A Fusion Analysis and Evaluation Tool for Multi-Sensor Classification Systems

ROMMEL NOVAES CARVALHO
KUOCHU CHANG

Multi-Sensor Fusion is founded on the principle that combining information from different sensors will enable a better understanding of the surroundings. However, it would be desirable to evaluate how much one gains by combining different sensors in a fusion system, even before implementing it. This paper presents a methodology and tool that allows a user to evaluate the classification performance of a multi-sensor fusion system modeled by a Bayesian network. Specifically, we first define a generic global confusion matrix (GCM) to represent classification performance in a multi-sensor environment, we then develop a methodology with analytical convergence bounds to estimate the performance. The resulting system is designed to answer questions such as: (i) What is the probability of correct classification of a given target using a specific sensor individually? (ii) What if a specific set of sensors combined together are used instead? (iii) What is the performance gain by adding another sensor to this set? and (iv) Which sensors provide a better cost/benefit ratio? These questions are answered based on the probability of correct classification that can be analytically estimated using Bayesian inference with the given sensor models defined by confusion matrices. The principle that combining information enhances the understanding of the surroundings is also supported by the analysis made in example models for air target tracking and classification using the developed tool.

Manuscript received November 2, 2009; revised January 18, 2011, March 28, 2011, April 22, 2012; released for publication April 23, 2012.

Refereeing of this contribution was handled by Mujdat Cetin.

Authors' address: Department of Systems Engineering & Operations Research, Volgenau School of Information Technology and Engineering, George Mason University, Fairfax, VA 22030, E-mail: (rommel.carvalho@gmail.com, kchang@gmu.edu).

1557-6418/12/$17.00

1. INTRODUCTION

Fusion of information from multiple sources to achieve performances exceeding those of individual sources has been recognized in diverse areas [17] such as reliability, forecasting, pattern recognition, neural networks, decision fusion, and statistical estimation. In engineering systems, the fusion methods have proven to be particularly important since they can provide system capabilities with multiple sensors significantly beyond those of single sensor systems. Multi-sensor data fusion allows the combination of information from sensors with different physical characteristics to enhance the understanding of the surroundings and provide the basis for planning, decision-making, and control of autonomous and intelligent machines. It seeks to combine information from multiple sensors and sources to achieve inferences that are not feasible from a single sensor or source.

To fully exploit the capabilities of a fusion system, modeling and performance evaluation methodologies are critical in order to optimally design and effectively evaluate fusion performance of multiple heterogeneous sensor data. In particular, a systematic approach to evaluate the overall performance of the system is indispensable. To allow developers and users to assess their fusion system performance under various conditions before a data fusion system is deployed, a tool based on the Fusion Performance Model (FPM) [8] was developed with a focus on one of the most important performance measures, spatial and classification performance modeling and prediction. Note that the purpose of the FPM is to predict performance given sensor suite and operating conditions.

For a sensor fusion system, typical questions that could be asked would be "what is the best achievable performance, and is it good enough?" The FPM will be able to answer the first question and if the answer is "not good enough," a sequence of "what if" scenarios can be added for FPM to conduct new assessments. Those scenarios may include changing operating conditions, such as signal-to-noise ratio (SNR), geometry, and revisit rate, to name a few of the existing sensors or adding new sensors. The assessment results can then be used to better manage sensors and allocate system resources.

While the FPM model described in [8] developed a kinematic performance prediction methodology and defined the classification performance model and [7] described an analytical method to predict classification performance and an efficient approximate formula to estimate the average probability of correct classification given sensor characteristics, there is still a lack of effective tools to evaluate a fusion system performance as described in [8] and [7] in an easy and accessible way in order to make the assessment results promptly available to better control sensors and allocate system resources.

In particular, for the case of discrete reporting elements—the sensor exploitation system's estimate of target type or activity, it is more complicated to predict fusion performance for target/activity identification or situation assessment. For example, observations of the target's attributes, such as feature-modulation frequency, radar emissions characteristics, and visual image, may be used to establish target identity. They are based on a transformation between observed target attributes and a labeled identity. Methods for identity estimation involve [15] pattern recognition techniques based on clustering algorithms, neural networks, or decision-based methods such as Bayesian inference, Dempster-Shafer's method, or weighted decision techniques.

The focus of this paper is on developing a methodology and software tool to model and evaluate performance of a multi-sensor classification system. Specifically, we define a generic classification performance metric for multi-sensor fusion, called global confusion matrix (GCM), from the local sensor confusion matrices described by Bayesian network models. We then develop a stochastic simulation methodology with analytical convergence bounds to estimate the performance. Based on the methodology, a software tool is developed to help a decision maker answer the following questions: (i) What is the probability of correct classification of a given target using a specific sensor individually? (ii) What if a specific set of sensors combined together is used instead? (iii) What is the performance gain by adding another sensor to this set? and (iv) Which sensors provide a better cost/benefit ratio? We apply Bayesian network (BN) to model the relationship between target variable and various levels of observables and compute the defined performance accordingly. We assume that the BN model is given, should that be created from expert knowledge, learning from past data or any other method, and our goal is to assess its performance. In other words, the methodology is generic and independent of the source of the model. However, if the model itself is inaccurate due to limited training data or insufficient domain expertise, then we have to take into account the model uncertainty when assessing the performance. We demonstrate the types of evaluation and conclusions that can be achieved with this tool using an example model from a model-based identification (MBID) component described in [5].

The evaluation process described was implemented as an extension of a free, Java based, and open-source probabilistic network framework, UnBBayes [2–4, 10, 13]. This framework proved to be an interesting alternative since it already had Bayesian networks representation, simulation, and inference algorithms built-in, making the development of the evaluation module much easier and faster.

This paper is organized as follows. Section 2 reviews the main concepts concerning the Fusion Performance Model described in [7, 8]. Section 3 describes the technical approach used to implement the method to predict classification performance based on the FPM. Section 4 derives an analytical convergence property of the evaluation methodology and predicts number of simulation trials needed in order to achieve a desirable error bound. Section 5 shows an overview of the probabilistic network framework used, UnBBayes, to implement the evaluation module. Section 6 presents the evaluation tool and its use in several example models. Finally, Section 7 relates the main contributions of this paper and some future work.

2. MODELING CLASSIFICATION FUSION PERFORMANCE

Currently, data fusion systems are used extensively for target tracking, automated identification of targets, situation assessments, and some automated reasoning applications [15]. This paper uses a Bayesian network model that is a part of a model-based identification (MBID) component of an effort to design a decision-theoretic sensor management system. This model, described in [5], is used for incorporating target identification (ID) into a multiple-hypothesis tracking (MHT) system in a multi-sensor environment.

Bayesian networks [9] are directed acyclic graphs (DAGs), where the nodes are random variables, and the arcs specify the independence assumptions that must hold between the random variables (the arc points from the parent to the child node). These independence assumptions determine what probability information is required to specify the probability distribution among the random variables in the network.

To specify the probability distribution of a Bayesian network, one must give the prior probabilities of all root nodes (nodes with no parents) and the conditional probabilities of all other nodes given all possible combinations of their parents. Bayesian networks allow one to calculate the conditional probabilities of the nodes in the network given that the values of some of the nodes have been observed.

In addition to the convenient and flexible representation, a major benefit of using BNs is the existence of many powerful probabilistic inference algorithms, such as the distributed algorithm [16], the influence diagram algorithm [18], the evidence potential algorithm [14], simulation algorithms [11, 20], and the symbolic probabilistic inference (SPI) algorithm [19].

For MBID system mentioned earlier [5], the BN is used to relate the target states to the detected measurements at the sensors. Each evidence node represents the detected observation from a source at a given sensor. The conditional probabilities depend on the propagation from the target to the sensor, array gain, detection thresholds, etc. Other information such as relative geometry between target and sensor, the strength of the target, and the transmitted energy can also be summarized in the conditional probability of the received measurement given the target state.

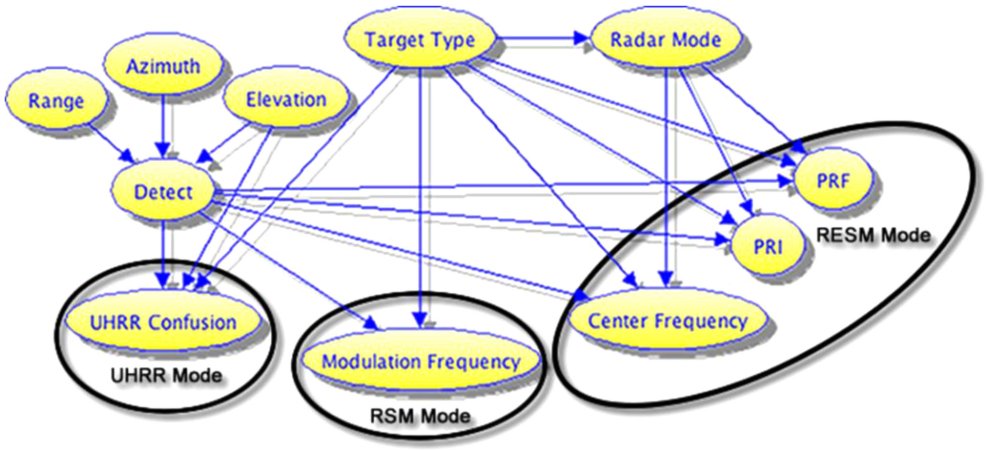

Fig. 1. A Bayesian network model example for a model-based identification (MBID) system.

It is assumed that in the system, there are two types of sensors: electronically scanning radar (ESA) and Infrared search and track (IRST). In addition to the regular search and update capabilities, the ESA radar is modeled to have three identification modes: ultra-high-resolution radar (UHRR), radar signal modulation (RSM), and radar electronic support mode (RESM). Since the radar detection and observation processes are fairly complicated and cannot be easily expressed in a simple form, a Bayesian network is used to model the processes and compute the association likelihoods as well as manipulate the target state distribution. In this system, a centralized fusion architecture is assumed, i.e., data collected from multiple sensors are pooled together in a central site where they are combined.

UHRR is an active technique and is basically an imaging technique that will be able to identify features of an airplane and therefore infer the target type. RSM is an active technique that can detect a target feature-modulation frequency. RESM is a passive technique that can observe the characteristics of the target's radar emissions. Based the observed features of the radar signal, the MBID system will infer a radar mode that will, in turn, be an evidence for a target type.

Figure 1 shows the BN that contains all three identification modules. Note that at any given moment, only one module can be active. In other words, only one type of evidence can be attached to the network. In the figure, the UHRR module is represented by the node "UHRR Confusion," RSM is represented by the node "Modulation Frequency" (MF), and RESM is represented by the remaining three nodes, "Center Frequency" (CF), "PRI," (pulse repetition interval) and "PRF" (pulse repetition frequency). Note that in all three radar modes, the observation is dependent on the probability of detection represented by the node "Detected." Probability of detection is a function of target range and aspect angle. The measurements from the three radar modes can either be discrete or continuous values. For example, the observation of UHRR is the actual target type, which can only be one of the given values. On the other hand, the observations for other radar modes have continuous values and can assume any value within the defined ranges.

Despite these qualitative notions and quantitative calculations of improved system operation by using multiple sensors and fusion processes, actual implementation of effective data fusion systems is far from simple. In practice, the combination of sensor data may actually produce worse results than could be obtained by tasking the most appropriate sensor in a sensor suite. This is caused by the attempt to combine accurate (i.e., good data) with inaccurate or biased data, especially if the uncertainties or variances of the data are unknown [15]. Before a data fusion system is deployed, developers and users need to be able to assess their fusion system performance under various conditions. This paper develops a tool based on the Fusion Performance Model (FPM) described in [8]. The focus is on predicting the classification performance. To do so, first we define the following terminologies [7].

Local Confusion Matrix (LCM): Local confusion matrices are the ones based on single sensor classification system observations. There are two types of LCMs—feature level LCM defined as the conditional probability/likelihood tables of the observable/evidence nodes given their parent nodes, and label (target ID) level LCM, $Pr(\text{Obs} \mid T = j)$, defined as the conditional probability tables of the observable/evidence given true target ID. It is the latter that will be used to obtain the global confusion matrix (GCM).

Global Confusion Matrix (GCM): Global confusion matrix is obtained based on multiple sensor classification observations with a given LCM over time. Each element in the GCM is defined as the probability of inferred[1] target class given true target class, $\text{GCM}(i, j) = \Pr(I = i \mid T = j)$. Note that the GCM is applicable to both single sensor and multiple sensor systems.

[1]The posterior probability of the target class being true given the sensor observations.

TABLE I
"Target Type" Node's Global Confusion Matrix Example

	B1	B2	R1	R2	N	Unk
B1	0.80	0.02	0.10	0.02	0.02	0.04
B2	0.02	0.70	0.02	0.14	0.08	0.04
R1	0.10	0.02	0.80	0.02	0.02	0.04
R2	0.02	0.14	0.02	0.70	0.08	0.04
N	0.02	0.15	0.04	0.15	0.60	0.04
Unk	0.04	0.04	0.04	0.04	0.04	0.80

Average Probability of Correct Classification (Pcc): Average probability of the class corresponding to the true class of the associated target. It is the average value of the diagonal elements of GCM, computed as,

$$Pcc = \frac{1}{k}\sum_i \text{GCM}(i,i) = \frac{1}{k}\text{Trace(GCM)} \qquad (1)$$

where k is the total number of target classes.

Table I illustrates a global confusion matrix example for the "Target Type" node in Fig. 1, where the columns represent the predicted class and the rows the true class.[2] Therefore, the diagonal elements represent the probabilities of correct classification. In this example, the probability of correct classification is 0.733.

3. COMPUTING CLASSIFICATION FUSION PERFORMANCE

To compute the GCM is to infer the target ID/type based on a series of sensor reports. This is similar to state-estimation, where the state of a system is estimated based on observed measurements. Similar to the Kalman filter, which allows for off-line estimation of the expected tracking performance (covariance matrix), there is a need for a systematic approach to evaluate the classification performance of a sensor, or multiple sensors.

In order to predict classification performance with a Bayesian network model, we need $\Pr(\text{Obs} \mid T = j)$, the sensor target ID level LCM. This can be done by "predicting" the observation distribution using forward inference given a target ID with either a simulation method or an exact (e.g., Junction tree [14]) algorithm depending on the network configuration. Given the LCM, the GCM can then be computed as,

$$\begin{aligned}\text{GCM}(i,j) &= \Pr(I = i \mid T = j) \\ &= \sum_{\text{Obs}} \Pr(I = i \mid \text{Obs}, T = j)\Pr(\text{Obs} \mid T = j) \\ &\approx \sum_{\text{Obs}} \Pr(I = i \mid \text{Obs})\Pr(\text{Obs} \mid T = j) \qquad (2)\end{aligned}$$

where I is the inferred target ID, T is the true target ID, and Obs is the sensor observation.[3]

[2]"B1" and "B2" are for Blue classes, "R1" and "R2" are for red classes, "N" is for neutral class, and "Unk" is for unknown class that do not belong to any other class.

[3]Note that to derive (2), a Markov chain property of T_{Obs} I is assumed.

Note that this will give us a square matrix where each row indicates that if the target $T = j$ is true, what is the probability of the sensor/classifier inferring it as $I = i$ given a single report Obs from a sensor. The performance measure can then be defined as the average correct classification probability as described earlier. When a total of n observations are reported by the sensors, the expression in (2) will need to be summed over all possible realizations of Obs, namely, an exponential enumeration of all $|\text{Obs}|^n$ possible realizations.

In general, the calculations for the elements of the GCM are computationally extensive. There are two ways to do so. One is to use the Monte Carlo approach where we randomly simulate the sensor observations based on a given BN model and recursively update the target state probability. Another way is to use analytical performance model.

Our goal here is to develop a mixed approach where analytical calculations will be performed whenever feasible. Otherwise, a stochastic simulation will be used. We have developed a very efficient polynomial-time analytical approach to approximate the GCM based on the assumption that the sensor observations are conditionally independent given the target ID [21]. Due to potentially high model complexity, we also developed an approximation method based on stochastic simulation. The algorithm is briefly summarized in Figure 2.

1. Select the target node, the corresponding evidence nodes, the condition on which we want to evaluate the BN, and the number of simulation trials to be generated from the model. Note that, in this implementation, only one target node is allowed.

2. Simulate the data based on the number of trials desired from the BN model using the stochastic sampling algorithm (see [6, 11–12, 20–21] for details). Note that in general the error of the estimated probabilities is inversely proportional to the sample size. More detail is given in Section 4.

3. Compute the approximate conditional probabilities, based on its frequency of occurrence, of the evidence nodes given target node, $\Pr(\text{Obs} \mid T = j)$.

4. Compute probabilities for predicted target ID given evidences using the probabilities computed in step 3.

$$\begin{aligned}\Pr(I = i \mid \text{Obs}) &= \frac{\Pr(\text{Obs} \mid I = i)\Pr(I = i)}{\Pr(\text{Obs})} \\ &= \frac{\Pr(\text{Obs} \mid I = i)\Pr(I = i)}{\sum_j \Pr(\text{Obs} \mid I = j)\Pr(I = j)} \qquad (3)\end{aligned}$$

where the prior probability $\Pr(I = i)$ is retrieved from the marginal distribution of the target node.

5. Finally, compute the global confusion matrix, as described in equation (2), by using the values computed in steps 3 and 4.

Fig. 2. Estimating GCM with stochastic simulation.

Some of the nodes in the network are neither target nor evidence nodes, however these "hidden" nodes are essential to describe the sensor models. For sensitivity analysis, they can be defined by "conditioning" on a set of specific values. Although the estimation error is inversely proportional to the sample size, its true value also depends on the number of nodes (evidences and target) considered and their size (number of sates). Therefore, the estimation error obtained here might be much smaller than its actual value.

4. ALGORITHM CONVERGENCE ANALYSIS

As mentioned above, due to high complexity and possible hybrid nature (mixed variables) of the model, we may have to apply approximate method to estimate $\Pr(\text{Obs} \mid T = j)$ based on stochastic simulation and subsequently compute the posterior probability according to the Bayes rule, namely, (3). To do so, one critical question to be answered is how fast the simulation algorithm converges when estimating the GCM and does it converge to the correct probability. This section derives an analytical convergence rate of the FPM algorithm and estimates the number of simulation trials needed in order to achieve a desirable accuracy level (error bounds).

With (2), let Y_{Oj} be the indicator function for estimating $\Pr(\text{Obs} \mid T = j)$, i.e., $Y_{Oj} = 1$ when "Obs" is realized given $T = j$ in a particular simulation trial and $Y_{Oj} = 0$ otherwise. Then it can be easily shown that when the sample size n is large, the probability distribution of $X_{Oj} \equiv \sum Y_{Oj}/n$ can be approximated by the normal distribution,

$$p(X_{Oj}) \sim N[X_{Oj}; \bar{X}_{Oj}, \sigma^2_{Oj}] \tag{4}$$

where $\bar{X}_{Oj} = P_{Oj} \equiv \Pr(\text{Obs} \mid T = j)$ is the mean and $\sigma^2_{Oj} = \text{var}(X_{oj}) \approx P_{Oj}(1 - P_{Oj})/n$ is the variance of the random variable X_{oj} respectively.

To estimate the probability of correct classification, as shown in (1), we only need to focus on the diagonal elements of the GCM. From (1)–(3), it can be easily shown that,

$$\begin{aligned}\text{GCM}(i,i) &= \Pr(I = i) \sum_{\text{Obs}} \frac{\Pr(\text{Obs} \mid I = i)^2}{\Pr(\text{Obs})} \\ &= \Pr(I = i) \sum_{\text{Obs}} \frac{\sigma^2_{Oi}}{\Pr(\text{Obs})} \left[\frac{\Pr(\text{Obs} \mid I = i)}{\sigma_{Oi}}\right]^2.\end{aligned} \tag{5}$$

Assuming the size of the state space of Obs is relatively large, i.e., $|\text{Obs}| \equiv m \gg 1$, and assuming that $P_{Oi} \ll 1$ and $P_O \equiv \Pr(\text{Obs}) \approx P_{Oi}$,[4] then

$$\text{GCM}(i,i) = \frac{\Pr(I = i)}{n} \sum_{\text{Obs}} \left[\frac{\Pr(\text{Obs} \mid I = i)}{\sigma_{Oi}}\right]^2. \tag{6}$$

[4]This approximation could be poor.

Let $G_{i,i}$ be the indicator function of $\text{GCM}(i,i)$, then the sample mean of $G_{i,i}$ can be approximated by,[5]

$$\bar{G}_{i,i} \approx \frac{\Pr(I = i)}{n} \sum_{\text{Obs}} \left[\frac{X_{Oi}}{\sigma_{Oi}}\right]^2 \tag{7}$$

where $Z \equiv \sum_{\text{Obs}} [X_{Oi}/\sigma_{Oi}]^2$ follows a non-central Chi-square distribution with $\bar{Z} = m + \lambda$ and $\sigma_z^2 = 2(m + 2\lambda)$, where

$$\begin{aligned}\lambda &= \sum_{\text{Obs}} \left[\frac{\bar{X}_{Oi}}{\sigma_{Oi}}\right]^2 = \sum_{\text{Obs}} \left[\frac{\bar{X}^2_{Oi}}{\bar{X}_{Oi}(1 - \bar{X}_{Oi})/n}\right] \\ &= n \sum_{\text{Obs}} \left[\frac{\bar{X}_{Oi}}{(1 - \bar{X}_{Oi})}\right]\end{aligned} \tag{8}$$

In the case when P_{Oj} is approximated by a uniform distribution, $\bar{X}_{Oi} \approx 1/m$, then $\lambda = mn/(m-1) \le mn$, and $\sigma_z^2 = 2(m + 2\lambda) \le 2m(1 + 2n)$. Therefore, with $P_i \equiv \Pr(I = i)$, the variance of $\bar{G}_{i,i}$ is,

$$\sigma^2_{\bar{G}_{i,i}} = (P_i/n)^2 \sigma_z^2 < P_i^2 \frac{2m(1 + 2n)}{n^2}. \tag{9}$$

Finally, assume $\Pr(I = i)$ is approximately uniform, since from (1), $P_{\text{CC}} = (1/k) \sum_{i=1}^{k} \text{GCM}(i,i)$, the variance of the estimate $\hat{P}_{\text{CC}}$ is $\sigma^2_{\text{CC}} = (1/k^2) \sum_{i=1,\ldots,k} \sigma^2_{\bar{G}_{i,i}}$, where k is the state space size of the ID node, then

$$\sigma^2_{\text{CC}} = \frac{1}{k^2} \sum_i (P_i/n)^2 \sigma_z^2 < \frac{2m(1 + 2n)}{k^3 n^2}. \tag{10}$$

With Chebyshev's inequality, for any $\varepsilon > 0$,

$$P_r(|P_{\text{CC}} - \hat{P}_{\text{CC}}| < \varepsilon) \ge 1 - \frac{\sigma^2_{\text{CC}}}{\varepsilon^2}. \tag{11}$$

Equations (10)–(11) provide a performance bound for the absolute error of the average correct classification probability given the observation state space size, the target state space size, and the number of simulation trials. When P_{CC} is normally distributed, then a tighter bound can be obtained as follows,

$$P_r(|P_{\text{CC}} - \hat{P}_{\text{CC}}| < \varepsilon) = \alpha \tag{12}$$

where $\varepsilon = \Phi^{-1}((1 + \alpha)/2)\sigma_{\text{CC}}$, α is the confidence level, and Φ is the CDF of the standard normal distribution.

5. PERFORMANCE EVALUATION

Recall that the main objective of the tool developed in this paper is to evaluate the fusion performance and quantify how much one gains by combining different sensors in a fusion system. Specifically, the system is designed to help a decision maker answer the following questions: (i) What is the probability of correct classification of a given target using a specific sensor individually? (ii) What if a specific set of sensors combined together is used instead? (iii) What is the performance gain by adding another sensor to this set? and (iv) Which sensors provide a better cost/benefit ratio?

[5]Likewise, this could be a rough approximation.

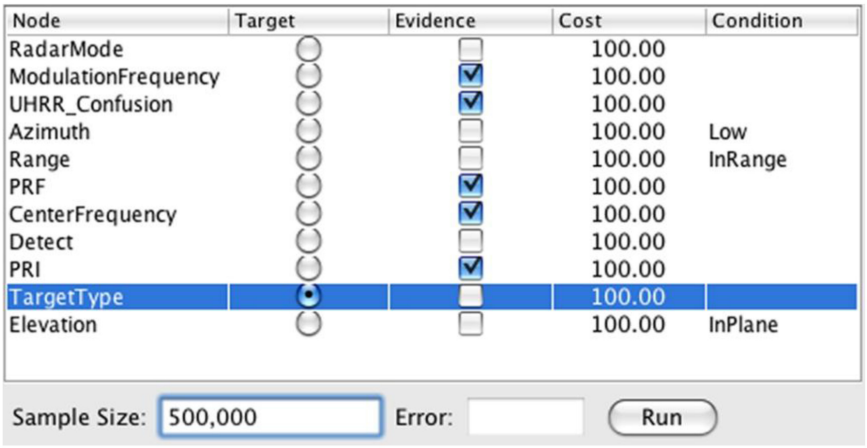

Fig. 3. UnBBayes' evaluation module input panel.

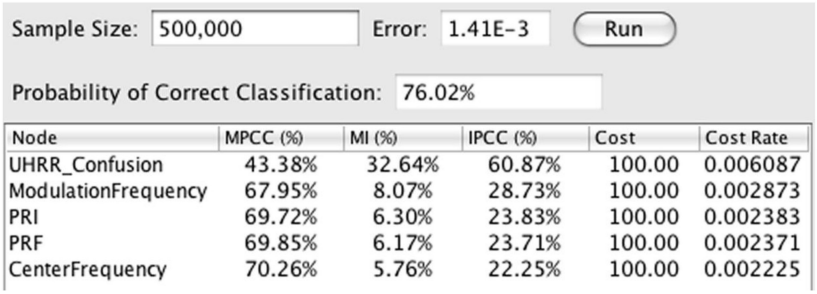

Fig. 4. UnBBayes' evaluation module output panel.

These questions can be answered by using the technical approach described in Sections 3–4. Figure 3 shows the necessary inputs that need to be specified for the FPM evaluation module. The inputs are:

Target node: Select the target node of interest.

Evidence nodes: Choose the evidence nodes, they are the sensor observables in a Multi-Sensor Classification System.

Cost: The cost associated with the evidence node. It is assigned heuristically or based on *a priori* knowledge of the cost of allocating the corresponding sensor resource.

Condition: Choose the state for the conditioning nodes, which represent an optional artificial context for sensitivity analysis. These nodes cannot be target or evidence.

Sample size: The number of trials to be generated from the model. The larger the number the more accurate the result, however the longer it will take to compute.

Error bound: In case the error bound is given, the sample size will be automatically computed based on the analysis given in Section 4.

Figure 4 presents the outputs computed in UnBBayes' evaluation module that can answer such questions. The outputs include:

GCM: The global confusion matrix computed for the selected target node and all the chosen evidence nodes.

Error: As explained in Section 4, the error can be approximately computed by equations (11) and (12).

Probability of Correct Classification (Pcc): The probability of correct classification computed from the GCM considering all evidence nodes.

Marginal PCC (MPCC): The probability of correct classification computed from the GCM given all evidence nodes other than the one presented in the row (see "Node" column).

Marginal Improvement (MI): The probability of correct classification gained by adding the node presented in the row to the rest of other nodes,

$$\text{MI} = \text{PCC} - \text{MPCC}. \tag{13}$$

Individual PCC (IPCC): The probability of correct classification computed from the LCM considering only the evidence presented in the row.

Cost Rate: The individual probability of correct classification over the cost,

$$\text{Cost Ratio} = \frac{\text{IPCC}}{\text{Cost}}. \tag{14}$$

Using the tool and its output just presented we were able to analyze the example model described in Section 2 for air target tracking and classification. Recall that in this example we have three identification modules that represent the evidence nodes. UHRR is an active technique and is basically an imaging technique that will be able to identify features of an airplane and

therefore infer the target type. RESM (represented by nodes "Center Frequency"—CF, "PRF," and "PRI") is an passive technique that can detect a target feature-modulation frequency. RSM (represented by the node "Modulation Frequency"—MF) is an active technique that can observe the characteristics of the target's radar emissions. Based the observed features of the radar signal, the MBID system will infer a radar mode that will in turn be an evidence for a target type.

In the BN model, the target type node has 6 different possible labels: [*B*1, *B*2, *R*1, *R*2, *Neutral*, *Unknown*] with a given prior probability distribution. The feature of UHRR is an identification of target type with conditional probability of a UHRR ID given the true target type. This confusion matrix is indexed by relative target elevation, which we assume is known at the time of the UHRR action. The detailed description of each node and their conditional probability tables were given in [5].

As shown in the BN model, for all identification modules, the observation is dependent on the probability of detection represented by the node "Detected." The detection probability of each target is a function of the relative geometry between the target and the sensor. In reality, the values of the kinematic states need to be assigned dynamically for each target. In the test scenario, we selected the values of three kinematic state nodes, "Range," "Azimuth," and "Elevation," such that the detection probability is approximately 0.95.

Table II shows different sets of evidence nodes used to detect the node "Target Type" using exact and an approximate (with an error strictly lower than 2%) computation. To be concise we did not include all the information computed in UnBBayes in this table, but most of them can be derived from the table. For instance, the MI for the node UHRR in the evidence set UHRR+RSM is 65.36% minus the IPCC of the RSM, which is also, in this case, the MPCC of the node UHRR in this set. So MI for UHRR = 65.36% − 29.27% = 35.69%, while the MI for RSM = 65.36% − 61.90% = 3.46%.

With the information obtained in Table II, the decision maker is able to understand how the system works and which set of sensors work better together by comparing individual performance as well as marginal improvements when more than one sensor is used. For example, with RESM (CF+PRI+PRF) alone, the Pcc is about 33%; with RSM alone, the Pcc is about 29%; and with UHRR alone, the Pcc is about 62%. With the first two together, the Pcc increases to only 43%, while with all three of them, the Pcc increases to over 71%.

Another benefit of using the tool is that the decision maker could analyze the cost/benefit ratio of each sensor resource to determine the best allocation strategy. The cost ratio can also be integrated into an automatic sensor resource management (SRM) algorithm for changing the sensor mode dynamically on a real time basis.

The Pcc is used to evaluate the performance of the model given that the model is available and assumed correct. However, it is not used to judge the accuracy of the model. To test the robustness of the FPM evaluation methodology, we evaluated the same structure model but with different parameter values, meaning we kept the nodes and arcs the same but changed the conditional probability tables (CPT) by adding some noise to them. The goal is to verify that even if the model is somewhat imprecise, we can still apply the evaluation process to obtain a reasonable result. The results in Table III show that the estimated Pcc performance is relatively insensitive to the model uncertainty. Note that the noise level in Table III represents the uncertainty magnitude in the model quantified by the random variations in percentage of the conditional probabilities.

To compare the analytical performance bounds derived in Section 4 and the simulation results, Figs. 5–7 show the relationship between sample size and the absolute estimation error given different observation state space sizes with 99% confidence bounds ($\alpha = 0.99$) predicted by (12). The target state space size is assumed to be fixed ($k = 6$). As can be seen from the figures, the theoretical analysis provides good performance bounds

TABLE II
Classification Performance Prediction with Different Evidence Sets

# Ev. Nodes	Evidence Set	Pcc Exact	Pcc 2% Error
1	RESM (CF)	22.28%	22.88%
	RESM (PRF)	23.73%	24.36%
	RESM (PRI)	23.82%	25.01%
	RSM	28.67%	29.27%
	UHRR	61.02%	61.90%
2	RESM (CF+PRI)	27.85%	29.75%
	RESM (CF+PRF)	27.81%	29.08%
	RESM (PRI+PRF)	28.91%	30.18%
	UHRR+RSM	65.48%	65.36%
3	RESM (CF+PRI+PRF)	31.72%	32.93%
	RSM+RESM (CF+PRI)	38.76%	39.85%
	RSM+RESM (CF+PRF)	38.72%	39.72%
	RSM+RESM (PRI+PRF)	39.67%	40.70%
	UHRR+RESM (CF+PRI)	65.61%	66.15%
	UHRR+RESM (CF+PRF)	65.58%	66.37%
	UHRR+RESM (PRI+PRF)	66.16%	66.39%
4	RSM+RESM (CF+PRI +PRF)	42.14%	43.30%
	UHRR+RESM (CF+PRI+PRF)	67.55%	68.54%
	UHRR+RSM+RESM (CF+PRI)	69.32%	70.14%
	UHRR+RSM+RESM (CF+PRF)	69.31%	69.99%
	UHRR+RSM+RESM (PRI+PRF)	69.78%	70.61%
5	UHRR+RSM+RESM (CF+PRI+PRF)	70.95%	71.72%

TABLE III
Pcc for Models with Different Evidence Nodes

Noise	Pcc	
	Exact	Approximate
0%	70.95%	72.73%
5%	67.82%	69.79%
10%	66.36%	68.37%

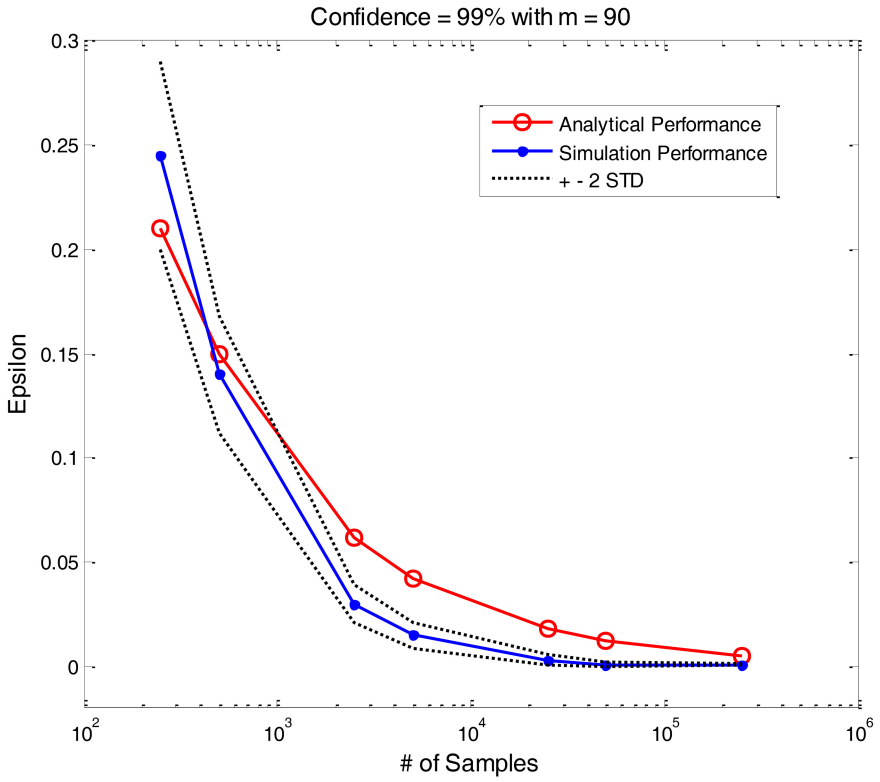

Fig. 5. Absolute error as a function of sample size with $m = 90$.

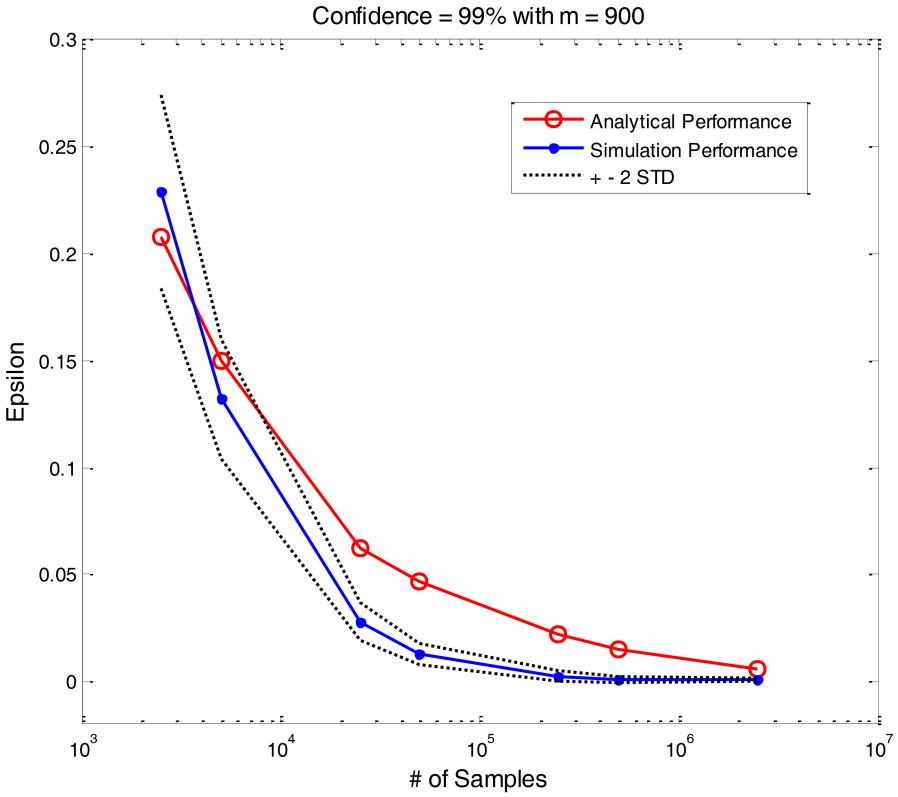

Fig. 6. Absolute error as a function of sample size with $m = 900$.

under different conditions (i.e., various observation state space sizes). The bounds work well particularly when the sample sizes are relatively small or relatively large. However, the analytical bounds are somewhat conservative otherwise due to various approximations employed in the analysis.

Finally, we evaluated the fusion performance of a classification system based on a Bayesian network

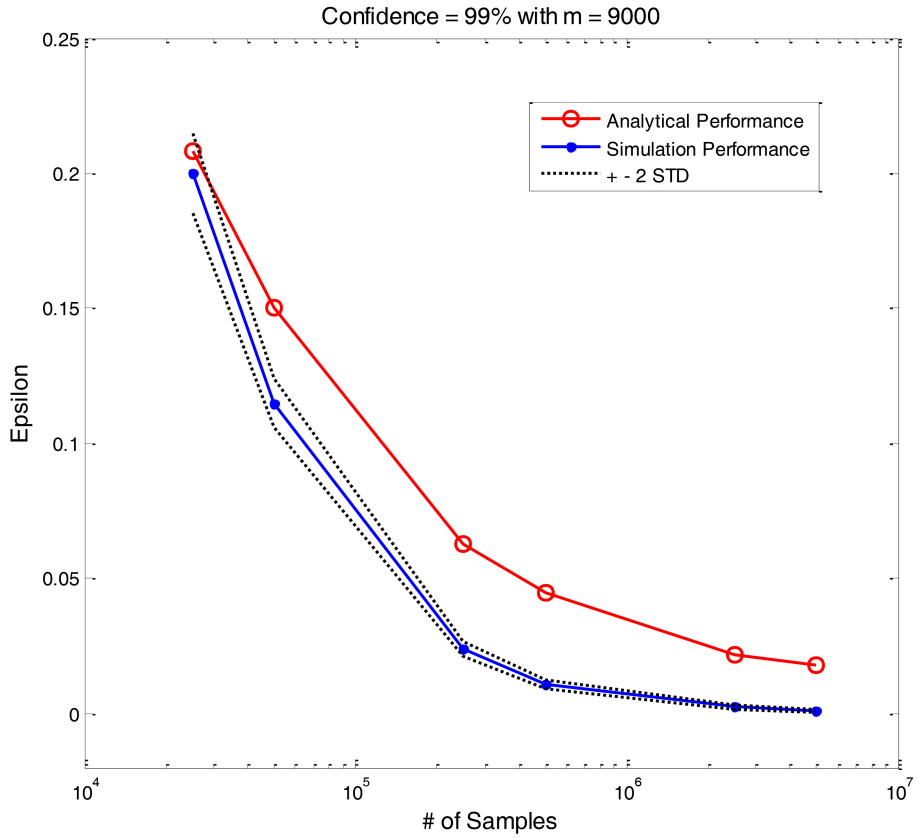

Fig. 7. Absolute error as a function of sample size with $m = 9000$.

model used for Combat ID and threat assessment described in [1] (see Fig. 8). According to [1] the model can be used in a number of ways to infer information about the Hostility, ID Platform and Threat variables.

Table IV shows the Pcc for different target nodes given all evidence nodes or a subset of them, which has the nodes IFF, ATR, ESM, Speed, OnDataLink, and Intelligence. The Pcc values were all computed with a sample size of 2.5M. If the classifications were selected at random the Pcc for Hostility, ID Platform, and Threat would be 16.66%, 12.50%, and 50.00%, respectively, since Hostility has 6 possible states, ID Platform has 8, and Threat has 2 states. Therefore, it can be seen that this model has almost the same performance for the classification of Hostility and Threat as if the classification were being selected at random. Although the Pcc for the ID Platform is slightly better, the decision maker might still consider it low. Note that, however, the correct classification performance is estimated with a single sample observation from each sensor. For multiple observations from different sensors or from a single sensor over multiple sampling times, additional analysis is required. For details, see [21].

With these two models analyzed, we can see the full benefit of using our tool for performance evaluation. We were not only able to detect when a model is useful for classification but also able to detect when it is not as efficient. Furthermore, the same model might not have a good performance for classifying a specific target node, but it might be good for classifying a different one.

6. CONCLUSIONS

In this paper, we present a tool that allows a user to evaluate the classification performance of a multi-sensor fusion system modeled by a Bayesian network. With the Fusion Performance Model (FPM) described in [7] and [8], we developed a new module and integrated it with the free, open-source, and platform independent probabilistic network framework UnBBayes.

We demonstrate the functionalities of the tool with a model-based ID example for air target tracking and classification. We were able to answer questions related to probability of correct classification of a given target using a specific individual sensor resource or a set of resources. We were also able to evaluate the marginal performance gain and cost/benefit ratio of each sensor resource. This tool is very valuable for a decision maker to analyze trade-off between performance and costs and to select proper sensor suites according to requirements and constraints. As far as we could tell, there is no other tool available for evaluating a fusion system performance as described in this paper.

We developed an analytical convergence analysis where we derived theoretical formulae to estimate the convergence rate and predict the number of simulation trials needed in order to achieve a desirable accuracy level (error bounds). We also compared the simulation

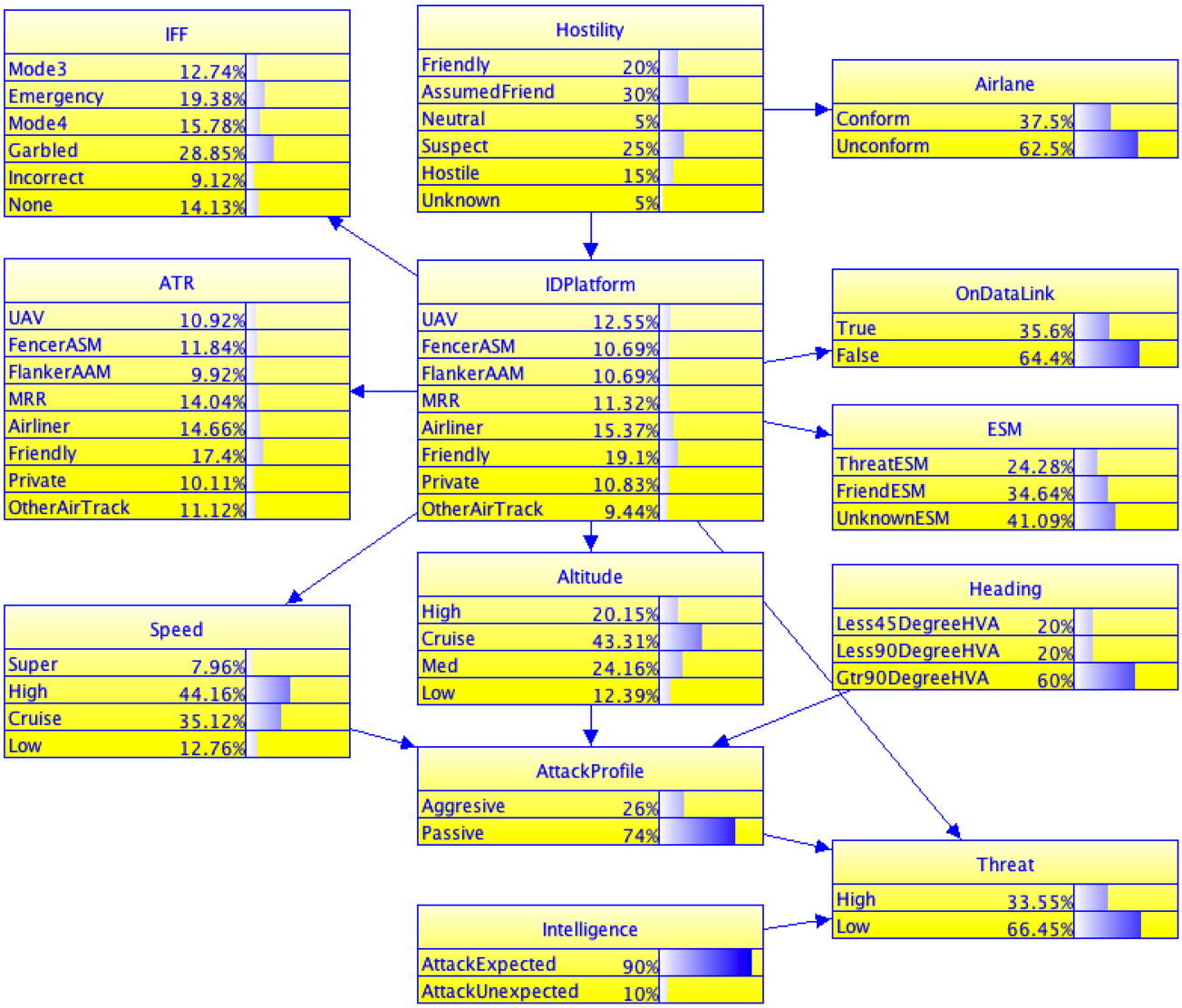

Fig. 8. A simplified model of a Bayesian network used for Combat ID and threat assessment.

results with the analytical ones and we showed that the bounds work well under different conditions.

We also showed that the tool developed is useful to identify the quality of the classification models. Moreover, the performance of the model depends on the node to be classified, i.e., the same model might not have a good performance for classifying a specific target node, but it might be good for classifying a different one.

This research was conducted as part of a larger effort to design an integrated multi-sensor tracking and ID performance evaluation system. A major advantage of using the current approach is the flexibility of modifying the Bayesian models to account for various potential environmental or sensor changes. One important future research direction is to integrate the kinematic tracking module into a combined track/ID performance evaluation system and to extend the system to accommodate for high level fusion. Additionally, we intend to incorporate other efficient analytical or simulation algorithms to improve the computational efficiency of the tool. Finally, since the model itself might be inaccurate due to limited training data or insufficient domain expertise, it is important to take into account the model uncertainty while assessing its performance.

TABLE IV
Pcc for Different Target Nodes Given all Evidence Nodes or a Subset of Them (IFF, ATR, ESM, Speed, OnDataLink, and Intelligence)

Target Node	Pcc Given Evidence Nodes	
	All	Subset
Hostility	38.99%	19.18%
ID Platform	61.35%	48.16%
Threat	65.64%	52.84%

REFERENCES

[1] P. Bladon, P. S. Day, T. Hughes, and P. Stanley
High-level fusion using Bayesian networks: Applications in command and control.
In *Information Fusion for Command Support*, Proceedings RTO-MP-IST-055, Neuilly-sur-Seine, France, 2006.

[2] R. N. Carvalho, M. Ladeira, L. L. Santos, and P. C. Costa
A GUI tool for plausible reasoning in the semantic web using MEBN.
In *Proceedings of the Seventh international Conference on Intelligent Systems Design and Applications*, pp. 381–386; ISDA, IEEE Computer Society, Rio de Janeiro, Brazil, 2007.

[3] R. N. Carvalho, M. Ladeira, L. L. Santos, S. Matsumoto, and P. C. Costa
UnBBayes-MEBN: Comments on implementing a probabilistic ontology tool.
In *Proceedings of the IADIS International Conference on Applied Computing*, Portugal, 2008, pp. 211–218.

[4] R. N. Carvalho, M. Ladeira, L. L. Santos, S. Matsumoto, and P. C. Costa
A GUI tool for plausible reasoning in the semantic web using MEBN.
Book Innovative Applications in Data Mining, DOI: 10.1007/978-3-540-88045-5_2, Springer Berlin/Heidelberg, 2009, pp. 17–45.

[5] KC. Chang and R. Fung
Target identification with Bayesian networks in a multiple hypothesis tracking system.
Optical Engineering, **36** (Mar. 1997), 684–691.

[6] KC. Chang and D. He
Inference with importance sampling for dynamic Bayesian networks.
In *Proceedings of Fusion 2005*, Philadelphia, C2-1, July 2005.

[7] KC. Chang, E. Sivaraman, and M. Liggins
Performance modeling for multisensor tracking and classification.
In *Proceedings of SPIE Defense and Security Symposium*, Vol. #5429, Orlando, FL, Apr. 2004.

[8] KC. Chang, Y. Song, and M. Liggins
Performance modeling for multisensor data fusion.
In *Proceedings of SPIE AeroSense*, Orlando, FL, Apr. 2003.

[9] E. Charniak
Bayesian network without tears.
AI Magazine, 1991, pp. 50–63.

[10] P. C. Costa, M. Ladeira, R. N. Carvalho, L. L. Santos, S. Matsumoto, and K. B. Laskey
A first-order Bayesian tool for probabilistic ontologies.
In *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference*, Menlo Park, CA: The AAAI Press, 2008, pp. 631–636.

[11] R. Fung and KC. Chang
Weighting and integrating evidence for stochastic simulation in Bayesian networks.
In *Proceedings of the 5th Workshop on Uncertainty in AI*, Windsor University, Aug. 1989.

[12] M. Henrion
Propagating uncertainty in Bayesian networks by probabilistic logic sampling.
In *Proceedings of Uncertainty in Artificial Intelligence*, J. Lemmer and L. Kanal (Eds.), North-Holland, Amsterdam, 1986, pp. 317–324.

[13] M. Ladeira, D. C. Silva, M. H. P. Vieira, R. N. Carvalho, M. S. Onishi, and W. T. Silva
Platform independent and open tool for probabilistic networks (in Portuguese, "Ferramenta Aberta e Independente de Plataforma para Redes Probabil´sticas").
In *Proceedings of XXIII Congresso da Sociedade Brasileira de Computação*, Campinas, Brazil, 2003.

[14] S. L. Lauritzen and D. J. Spiegelhalter
Local computations with probabilities on graphical structures and their application in expert systems.
Journal Royal Statistical Society B, 50 (1988).

[15] M. Liggins, D. Hall and J. Llinas (Eds.)
Handbook of Multisensor Data Fusion: Theory and Practice (2nd ed.).
Sept. 2008.

[16] J. Pearl
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.
Morgan Kaufmann Publishers, 1988.

[17] N. Rao
Multisensor fusion under unknown distributions: Finite sample performance guarantees.
Multisensor Fusion, Kluwer Academic Publishers, 2002.

[18] R. D. Shachter
Intelligent probabilistic inference.
In *Proceedings of the Uncertainty in Artificial Intelligence*, L. N. Kanal and J. F. Lemmer (Eds.), North-Holland, Amsterdam, 1986.

[19] R. Shachter, A. D. Favero, and B. D'Ambrosio
Symbolic probabilistic inference: a probabilistic perspective.
In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 1990.

[20] R. Shachter and M. Peot
Simulation approaches to general probabilistic inference on belief networks.
In *Proceedings of the 5th Workshop on Uncertainty in Artificial Intelligence at the Association for the Advancement of Artificial Intelligence*, Windsor University, Aug. 1989.

[21] E. Sivaraman and KC. Chang
Performance evaluation of multi-sensor classification systems.
IEEE Transactions on Aerospace and Electronic Systems, **43** (Oct. 2007), 1265–1281.

Rommel N. Carvalho is now a postdoctoral research associate at the Center of Excellence in Command, Control, Communications, Computing and Intelligence (C4I) at George Mason University (GMU). During the 3 years of his Ph.D., he was a graduate research assistant in the Department of Systems Engineering and Operations Research at GMU, VA. He received his Master in Computer Science and his Bachelor of Computer Science from University of Bras´lia, DF, Brazil, in 2008 and 2003, respectively. He is an Artificial Intelligence (AI) researcher with focus on uncertainty in the Semantic Web using Bayesian Inference, Data Mining, Software Engineering and Java Programming. Awarded programmer with experience in implementation of Bayesian Network systems (UnBBayes), Multi-Entity Bayesian Network and Probabilistic Web Ontology Language (PR-OWL), and various web-based applications. Rommel N. Carvalho has been working for the Brazilian Government at the Office of the Comptroller General (CGU) as an Information Technology (IT) expert since 2005. He has also done extensive research on fraud detection and prevention for the Brazilian Government and situation awareness for the U.S. Navy. In most of the systems he helped develop he was the project manager, which gave him the experience necessary to get the Project Management Professional (PMP) certificate. During his Ph.D., he has published over 15 papers, among conference and workshop papers, book chapters, journal papers, and workshop proceedings.

Kuo-Chu Chang received the M.S. and Ph.D. degrees in electrical engineering from the University of Connecticut in 1983 and 1986 respectively.

From 1983 to 1992, he was a senior research scientist in Advanced Decision Systems (ADS) division, Booz-Allen & Hamilton, Mountain View, CA. In 1992, he joined the Systems Engineering and Operations Research Department, George Mason University where he is currently a professor. His research interests include estimation theory, optimization, signal processing, and multisensor data fusion. He is particularly interested in applying unconventional techniques in the conventional decision and control systems. He has more than 25 years of industrial and academic experience and published more than one hundred and fifty papers in the areas of multitarget tracking, distributed sensor fusion, and Bayesian Networks technologies. He was an associate editor on Tracking/Navigation Systems from 1993 to 1996 and on Large Scale Systems from 1996 to 2006 for *IEEE Transactions on Aerospace and Electronic Systems*. He was also an associate editor of *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, from 2002 to 2007.

Dr. Chang is a member of Etta Kappa Nu and Tau Beta Pi.