlation' among the feature vectors leads to unacceptable performance. In practical ATR applications, class conditional dependence is more the norm rather than the exception. The classifier design must be sufficiently equipped to deal with these situations.

Furthermore, as will be discussed in Section IV, designs that are based on class conditional independence can work under certain circumstances, but there can be unintended consequences that can lead to poor performance. This applies not only to 'likelihood' fusion but also classifier/ID (identification) fusion as well.

3. Design

In this section, three procedures are developed that operate on the likelihoods. A method is described that allows conditioning on prior feature vectors for the third procedure. As well, a modified variant of a nonparametric classifier is discussed.

The setting for the design is as follows. There are N sensors, and each sensor provides one (and only one) feature vector x_n. with a dimension $\dim(x_n) = d_n$. The feature vectors x_n are assumed to arise from a commonly tracked target. There are M classes of interest, ω_m, for $m \in \{1, M\}$.

Three procedures are of interest: a) the feature level fusion approach, b) the 'intermediate' conditional independence likelihood fusion approach and c) the 'distributed' likelihood fusion approach. The decision law for all three procedures is the same. A max likelihood law is selected. The prior probabilities of the classes were not assumed to be known.

For the feature level fusion procedure, the decision law is

$$\delta(x^N) = \delta(x) = \omega_{m*} \iff$$

$$p(x|\omega_{m*}) \geq p(x|\omega_m)\ \forall m \neq m *$$

and, (1)

$$p(x|\omega_{m*}) \geq t_U$$

$$\delta(x) = \omega_{M+1} \quad o.w.$$

Here, x^N is the feature vector stack of all the N feature vectors provided by the N sensors and re-notated to x to simplify notation:

$$x^N = x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \tag{2}$$

The decision law in (1) states that ω_{m*} is the selected class when its conditional likelihood is higher than any other class and it surpasses a barrier threshold t_U. If that fails then the classifier decides class ω_{M+1} which is the nodeclare class.

Furthermore, it should be made clear that the likelihoods in (1) (and throughout the remaining designs) are, at best, estimates of the true likelihood. This is because there is no assumption that the underlying class conditional distribution functions follow any known parametric (statistical) form.

The decision procedure of (1) is termed the 'Centralized' (C) procedure since it has, at its disposal, all of the feature vectors for implementing the decision law of (1).

For the 'intermediate' likelihood fusion procedure, the ususal assumption of class conditional independence is invoked [10-12]. As such, the joint conditional distribution factors for N feature vectors:

$$p(x|\omega_m) = \prod_{n=1}^{N} p(x_n|\omega_m)$$

And so, the decision law of (1) becomes

$$\delta(x) = \omega_{m*} \iff$$

$$\prod_{n=1}^{N} p(x_n|\omega_{m*}) \geq \prod_{n=1}^{N} p(x_n|\omega_m)\ \forall m \neq m *$$

and, (3)

$$\prod_{n=1}^{N} p(x_n|\omega_{m*}) \geq t_U$$

$$\delta(x) = \omega_{M+1} \quad o.w.$$

As can be seen by (3), the stacked feature vector has been replaced by the likelihoods of the separate (sensor) feature vectors. Only the likelihoods are involved in the calculation of (3). These likelihoods are produced locally at each sensor and then sent to the location where (3) is calculated. Note that the factorization is *only* implemented between sensor feature vectors, not within the feature vector of a single sensor.

The procedure of (3) is termed the (class) Conditional Independence (CI) procedure.

Yet another procedure is needed. The CI procedure assumes (class) conditional independence. A distributed procedure is needed wherein this assumption can be relaxed. In order to do this, another factorization of $p(x|\omega m)$ is needed in order to obtain a distributed procedure that allows for the relaxed assumption. Such a factorization is immediate – the Bayes chain rule.

An example of the chain rule for three feature vectors is:
$p(x_3, x_2, x_1|\omega_m) = p(x_3|x_2, x_1, \omega_m)p(x_2|x_1, \omega_m)p(x_1|\omega_m)$

An iterative formula for $p(x|\omega_m)$ can be developed. Let $q_1(\omega_m) = p(x_1|\omega_m)$, and $r_1(\omega_m) = 1$ then $p(x_1|\omega_m) = q_1(\omega_m)r_1(\omega_m)$. Since $p(x_2, x^1|\omega_m) = p(x_2|x_1, \omega_m)p(x_1|\omega_m)$ then, this can also be written as $p(x_2, x_1|\omega_m) = q_2(\omega_m)r_2(\omega_m)$ with $q_2(\omega_m) = p(x_2|x_1, \omega_m)$ and $r_2(\omega_m) = q_1(\omega m)r1(\omega m)$

Inducing accordingly, an expression for $p(x|\omega_m)$ is $p(x|\omega_m) = q_N(\omega_m)r_N(\omega_m)$

The decision law for this procedure immediately becomes

$$\delta(x) = \omega_{m*} \iff$$

$$q_N(\omega_{m*})r_N(\omega_{m*}) \geq q_N(\omega_m)r_N(\omega_m)\ \forall m \neq m *$$

and,

$$q_N(\omega_{m*})r_N(\omega_{m*}) \geq t_U \tag{4}$$

$$\delta(x) = \omega_{M+1} \quad o.w$$

The procedure of (4) is termed the Distributed (D) procedure.

It is a minor note, but it is possible to perform subjoint calculations for the iteration procedure of (4). Consider the calculation for four feature vectors:

$$p(x_4, x_3, x_2, x_1|\omega_m) = p(x_4|x_3, x_2, x_1, \omega_m) \cdot$$
$$p(x_3|x_2, x_1, \omega_m)p(x_2|x_1, \omega_m)p(x_1|\omega_m)$$
$$= p(x_4|x_3, x_2, x_1, \omega_m)[p(x_3, x_2|x_1, \omega_m)]p(x_1|\omega_m)$$

The term in brackets is a subjoint combination of two feature vectors. The chain rule offers this level of flexibility of calculation for the determination of the final conditional likelihood. In this effort, the procedure of (4) was followed.

At this point, a discussion of the likelihoods that appear in (4) needs some development. It is apparent that feature vectors appear on the right-hand side of the condition along with the class of interest. This is atypical and is subsequently discussed.

It is the standard business of either a parametric or nonparametric classifier to compute the following $p(x|\omega_m)$

It is another thing entirely to compute $q_n(\omega_m) = p(x_n|x_{n-1}, x_{n-2}, \cdots, x_1\omega_m)$

Here, the likelihood has to be computed to include the conditioning upon the prior feature vectors. So, the conditioning does not just depend on the class, but also the previous features vectors to complete the Bayes chain rule calculation. To address this need, a novel implementation of the factorization was developed.

To enforce the conditioning on the previous feature vectors, a confining set of lines are used (in a vector space, these lines become confining hyperplanes). This is diagramed for a simple case in Figure 6.

Since the underlying distributions do not obey a standard statistical distribution, the usage of training vectors must be used as the probabilistic mass (also called the support). Figure 6 shows the scatter of support for both features. The training feature vectors are used to provide an estimate of (conditional) likelihoods to process a test vector.

Here, two features are shown, x_1 and x_2. Given x_1, the desire is to compute $p(x_2|x_1, \omega_m)$. Confining lines (hyperplanes in a vector space) are constructed about x_1 ($x_{n-1}, \cdots, x_1$ in general). The shaded area in Figure 6 depicts the constrained area (volume, in higher dimensions) of probabilistic support for use in the calculation of $p(x_2|x_1, \omega_m)$. The volume outside of the shaded area can't be used for the calculation of the conditional likelihood. This is necessary in order to enforce the conditioning on x_1.

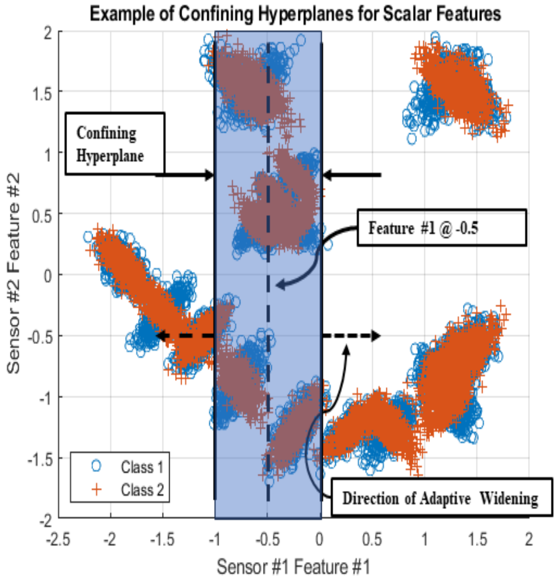

Figure 6: Conditioning on Prior Feature Vectors

The confining volume is termed a hyper-wedge. The wedge is assessed for support and the amount of support is compared to a threshold. If that threshold is not met a simple adaptation scheme is employed where the width of the wedge is increased and the amount of support is recalculated. If the support fails the wedge is widened again up to a final limit. If it fails at the widest iteration then what support is available is used as the final support for assessment by the nonparametric classification method which is described next.

The hyper-wedge is a simple device that is used to approximate the conditional likelihood. It is only an estimate of the true conditioning. There were no attempts to employ other support-based adaptive manifolds in this effort.

It should be important to note that if the hyper-wedge is extended to such a point that it encompasses the entire span of the feature space, it then becomes nothing other than the CI procedure since the conditioning on the feature has been relaxed. The D procedure was not allowed to expand to such a wide level.

A means of classifying test vectors using a training set of feature vectors is still needed. There is a fairly substantial collection of such methods. Most methods can be qualified as being global in nature (i.e., the whole training set is used spanning the entire feature space) and local in nature (a limited area about the test vector is used). Given the nature of the nonparametric conditional distributions anticipated (to include multimodality), it was a design decision to select a locally oriented method.

A modified variant of the k-Nearest Neighbors method was selected for this effort [15]. It is rather simple to implement (although the search procedure can drive the compute cycles), and provides a very highly articulated decision surface in complicated feature overlap situations. The modifications involved how nearest neighbors were found and changing the voting logic to pseudo-likelihoods (not unlike Parzen's method). The determination of neighbors is different than that discussed in [15]. Instead, the number of neighbors is determined on a class-by-class basis. This is because some classes may have more member features than others, especially in light of the hyper-wedge that is employed. The neighbors from each class are then weighted by ${}^{1}/_{d}$, where d is the distance between the test vector and the neighbor training vectors. These weighted values are then normalized by the number of nearest neighbors available from that class. This value becomes the pseudo-likelihood for that class. This classification method was used in all three of the procedures discussed above. (It is a minor point, but the pseudo-likelihoods can easily be normalized to correspond to the properties of correct likelihoods – this was not performed here because it was not necessary. The pseudo-likelihoods all have similar scale among each of the three procedures separately – the decision laws of all three procedures are adjusted accordingly along with the thresholds employed.)

4. Results

In this section, three interrelated sets of results are presented and discussed. A final example is also shown which helps crystallize two important questions that are relevant to ATR.

For the performance results to be discussed the Figures of Merit (FoMs) need to be introduced. Three FoMs are used and are traced in the performance curves as the threshold sweep is conducted. An ambiguity array (sometimes referred to as a confusion matrix) is presented in Table I (at a fixed threshold) which is used to describe the FoMs. The first FoM is the probability of declaration (P_{dec}). Taking class #2 in Table I as an example, this probability is determined by summing up the entries in the class #2 row except for the last column entry (which is the no-declare class). This value becomes the numerator. The denominator is the full sum across the class #2 row. For class #2, P_{dec} then becomes (0+97+1)/100 or 98%. The next FoM is P_{cc} which is the correct class (declaration) probability. This is calculated by taking the count at the intersection of the class #2 row and the class #2 column and dividing by the sum of the class #2 row except for the last column. Doing so provides a value of P_{cc} of (97)/(0+97+1) = 0.989. The third FoM is the probability of confidence, P_{conf}, and this is defined as the same intersection of the class #2 row with the class #2 column divided by the column sum at column #2. In the ambiguity array this value is: (97)/(2+97+3)= 0.989. This latter metric measures the degree of confidence of a class declaration. When a certain class is declared, it measures how often it arises from the correct class.

True Class	Declared Class			
	1	2	3	4
1	94	2	3	1
2	0	97	1	2
3	1	3	93	3

Table 1: Ambiguity Array for a Three-Class Problem.

The results shown are from simulation exercises. Each class for the training set (of feature vectors) was composed of a Monte Carlo of size 10,000. The test set was similarly constructed with a Monte Carlo size of 10,000 for each class. Each test feature vector is compared against the training set, likelihoods are gathered for the three procedures, and three decisions (at a given threshold) are made and then scored using the FoMs previously described. This is then repeated as the threshold is moved to its next value.

A scatter diagram for Case #1 is shown in Figure 7. There are two classes and two sensors with feature vector dimensions of: [5,7] for a total dimensionality of 12. The underlying distributions are clearly nonparametric in nature, to include multimodality. Four major modes are shown. The scatter support also seems to show poor feature separability since the two classes seem to be heavily overlapped. However, this is a figure of two dimensions of a 12-dimensional space. The training (and test) feature vectors are presented in a normalized space. The normalization used is the usual feature coordinate standardization. Each feature coordinate for all the classes is standardized by determining the mean and standard deviation and normalized accordingly.

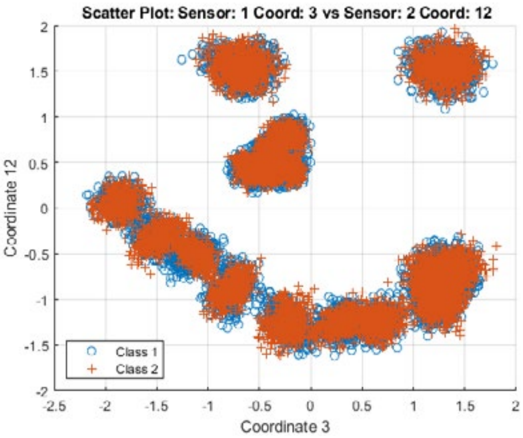

Figure 7: Scatter Diagram of Case #1

The performance traces for the three procedures are depicted in Figures 8-10. Procedure C is the first figure and shows reasonable performance with a P_{cc} of 0.965 for Class #1 (Class #2 was similar). Procedure CI is also performing well with a P_{cc} of 0.939. Procedure D performs well with a P_{cc} of 0.962, a departure from procedure C of less than 1%. For this situation, the hyper-wedge settings were: initial half-width: 0.12, iteration half-width: 0.15, maximum halfwidth: 0.72. The adaptive hyper-wedge of the procedure seems to perform adequately in approximating the joint method of procedure C. Procedure CI only lags behind procedure C by about 2.5% which is quite reasonable, despite the possible concern that the features may not be conditionally independent.

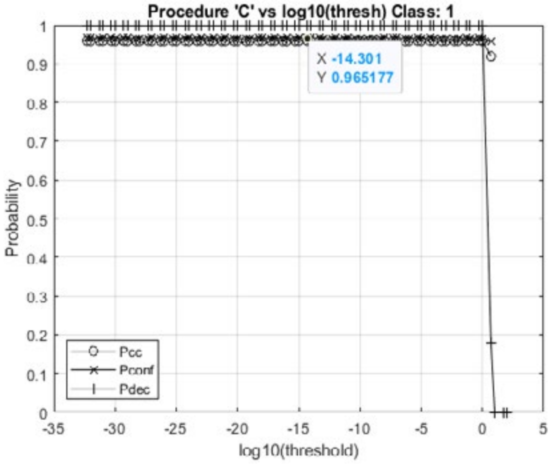

Figure 8: Procedure C Results, Case #1

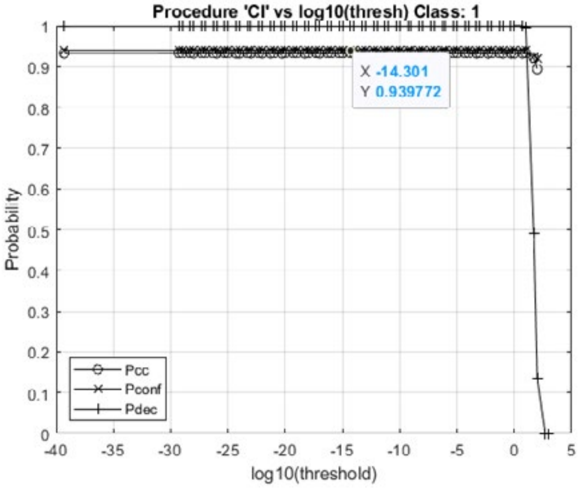

Figure 9: Procedure CI Results, Case #1

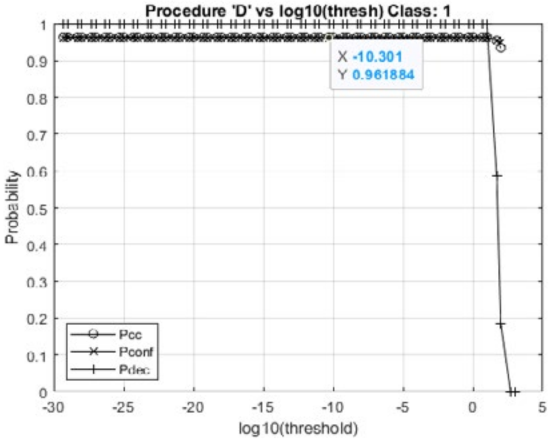

Figure 10: Procedure D Results, Case #1

The results for Case #2 are shown in Figures 11-13. The same general statistical specification from Case #1 was extended to this case. However, now there are four classes and five sensors. The feature vector dimensions for each of the sensors was: [4,5,6,5,6] for a total dimensionality of 26. The results of the best performing class for the CI procedure (which was class #2) were used in posting the results for the C and D procedures (the worst performing class was less than 1% from the class #2).

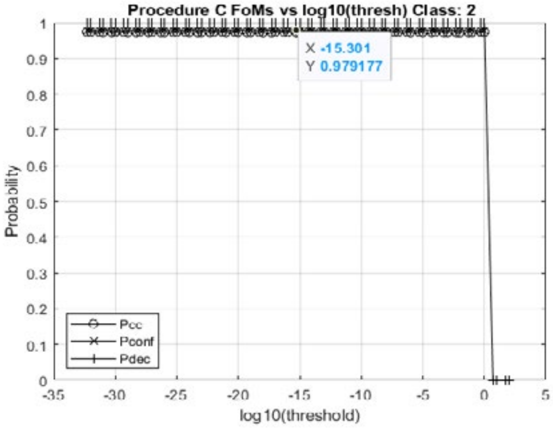

Figure 11: Procedure C Results, Case #2

From Figure 11, procedure C shows a performance of 0.979 which is superior. In Figure 12, procedure CI comes in at about 0.866 which is about 11% below that of procedure C. The performance is not bad unless it failed to meet a requirement higher than that. Procedure D (Figure 13) produced a result of 0.956, which was about 2.3% lower than procedure C. As well, the performance is not bad unless, it too, failed to meet a requirement. For procedure D, the hyper-wedge settings were: initial halfwidth: 0.12, iteration half-width: 0.15, maximum halfwidth: 0.72. For this case, procedure D is providing a reasonable approximation to the C procedure.

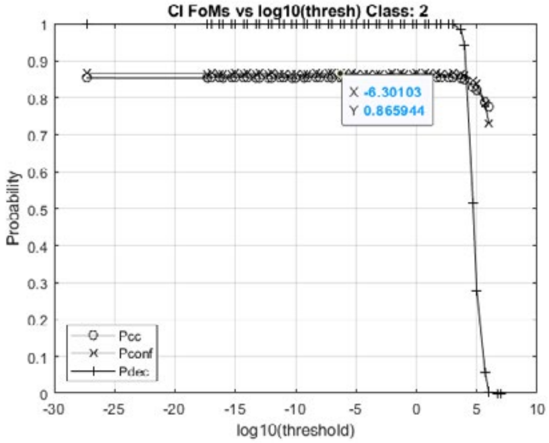

Figure 12: Procedure CI (Best) Results, Case #2

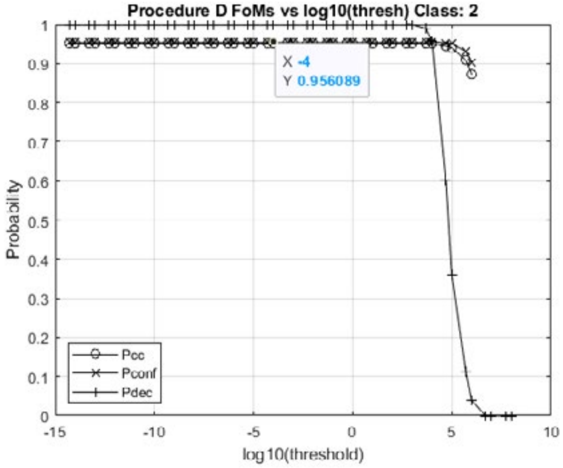

Figure 13: Procedure D Results, Case #2

The third Case is a comparative test to Case #2 in that there are still are four classes and five sensors. However, the feature vector dimensions for each of the sensors is now: [1,2,1,2,1] for a total dimensionality of 7 (which, to some extent, is the experiment of "missing features" as presented in [16]). The result of the best performing class from procedure CI is class #3 and it used for comparison against procedures C and D. The results of the three procedures are depicted in Figures 14-16.

The results from procedure C are shown in Figure 14. Compared to Figure 11, the procedure is (apparently) suffering from the lack of feature vector dimensionality. The result of 0.836 is 14% lower than that of Case #2. This performance may not meet an ATR requirement.

The result (Class 3) of procedure CI is shown in Figure 15. The results show a dramatic departure from that of Figure 12. The performance is at 0.441 (the worst performing class came in at 0.381). This is a severe drop form the result of Figure 12.

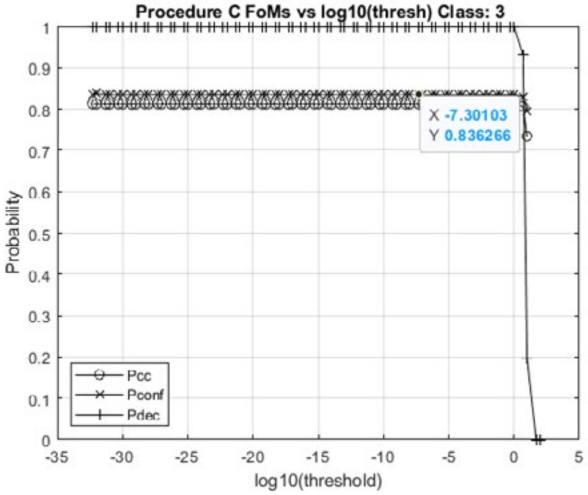

Figure 14: Procedure C Results, Case #3

The result of procedure D is shown in Figure 16. The result here also exhibits a downward trend in performance. The performance was only 0.774 which is 6.20% below the performance of the C procedure. The result from Case 2 was only 2.3% below procedure C in Case 2. Settings for the hyper-wedge were: initial half-width: 0.08, iteration half-width: 0.04, maximum half-width: 0.24. This was an aggressive setting. It was so aggressive that P_{dec} departed from unity as the hyper-wedge eliminated all the support. This is shown in Figure 16 at the top. P_{dec} dropped down to 0.9996 at the sensible edge of the threshold sweep. It should be clear that procedure D is having trouble approximating procedure C perfectly, while also struggling with feature vector separability (it can't suffer from conditional dependence as it is designed exactly for that condition).

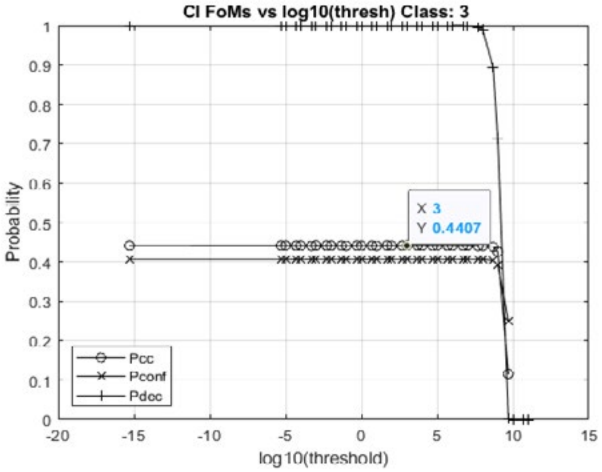

Figure 15: Procedure CI Results, Case #3

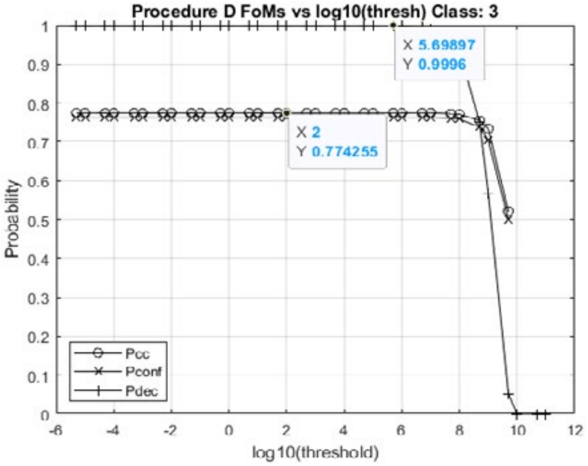

Figure 16: Procedure D Results, Case #3

The final case to discuss is Case #4 which is a separate case from the prior three cases. The case is the so-called '4corners problem' which is used as an empirical proof-bycontradiction for a related matter. It's relevance here is acutely useful in light of the results of Case #3. The results of this case are shown in Figures 17-21.

Figure 17 shows the feature scatter of two classes in two dimensions (each sensor has scalar features). The scatter exhibits stellar separability, so much so that the human eye can classify these classes immediately. If a (test) feature vector arrived at the coordinates of (1,-1), it would immediately be classified as class #2. This was only possible by the separability of the scatter of the features.

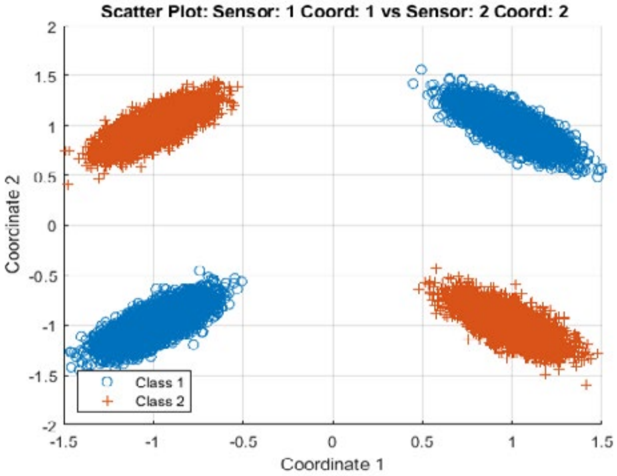

Figure 17: Feature Scatter for Case #4

Figure 18 shows the performance of procedure C for this problem. In agreement with what can be seen visually, the performance is at unity – perfect.

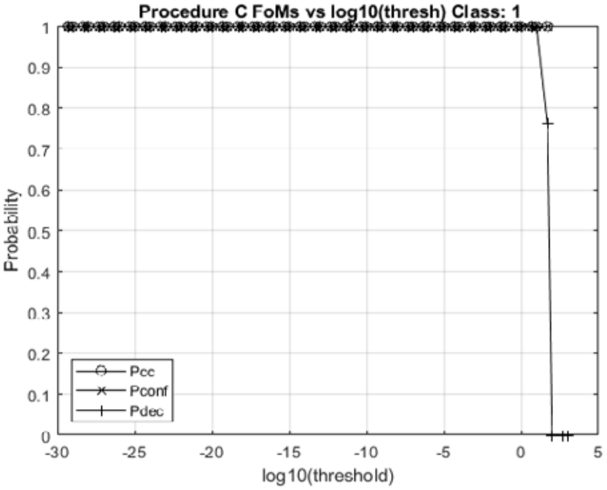

Figure 18: Procedure C Results, Case #4

But then, procedure CI, in Figure 19 displays a result that is no better than a coin-flip experiment (there are only two classes here). The results are as bad as they can be. This is a death knell case for this procedure. The reason for this is quite simple. If the 2-dimensional figure was collapsed along the y-axis onto the x-axis, the feature scatter would be compressed onto the x-axis, and the two classes would posit themselves on top of each other. Viewing up the yaxis, it can be seen that the scatter above lies right on top of the scatter below (but they are from different classes). The same holds true compressing the x-axis onto the y-axis. These operations completely destroy the innate separability of the 2-dimensional problem. But this is exactly what the CI procedure does when computing likelihoods from a single sensor and then combining those likelihoods with the remaining sensors. The CI procedure, by its design, cannot 'visualize' the full field of the scatter (really, the probabilistic support) since its focus is based on a single sensor alone, prior to amalgamating the results from the remainder of the sensors (with their partial 'view' as well) to compose a final classification result.

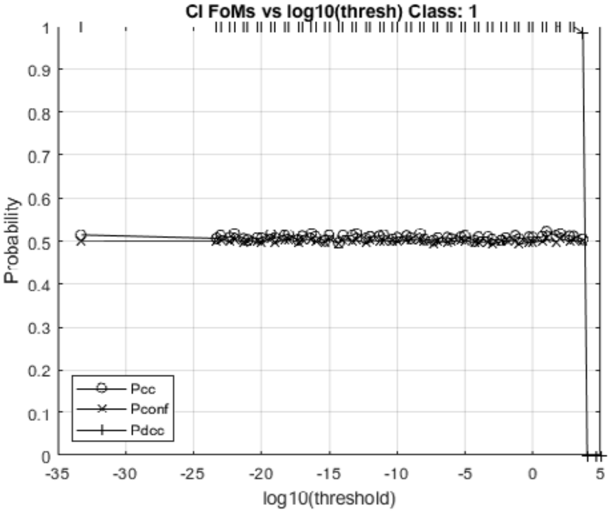

Figure 19: Procedure CI Results, Case #4

Procedure D is shown in Figure 20. It replicates the C procedure, as it should. The procedure is completely agnostic to feature vector collisions (that are occurring in a single dimension) since it is iterating through all the sensors before making a decision. The D procedure, even in challenging cases, seems to provide a very reasonable approximation to the C procedure.

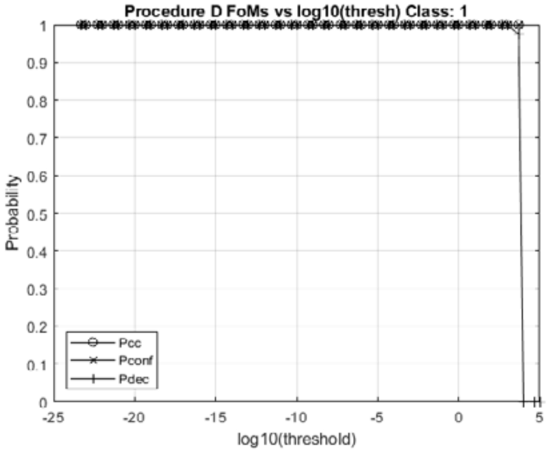

Figure 20: Procedure D Results, Case #4

Finally, the performance of sensor #2 (sensor #1 was equivalent) is shown. The result prides no surprise and is virtually the same as the CI procedure, for exactly the same reasons given above.

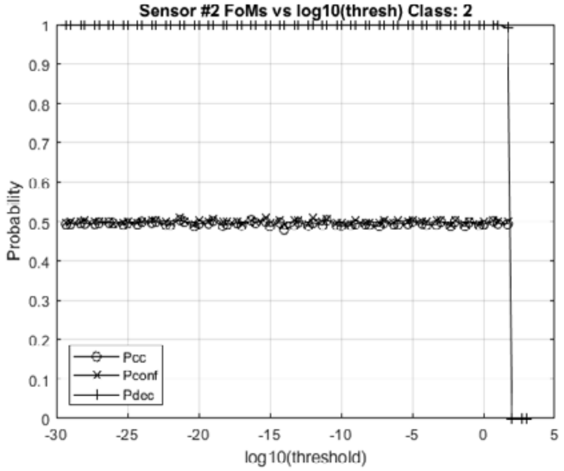

Figure 21: Sensor #2 Results, Case #4

But this begs an important question. Given the problem of Case #4, suppose each sensor was allowed to make a classification decision and submit their results to a classification fusion center. The performance of a fused classification/ID procedure (no matter what operators are used: Boolean logic, etc., etc.) on the separate sensor classifications would be equally poor. The same concern holds true for Artificial Intelligence (AI) learning techniques which are popularly researched topics at this time. Applying learning techniques when the information available are the separate sensor decisions and/or their likelihoods is tantamount to information that has the quality of a coin-flip. When the 'visualization' of the full amount of feature information has been collapsed along each sensor (thus abrogating the feature separability), it becomes a *fait accompli*. The damage was done before any possible rectification can be pursued subsequently. This example serves as a warning to the dangers of not having a complete portfolio of all the sensor feature vectors before making a decision. Regardless of whether an intermediate conditional independent likelihood fusion procedure (procedure CI) is performed or a classification/ID fusion procedure is used, the results are going to depart severely from that performance which is possible. *Caveat emptor.*

Finally, a cursory examination of the scatter in Figure 17 does not reveal any severe instance of 'correlation' or (class conditional dependence). But a second look reveals some issues. Given knowledge that the test vector arises from class #1, and given that sensor #1 measures a value of -1, then it is not only improbable, but indeed impossible to get a value from sensor #2 that is above -0.4, despite a considerable amount of mass that lies above 0.5 (for class #1). The notion of conditional dependence can be frustratingly non-trivial, especially in a highly dimensioned feature space.

5. Summary

A distributed Automatic Target Recognition procedure has been developed and exercised for the purpose of processing feature vectors and providing a consolidated class decision. The performance is quite good and closely approximates a centralized procedure.

It should be clear that issues exist in Automatic Target Recognition (classification), as well as the ensuing methods for sensor data fusion for the purposes of classification/recognition (ID/classifier fusion).

It is very unclear to what *degree* that (class) conditional dependence has on classifier performance (even in the instance that conditional independence has been assumed). (Conditional independence is exact. That is when the factorization products of the likelihoods of the separate sensors equals the likelihood of the joint conditional.) Conditional dependence is not exact. It simply means that a factorization of the joint distribution will not yield likelihoods that equal the calculation of the likelihood from the joint conditional distribution. What matters is the *degree* of departure from conditional independence.

Furthermore, mathematical measures of conditional dependence are in their infancy (principal components and correlation coefficients). The measures do not extrapolate well to classifier performance. What complicates the situation is that conditional dependence occurs simultaneously with feature vector separability problems.

The impact of poor feature vector separability also seems to be without reasonable formal mathematical measures, especially for underlying distribution functions which are nonparametric in nature (to include multimodality). These measures need to be developed, not only at full dimensional space of the feature vectors, but also at the single sensor levels (wherein only a partial picture of the separability is available).

Of these two problems, it seems apparent that feature vector separability dominates as the most telling issue.

These problems make the field of target recognition/classification more a practice in art than an exercise of engineering discipline that is well anchored by mathematical formalism (with the aspiration of mathematical rigor to follow).

These two areas require further research.

References

1. Tenney, R. R., & Sandell, N. R. (1981). Detection with distributed sensors. *IEEE Transactions on Aerospace and Electronic systems,* (4), 501-510.
2. Reibman, A. R., & Nolte, L. W. (1987). Optimal detection and performance of distributed sensor systems. *IEEE Transactions on Aerospace and Electronic Systems,* (1), 24-30.
3. Thomopoulos, S. C., Viswanathan, R., & Bougoulias, D. C. (1987). Optimal decision fusion in multiple sensor systems. *IEEE Transactions on Aerospace and Electronic Systems,* (5), 644-653.
4. Reibman, A. R., & Nolte, L. W. (1987). Design and performance comparison of distributed detection networks. *IEEE Transactions on Aerospace and Electronic Systems,* (6), 789-797.
5. Dasarathy, B. V. (1991). Decision fusion strategies in multisensor environments. *IEEE transactions on systems, man, and cybernetics, 21*(5), 1140-1154.
6. Hall, D. L., & Llinas, J. (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE, 85*(1), 6-23.
7. Schubert, C. M., Oxley, M. E., & Bauer, K. W. (2005, March). The inclusion of correlation effects in the performance of multiple sensor and classifier systems. *In 2005 IEEE Aerospace Conference* (pp. 1-11). IEEE.
8. Stubberud, S. C., Kramer, K. A., & Geremia, J. A. (2007). Feature object extraction: evidence accrual for the level 1 fusion classification problem. *IEEE Transactions on Instrumentation and Measurement, 56*(6), 2705-2716.
9. Laine, T. I. (2005). *Optimization of automatic target recognition with a reject option using fusion and correlated sensor data*. Air Force Institute of Technology.
10. Demirbas, K. (1988). Maximum a posteriori approach to object recognition with distributed sensors. *IEEE transactions on aerospace and electronic systems, 24*(3), 309-313.
11. Rao, B. S., & Durrant-Whyte, H. (1993). A decentralized Bayesian algorithm for identification of tracked targets. *IEEE Transactions on Systems, Man, and Cybernetics, 23*(6), 1683-1698.
12. Wei, M., Gan-Lin, S., & Hong-feng, W. (2003, July). Distributed bayesian target identification algorithm. *In Sixth International Conference of Information Fusion, 2003. Proceedings of the* (Vol. 2, pp. 1379-1383). IEEE.
13. Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distribution. *Bulletin of the Calcutta Mathematical Society, 35*, 99-110.
14. Liggins II, M., Hall, D., & Llinas, J. (Eds.). (2017). *Handbook of multisensor data fusion: theory and practice*. CRC press.
15. Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis* (Vol. 3, pp. 731-739). New York: Wiley.
16. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Bayesian decision theory. *Pattern classification, 11*(4), 99-102.