Multi-sensor Data Fusion: From Algorithm and Architecture Design to Applications
# Contents

<table>
<thead>
<tr>
<th>List of Figures</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributors</td>
<td>xi</td>
</tr>
<tr>
<td>1 Belief function based multi-sensor multi-target classification solution</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Basics on belief function theory</td>
<td>3</td>
</tr>
<tr>
<td>1.2.1 Knowledge representation</td>
<td>3</td>
</tr>
<tr>
<td>1.2.2 Knowledge combination</td>
<td>3</td>
</tr>
<tr>
<td>1.2.2.1 Conjunctive rule of combination</td>
<td>4</td>
</tr>
<tr>
<td>1.2.2.2 Disjunctive rule of combination</td>
<td>4</td>
</tr>
<tr>
<td>1.2.3 Decision making</td>
<td>4</td>
</tr>
<tr>
<td>1.2.4 Generalized Bayes Theorem</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Local tracking solution</td>
<td>5</td>
</tr>
<tr>
<td>1.3.1 Targets evolution model</td>
<td>5</td>
</tr>
<tr>
<td>1.3.2 Local tracking algorithm</td>
<td>6</td>
</tr>
<tr>
<td>1.4 Local classification</td>
<td>6</td>
</tr>
<tr>
<td>1.4.1 Behaviors likelihoods calculation</td>
<td>7</td>
</tr>
<tr>
<td>1.4.2 Bayesian classifier</td>
<td>7</td>
</tr>
<tr>
<td>1.4.3 Credal classifier</td>
<td>8</td>
</tr>
<tr>
<td>1.5 Global classification</td>
<td>9</td>
</tr>
<tr>
<td>1.5.1 Track-to-track association</td>
<td>9</td>
</tr>
<tr>
<td>1.5.2 Local classifications fusion</td>
<td>11</td>
</tr>
<tr>
<td>1.5.2.1 Bayesian rules of combination</td>
<td>11</td>
</tr>
<tr>
<td>1.5.2.2 Credal rules of combination</td>
<td>11</td>
</tr>
<tr>
<td>1.6 Maritime targets simulation example</td>
<td>11</td>
</tr>
<tr>
<td>1.6.1 Description</td>
<td>12</td>
</tr>
<tr>
<td>1.6.2 Simulation and results</td>
<td>14</td>
</tr>
<tr>
<td>1.6.3 Assignment simulation</td>
<td>14</td>
</tr>
<tr>
<td>1.6.4 Local classification results</td>
<td>15</td>
</tr>
<tr>
<td>1.6.5 Global classification results</td>
<td>16</td>
</tr>
<tr>
<td>1.7 Conclusion</td>
<td>18</td>
</tr>
</tbody>
</table>

Bibliography 21
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Multi-sensor classification approach flowchart</td>
<td>9</td>
</tr>
<tr>
<td>1.2</td>
<td>Track-to-track assignment test</td>
<td>14</td>
</tr>
<tr>
<td>1.3</td>
<td>Multi-target scenario</td>
<td>15</td>
</tr>
<tr>
<td>1.4</td>
<td>Local classifications with a reliable sensor</td>
<td>16</td>
</tr>
<tr>
<td>1.5</td>
<td>Local classifications with an unreliable sensor</td>
<td>16</td>
</tr>
<tr>
<td>1.6</td>
<td>Bayesian and credal local classifications with different levels of uncertainty</td>
<td>17</td>
</tr>
<tr>
<td>1.7</td>
<td>Global classification results using different rules of combination</td>
<td>17</td>
</tr>
<tr>
<td>1.8</td>
<td>Multi-target scenario</td>
<td>18</td>
</tr>
</tbody>
</table>
Contributors

Samir Hachour
Univ. Lille Nord de France, UArtois,
EA 3926 LGI2A
Béthune, France

François Delmotte
Univ. Lille Nord de France, UArtois,
EA 3926 LGI2A

David Mercier
Univ. Lille Nord de France, UArtois,
EA 3926 LGI2A
Béthune, France
Belief function based multi-sensor multi-target classification solution

CONTENTS

1.1 Introduction ................................................................................. 2
1.2 Basics on belief function theory ............................................. 3
  1.2.1 Knowledge representation ................................................. 3
  1.2.2 Knowledge combination .................................................... 3
    1.2.2.1 Conjunctive rule of combination ....................... 4
    1.2.2.2 Disjunctive rule of combination ....................... 4
  1.2.3 Decision making ................................................................. 4
  1.2.4 Generalized Bayes Theorem ............................................. 4
1.3 Local tracking solution .............................................................. 5
  1.3.1 Targets evolution model .................................................... 5
  1.3.2 Local tracking algorithm ................................................... 6
1.4 Local classification ................................................................. 6
  1.4.1 Behaviors likelihoods calculation ....................................... 7
  1.4.2 Bayesian classifier ............................................................ 7
  1.4.3 Credal classifier ................................................................. 8
1.5 Global classification ............................................................... 9
  1.5.1 Track-to-track association ............................................... 9
  1.5.2 Local classifications fusion .............................................. 11
    1.5.2.1 Bayesian rules of combination .......................... 11
    1.5.2.2 Credal rules of combination .......................... 11
1.6 Maritime targets simulation example ....................................... 11
  1.6.1 Description .................................................................... 12
  1.6.2 Simulation and results ..................................................... 14
  1.6.3 Assignment simulation .................................................... 14
  1.6.4 Local classification results ............................................. 15
  1.6.5 Global classification results ........................................... 16
1.7 Conclusion ............................................................................... 18
1.1 Introduction

This chapter proposes a multi-sensor multi-target classification architecture. Each sensor is supposed to locally track a set of randomly appearing and disappearing targets using Interacting Multiple Models (IMM) algorithms [1, 2]. Based on targets estimated kinematic data, a local classification step is performed, it aims to recognize the classes of targets (e.g. go-fast boat, military boat, cargo, etc.) based on their behaviors. A special case of classification is considered, a case where classes are nested: a go-fast boat can behave as a military boat or cargo, a military boat can evolve as slowly as a cargo but can not go as fast as a go-fast boat and finally a cargo can not evolve neither as fast as a military boat nor as fast as a go-fast boat. On such case of classification, belief function based classifiers are shown to perform better than classical Bayesian classifiers, namely in [20] for a single target classification and in [11] for multiple target classification. Bayesian and credal classifiers are reminded in this chapter, they are locally processed at each sensor level. Classification results are shown to be deteriorated when it comes to manage high sensor uncertainties. In order to enhance the classification results of the targets, a multi-sensor approach is needed [21]. The question is: how to fuse data coming from multiple sensors?

This chapter focuses on the multi-sensor fusion center, and provides two main contributions. The first contribution is about a parameterless track-to-track assignment solution where already existing solutions are mostly parameter-dependent [12, 2, 8]. The track-to-track solution aims to get a consensus on the tracked targets, it is shown in [10] that the proposed solution has the advantage of getting over parameters training. The track-to-track solution receives targets estimates provided by sensors’ IMM at the entry and matches them in order to get a consensus on the commonly tracked targets. Local classification results of the commonly tracked targets are then fused in order to get a better classification performance.

The second contribution of this chapter concerns the way local classification results are fused in order to obtain the best classification results. Regarding this issue, some credal and Bayesian fusion rules are tested. It is shown, for the considered classification case (nested classes), that the credal disjunctive rule of combination provides the best results. Notice that in the Bayesian framework, none equivalence to the credal disjunctive rule of combination exists, some other Bayesian rules of combination, provided in [4], are instead tested. The comparison result was presented in [9] and extended in this chapter.

Section 1.2 of this chapter provides some basics on belief function theory. Section 1.3 gives some information on the adopted tracking solution. Section 1.4 describes the Bayesian and the credal classifiers which are locally processed at each sensor level. Section 1.5 highlights the two main contributions of this
chapter, namely: the parameterless track-to-track solution and the motivation of using the disjunctive credal rule of combination, which are processed at the fusion center. Finally, a nested classes simulation example is provided in section 1.6. It is about a piracy surveillance where the proposed fusion solution is shown to be efficient, especially, in the case of high sensors uncertainties.

\section{Basics on belief function theory}

Belief functions are often referred to as Dempster-Shafer theory with respect to Arthur Dempster and Glenn Shafer who introduced the formalism \cite{7, 14}, the theory is also known by the credal formalism. Belief functions in the works of Smets are gathered in a framework called the Transferable Belief Model (TBM) \cite{16}. Some basic notions on belief function theory are provided in this section.

\subsection{Knowledge representation}

Knowledge in the credal theory is expressed on a finite set of mutually exclusive and exhaustive hypotheses $H = \{h_1, h_2, ..., h_N\}$ which is referred to as the frame of discernment. Belief can be given to the hypotheses as singletons or sets of hypotheses $A \subseteq H$. This is the major advantage over the probability measures who require a precise knowledge about the singletons only. The uncertainty is represented by a mass function (basic belief assignment) $m$ expressed on $2^H$ which satisfies:

$$\sum_{A \subseteq H} m(A) = 1.$$  \hfill (1.1)

Subsets $A \subseteq H$ with $m(A) > 0$ are referred to as focal elements.

The plausibility function $Pl$ can also be used to model knowledge, it is in one to one correspondence with $m$, it is given as follows:

$$Pl(A) = \sum_{A \cap B \neq \emptyset} m(B).$$  \hfill (1.2)

Some other functions are used to express knowledge, they can be found in \cite{17}.

\subsection{Knowledge combination}

Information is often propagated in form of mass functions. When more than one mass function are expressed on the same frame of discernment, they can be combined. Several combination rules are presented in \cite{17}. In this chapter, the conjunctive and disjunctive rules of combination are considered.
1.2.2.1 Conjunctive rule of combination

The conjunctive rule of combination assumes that the masses to combine are provided from independent and reliable sources. For two mass functions \( m_{s_1} \) and \( m_{s_2} \) provided by sources \( s_1 \) and \( s_2 \), the combination rule is given by:

\[
m_{s_1 \sqcap s_2}(A) = (m_{s_1} \sqcap m_{s_2})(A) = \sum_{A_1 \cap A_2 = A} m_{s_1}(A_1)m_{s_2}(A_2). \tag{1.3}
\]

A normalized version of the formula in (1.3) exists, it is referred to as Dempster’s rule of combination, and is defined by:

\[
m_{s_1 \sqcap s_2}(A) = \frac{\sum_{A_1 \cap A_2 = A} m_{s_1}(A_1)m_{s_2}(A_2)}{1 - \sum_{A_1 \cap A_2 = \emptyset} m_{s_1}(A_1)m_{s_2}(A_2)}. \tag{1.4}
\]

1.2.2.2 Disjunctive rule of combination

The disjunctive rule of combination supposes that at least one of the sources is reliable. Disjunctive combination of two mass functions \( m_{s_1} \) and \( m_{s_2} \) which are provided by sources \( s_1 \) and \( s_2 \), is defined as follows:

\[
m_{s_1 \cup s_2}(A) = \sum_{A_1, A_2 | A_1 \cup A_2 = A} m_{s_1}(A_1)m_{s_2}(A_2). \tag{1.5}
\]

1.2.3 Decision making

When necessary, probability measures can be calculated from mass functions. This refers to the pignistic transformation [18]. The pignistic probability \( \text{BetP} \) over hypothesis \( h_i \in H \) is defined by:

\[
\text{BetP}([h_i]) = \sum_{h_i \in A} \frac{m(A)}{A \mid (1 - m(\emptyset))}. \tag{1.6}
\]

1.2.4 Generalized Bayes Theorem

Based on the idea of the well known Bayesian Theorem, Smets introduces the Generalized Bayesian Theorem (GBT) [16], which requires two steps. Based on likelihoods \( l(h_i | z) = P(l(h_i | z)) \), a conditional mass function \( m(A | z) \) for each \( A \subseteq H \) is firstly calculated as follows:

\[
m(A | z) = \prod_{h_i \in A} l(h_i | z) \prod_{h_i \in \bar{A}} (1 - l(h_i | z)). \tag{1.7}
\]
The second step of the GBT consists in updating the result with priors, as with the classical Bayes Theorem. This is done using the conjunctive rule (1.4). If no prior knowledge is available, an \textit{a priori} vacuous belief function is considered ($m(H) = 1$).

### 1.3 Local tracking solution

This section explains the idea behind multi-target tracking problem and gives some information about the adopted solution to estimate targets trajectories.

#### 1.3.1 Targets evolution model

A Jump Markov Chain Model (JMCM) is often used to represent the evolution of a maneuvering target. An example of a JMCM is given by equation (1.8), it illustrates the evolution of a given target $t$ being in evolution model $m_l$, with $l = \{1, ..., r\}$ and $r$ being the number of possible known maneuvering models.

\[
x_k^t = F x_{k-1}^t + G u_k^t(m_l) + w_k^t, \tag{1.8}
\]

where, $x_k^t \in \mathbb{R}^p$ represents the $t^{th}$ target state vector at time $k$, with $F$ being the $(p \times p)$ state matrix and $u_k^t$ represents the $t^{th}$ target deterministic input, which represents simply a known acceleration mode $m_l$. The parameter $w_k^t$ represents the state Gaussian noise with covariance matrix $Q$. The input matrix is denoted $G$. Detailed information about the adopted models can be found in [3, 22, 15]. For simplicity, the measurements are taken according to a linear model given as follows:

\[
z_k^j = H x_k + v_k, \tag{1.9}
\]

where, $z_k^j \in \mathbb{R}^q$ is the $j^{th}$ received observation at time $k$, with $j \in \{1, 2, ..., m\}$. The observation matrix of dimension $(q \times p)$ is noted $H$ and $v_k$ represents the measurement error, it is considered as a Gaussian noise with zero mean value and covariance matrix $R$. The set of measurements taken by the sensor $i$ at time $k$ is noted $Z_i = \{z_{k1}^i, z_{k2}^i, ..., z_{km_i}^i\}$, with $i \in \{1, 2, ..., S\}$ where $S$ is the number of sensors. Estimation of the state vector in equation (1.8) comes to calculate the following probability density function:

\[
p(x_k^t | u_k^t(m_1, ..., r), z_{1, ..., k}^t), \; t = 1, ..., n, \tag{1.10}
\]

where, $z_{1, ..., k}^t$ represents the cumulative measurement for the target $t$ until time step $k$. 

---

*Belief function based multi-sensor multi-target classification solution*
1.3.2 Local tracking algorithm

In default of being able to analytically calculate the complex multi-modal density in equation (1.10), it is estimated using Interacting Multiple Model (IMM) algorithms. A set of IMMs are performed by each sensor, each IMM tracks a given target. Estimation is performed through two main steps: prediction and update steps. The local tracking algorithm processed at each sensor level is summarized in Algorithm 1.

Algorithm 1 Local tracking algorithm.

Require: Measurements: \( z_j, \ j = \{1, 2, ..., m\} \).
Ensure: State estimates: \( \hat{x}_t, \ t = \{1, 2, ..., n\} \).

1. State vectors \( \hat{x}_t \) and measurements \( \tilde{z}_t, \ t = \{1, 2, ..., n\} \) prediction using IMMs prediction step.
2. Reception of the real measurements \( z_j, \ j = \{1, 2, ..., m\} \):
3. Assignment of the reals measurement \( z_j, \ j = \{1, 2, ..., m\} \) to predicted ones using the matching algorithm described in section 1.5.1:
   - Measurements are used to update the estimations if they are assigned to known targets.
   - Measurements which are not assigned to any known target are used to initiate new targets.
   - Known targets that do not receive any measurement are considered as non-detected.
4. State estimates \( \hat{x}_t, \ t = \{1, 2, ..., n\} \) update, using IMMs update step.

At this level each sensor would have estimated the state of \( n_i, \ i \in \{1, 2, ..., S\} \) targets and each IMM among the \( n_i \) ones would have calculated \( r \) evolution modes probabilities \( \mu_l \) and likelihoods \( \lambda_l \) with \( l = \{1, 2, ..., r\} \). These data are used in the classification step.

1.4 Local classification

It is considered that IMM algorithms contain an exhaustive list of all the targets possible evolution models. The list of models is given by:

\[
M = [m_1, m_2, ..., m_r],
\]  

(1.11)
Belief function based multi-sensor multi-target classification solution

where \( r \) represents the total number of models.

An \textit{a priori} knowledge on targets behaviors allows the clustering of the \( r \) different models in \( M \) to be made. The set of possible behaviors can be defined by: \( B = [b_1, b_2, ..., b_{nb}] \), where \( nb \) is the number of behaviors. The set of models belonging to the behavior \( b_i \), for example, is defined by \( M_{b_i} \subseteq M \), with \( i = 1, ..., nb \). The number of models in \( M_{b_i} \) is noted by \( r_{b_i} \). Once defined, behaviors likelihoods \( l_{b_i} \) are calculated from the different models likelihoods \( \lambda_j \) and probabilities \( \mu_j \), with \( j = 1, ..., r \) which are provided by the IMMs update steps.

### 1.4.1 Behaviors likelihoods calculation

Calculation of the behaviors likelihoods is performed as follows:

\[
  l(b_i) = \sum_{j : m_j \in M_{b_i}} \mu'_j \lambda_j, \quad i = 1, ..., nb, \tag{1.12}
\]

with:

\[
  \mu'_j = \frac{\mu_j}{\sum_{j : m_j \in M_{b_i}} \mu_j}, \quad j = 1, ..., r_{b_i}. \tag{1.13}
\]

In order to obtain the classes probabilities or pignistic probabilities, the calculated behaviors likelihoods, can either be processed by a Bayesian or credal classifiers, respectively.

### 1.4.2 Bayesian classifier

Algorithm 2 describes the principal steps processed by the Bayesian classifier.

**Algorithm 2** Bayesian classifier.

**Require:** Classes \textit{a priori} probabilities \( P(c_i | z_1, ..., z_{k-1}) \) and behaviors likelihoods \( l_{b_i} \), where \( i = 1, ..., nc \) and \( j = 1, ..., nb \).

**Ensure:** Classes \textit{a posteriori} probabilities \( P(c_i | z_1, ..., z_{k-1}) \).

1. Behaviors to classes likelihoods:

\[
  l(C) = M \times l(B), \tag{1.14}
\]

where \( M \) is a matrix representing the relations between behaviors and classes, \( l(B) \) represents likelihoods of the behaviors and \( l(C) \) represents likelihoods of the classes.

2. Classes a posteriori probabilities calculation:

\[
  P(c_i | z_1, ..., z_k) = \frac{l(c_i)}{\sum_{j=1}^{S} l(c_j)P(c_j | z_1, ..., z_{k-1})} P(c_i | z_1, ..., z_{k-1}), \tag{1.15}
\]
At each time step, each sensor $i$ returns a set $\Phi_i = \{\phi_{1i}, \phi_{2i}, ..., \phi_{ni}\}$ of probabilities, where $i = 1, ..., S$ with $S$ the number of sensors and $\phi_{ki}$ represents the probability distribution concerning the classification of target $t$.

### 1.4.3 Credal classifier

The credal classifier is executed on two main steps. First, behaviors mass function is computed, it is then transformed into classes mass function and a decision is made using the pignistic transformation.

**Algorithm 3 Credal classifier.**

**Require:** Classes a priori mass function $m^C_k$ and behaviors likelihoods $l(B)$.

**Ensure:** Classes a posteriori pignistic probabilities $Bet_{PC}(c_i)$.

1. Behaviors mass function calculation using the GBT:

$$m_k(D) = \prod_{b_i \in D} l(b_i) \prod_{b_i \notin D} (1 - l(b_i)),$$

where $D \subseteq B$.

2. Projection of the mass function into the classes space $C = \{c_1, c_2, ..., c_{nc}\}$ with $nc$ representing the number of classes.

$$m^C_k = \tilde{M} \times m^B_k,$$

where $\tilde{M}$ is a matrix expressing the relations between behaviors and classes, it contains the conditional masses $m(A|D)$, with $A \subseteq C$ and $D \subseteq B$.

3. Recursive mass functions combination: $m_k$ and $m_{k-1}$ are combined using the conjunctive rule [16]:

$$m^C_k(D) = \sum_{D_1, D_2 | (D_1 \cap D_2) = D} m^C_k(D_1)m^C_{k-1}(D_2),$$

where the initial belief $m^C_0$ being a vacuous mass function [19].

4. In order to make a decision, classes mass function $m^C_k$ is transformed into pignistic probabilities using equation (1.6).

For a global credal classification purpose, the local decision making (transformation into pignistic probabilities) can be avoided and the local mass functions are used to perform a global classification. The set of mass functions given by a sensor $i$ at time $k$ is denoted $M_i = \{m^1_k, m^2_k, ..., m^{ni}_k\}$, where $i = 1, ..., S$ and $S$ is the number of sensors.
1.5 Global classification

Figure 1.1 illustrates the complete multi-sensor multi-target algorithm. Each sensor \(i\) among the \(S\) designed ones makes a set of estimated state vectors \(\hat{X}_i = \{\hat{x}^1_i, \hat{x}^2_i, \ldots, \hat{x}^{n_i}_i\}\) basing on the set of taken measurements \(Z_i\) at time \(k\), by performing the described local tracking algorithm. Using the local classification algorithm, the sensor provides a set of mass functions \(M_i = \{m^1_i, m^2_i, \ldots, m^{n_i}_i\}\) for the credal classification and a set of probability distributions \(\Phi_i = \{\phi^1_i, \phi^2_i, \ldots, \phi^{n_i}_i\}\) provided by the Bayesian classifiers.

The idea behind the global classification is firstly, to get a consensus about the commonly tracked targets through a track-to-track assignment (association) solution, and second fuse their local classifications.

1.5.1 Track-to-track association

The box "track-to-track association" in figure 1.1 aims to establish a matching between sensors estimates sets \(\hat{X}_1, \ldots, \hat{X}_S\) elements in order to recognize in which order the local classification results have to be combined. The matching step is ensured by a newly proposed parameterless assignment method:

1. Plausibilities calculation: let \(r_{t,\ell} \in \{0, 1\}\) be the relation that \(\hat{x}^t_i\) is associated or not with \(\hat{x}^\ell_j\) \((r_{t,\ell} = 1\) means that target \(t\) estimated by sensor \(i\) corresponds to target \(\ell\) estimated by sensor \(j\), \(r_{t,\ell} = 0\) otherwise).

For each estimated target \(t \in \{1, \ldots, n_i\}\) given by a sensor \(i\), a plausibility function \(P_{\ell t}\) is built on the set \(\hat{X}^{\ell,\ast}_j = \{\hat{x}^1_j, \ldots, \hat{x}^{n_j}_j, \ast_j\}\)
of sensor $j$’s known targets. Element $\star_t$ represents the hypothesis that target $t$ is not known by sensor $j$. The plausibility $P_l$ is defined by:

$$P_l(\{\ell\}) = G_{\ell,t}, \quad \forall \ell \in \{1, \ldots, n_j\},$$

(1.19)

where $G_{\ell,t}$ is a likelihood measure calculated as in [3].

Plausibility $P_l(\{\ell\})$ represents the amount of belief supporting the association of $\hat{x}_t^i$ and $\hat{x}_\ell^j$.

The maximum plausibility that target with state $\hat{x}_t^i$ will be associated to one of the $n_j$ already known targets by sensors $j$ corresponds to $\max_{\ell=1,\ldots,n_j} (P_l(\{\ell\})) \leq 1$. This maximum can be lower than one, in particular if the frame of discernment formed by the set of known targets is not exhaustive. Indeed, a target with state $\hat{x}_t^i$ can correspond to a new object ($\star_t$) for sensor $j$. The plausibility of this event is thus defined by:

$$P_l(\{\star_t\}) = 1 - \max_{\ell=1,\ldots,n_j} (P_l(\{\ell\})).$$

(1.20)

2. Mass functions calculation: once a complete plausibility function $P_l$ is calculated on an exhaustive closed-world $\hat{X}_j \cup \{\star_t\}$, a corresponding mass function denoted by $m_t$ is obtained by a direct application of the Generalized Bayesian Theorem (GBT)[16, 6], recalled here for convenience:

$$m_t(A) = \prod_{\ell \in A} P_l(\{\ell\}) \prod_{\ell \notin A} (1 - P_l(\{\ell\})), \quad \forall A \subseteq \hat{X}_j^\star.$$

(1.21)

3. Assignment decision making: first, mass functions $m_t$ are transformed into pignistic probabilities $BetP_t$ then the best assignment relation is chosen as the one maximizing the following criterion:

$$\max \sum_{\ell,t} BetP_t(\{\ell\}) r_{\ell,t}, \quad \ell = \{1, \ldots, n_i + n_j\}, t = \{1, \ldots, n_j\}.$$

(1.22)

with respect to the following constraints:

$$\sum_{\ell} r_{\ell,t} \leq 1, \quad \sum_{t} r_{\ell,t} = 1,$$

(1.23)

$$r_{\ell,t} \in \{0,1\}, \forall \ell \in \{1, \ldots, n_i + n_j\}, \forall t \in \{1, \ldots, n_j\}$$

(1.24)

The constraint at left in (1.23) means that sensor $i$’s estimation of a given target can be assigned to sensor $j$’s known target, if not, it is considered as a new target for sensor $j$. The constraint at right in (1.23) means that a target known by sensor $j$ can be matched
with a target of sensor $i$. If the target is not known by sensor $i$, it is assigned to the extraneous element ($*$).

As in Denœux et al.’s [8] and GNN approaches, this problem can be solved using the Hungarian or Munkres algorithms [5].

Once a consensus is reached between sensors about the commonly tracked targets, their local classifications are fused using different Bayesian and credal rules of combination.

### 1.5.2 Local classifications fusion

The following Bayesian and credal fusion rules are used to fuse local classifications probabilities and mass functions respectively.

#### 1.5.2.1 Bayesian rules of combination

Some Bayesian rules of combination are tested in order to obtain the best global classification result:

- Conjunctive rule of combination:

$$P_{s_1 \cap s_2}(c_j) = \frac{P_{s_1}(c_j)P_{s_2}(c_j)}{\sum_{c_i \in C} P_{s_1}(c_i)P_{s_2}(c_i)}, \quad (1.25)$$

- Other Bayesian combinations:

$$P_{s_1, s_2}(c_j) = \Psi(P_{s_1}(c_j), P_{s_2}(c_j)), \quad (1.26)$$

Notice that no disjunctive rule of combination exists in the Bayesian framework, therefore, other combination rules are tested. As it was reported in [4], operator $\Psi$ can represent the average, min, max, etc.

#### 1.5.2.2 Credal rules of combination

Local credal classification results are fused using two different rules:

- Conjunctive rule of combination given by equation (1.4).

- Disjunctive rule of combination given by equation (1.5).

The presented theoretical knowledge is highlighted through the following simulation example.

### 1.6 Maritime targets simulation example
12 Multi-sensor Data Fusion: From Algorithm and Architecture Design to Applications

1.6.1 Description

A maritime targets classification scenario is considered. The simulation aims to identify targets types (cargo, military boat, go-fast boat, etc.) where the classification is based on the complexity of the performed maneuvers. A set $B = \{b_1, b_2, b_3\}$ of three behaviors are defined basing on targets maneuvering capacities:

- **Behavior 1 ($b_1$):** behavior of targets having a low maneuvering capacities (e.g. cargo).
- **Behavior 2 ($b_2$):** behavior of targets having a medium maneuvering capacities (e.g. military boat).
- **Behavior 3 ($b_3$):** behavior of targets having a high maneuvering capacities (e.g. go-fast boat).

Where the set of targets possible classes $C = \{c_1, c_2, c_3\}$ corresponds to $C = \{\text{Cargo, Military boat, Go-fast boat}\}$.

The state vector of a given target is represented by $x = [x \ \dot{x} \ y \ \dot{y}]$, it represents the position and the velocity on $(x, y)$ directions. The state vector of each target evolves following the model in equation (1.8), with a state matrix $F$ given by:

$$
F = \begin{bmatrix}
1 & \Delta T & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & \Delta T \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

where $\Delta T$ is the sampling time.

The deterministic input vector $u(m_l) = [a_x \ a_y]^T$ in equation (1.8) represents targets different acceleration modes. For example $u(m_1) = [0 \ 0]^T$ represents a constant velocity mode. The differences in the acceleration capabilities allow the classification to be made.

In the performed simulation, each target’s IMM is composed of 13 evolution models according to the different maneuvers which can be made in $x$ and $y$ directions (more details can be found in [13]). The different evolution models are distributed over the three possible targets behaviors, as follows:

- **$M_{b_1}$:** $[m_1]:$ models belonging to the behavior $b_1$.
- **$M_{b_2}$:** $[m_1, ..., m_5]:$ models belonging to the behavior $b_2$.
- **$M_{b_3}$:** $[m_1, ..., m_{13}]:$ models belonging to the behavior $b_3$.

Once behaviors are defined, their likelihoods are calculated as illustrated by equation (1.12). Knowledge on likelihoods space $B$ is then projected to classes space $C$ according to the following relations.

- **Relation 1:** target in behavior 1 can correspond to cargo, military or go-fast boats. All of them can evolve with a constant velocity. This relation can be written as: $b_1 = \{c_1, c_2, c_3\}$. 

Belief function based multi-sensor multi-target classification solution

• Relation 2: target in behavior 2, which has performed a medium maneuver, may correspond to a military or go-fast boat only. Cargos are supposed unable to perform any maneuver. This relation can be written as: \( b_2 = \{c_2, c_3\} \).

• Relation 3: target in behavior 3, which has performed a sharp maneuver, can only be a go-fast boat, because cargos and military boats can not perform sharp maneuvers. This relation can be written as: \( b_3 = \{c_3\} \).

Behaviors knowledge is transferred to classes knowledge using equation (1.14) for the Bayesian classifier, where matrix \( M \) is given by:

\[
M = \begin{bmatrix}
1/3 & 0 & 0 \\
1/3 & 1/2 & 0 \\
1/3 & 1/2 & 1
\end{bmatrix},
\]

which corresponds to the following conditions:

• if \( P(b_1) = 1 \Rightarrow P(c_1|b_1) = \frac{1}{3} \), \( P(c_2|b_1) = \frac{1}{3} \), \( P(c_3|b_1) = \frac{1}{3} \).

• If \( P(b_2) = 1 \Rightarrow P(c_1|b_2) = 0 \), \( P(c_2|b_2) = \frac{1}{2} \), \( P(c_3|b_2) = \frac{1}{2} \).

• If \( P(b_3) = 1 \Rightarrow P(c_1|b_3) = 0 \), \( P(c_2|b_3) = 0 \), \( P(c_3|b_3) = 1 \).

Behaviors belief is more precisely transferred into the classes space using equation (1.17) according to the relations described above. Knowledge transfer is performed from the space \( 2^B \) into space \( 2^C \) which is illustrated by \( (2^3 = 8) \times (2^3 = 8) \) transfer matrix \( \bar{M} \) given by:

\[
\bar{M} = 
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}.
\]

According to the following knowledge conditioning:

• \( m_k(\{c_1, c_2, c_3\}|b_1) = 1 \) (cf Relation 1)

• \( m_k(\{c_2, c_3\}|b_2) = 1 \) (cf Relation 2)

• \( m_k(\{c_1, c_2, c_3\}|\{b_1, b_2\}) = 1 \) (cf Relations 1 and 2)

• \( m_k(c_3|b_3) = 1 \) (cf Relation 3)

• \( m_k(\{c_1, c_2, c_3\}|\{b_1, b_3\}) = 1 \) (cf Relations 1 and 3)
Multi-sensor Data Fusion: From Algorithm and Architecture Design to Applications

- \( m_k(\{c_2, c_3\}|\{b_2, b_3\}) = 1 \) (cf Relations 2 and 3)
- \( m_k(\{c_1, c_2, c_3\}|\{b_1, b_2, b_3\}) = 1 \) (cf Relations 1, 2 and 3)

This matrix enables to compute the mass functions on the classes space \( C \), in order to take a decision on the targets' classifications.

1.6.2 Simulation and results

Some simulation results are given in this section, they are performed using Matlab\textsuperscript{TM}. The first one concerns the track-to-track assignment step where the parameterless proposed solution is compared with the widely used Global Nearest Neighbor GNN algorithm [3] which is a parameter-dependent solution. The second result reminds about a local classifications comparison performed by a reliable sensor. The result highlights the outperforming of the credal classifier over the Bayesian one. Finally, the last results consider sensors with high uncertainties such that local classifications are deteriorated, and results of their fusion are evaluated. The global classification is shown to be the best when using the disjunctive credal rule of combination.

1.6.3 Assignment simulation

A scenario of two nearby targets is considered in this section. It represents a conflicting situation illustrated by figure 1.2(a). The simulation aims to calculate the rate of false decisions made by the parameterless assignment solution compared to the GNN algorithm. Notice that GNN is a parameter-dependent algorithm, it depends on parameters \( \lambda \) which can be seen as a detection distance. False decisions rates in figure 1.2(b) are given for different values of \( \lambda \).

![Figure 1.2](image)

(a) Conflicting scenario.  
(b) False assignment rates.

**FIGURE 1.2**  
Track-to-track assignment test.

The simulation presented in this section shows that the value chosen for the parameter \( \lambda \) influence the performance of the corresponding algorithm. In fact, the parameter need to be trained in order to ensure an optimal performance.
A similar comparison is made in [10] including another parameter-dependent solution [8]. The comparison highlights the advantage of using the proposed parameterless solution.

### 1.6.4 Local classification results

In this section, the scenario given in figure 1.3 is considered. First, local Bayesian and credal classifications of a reliable sensor are given. Classification results concern only target 2 in figure 1.3. Knowing that target 2 evolves the most of the time with a constant velocity, it performs two maneuvers: a first medium maneuver at time interval [62, 66] and a sharp maneuver at time interval [80, 86]. Normally, target 2 is expected to be in doubt between the three classes during the first constant velocity evolution step, to be in doubt between the second and the third classes after its medium maneuver, because targets of class 1 can not perform any maneuver, and finally, to be classified in class 3 after its sharp maneuver.

![Figure 1.3](image-url)  
**FIGURE 1.3**  
Multi-target scenario.

The Bayesian and the credal classification results are, respectively, depicted in figures 1.4(a) and 1.4(b). It can be seen that the credal classifier provides almost the expected classification, namely, perfect doubt between all classes in the first step of movement, perfect doubt between the second and third class in the second step of movement and the final classification in the third class after the sharp maneuver. At the final step, the Bayesian classifier succeed to classify the target, but it fails to manage the imprecision of the first and second steps of the movement. More details and explanations about this result can be found in [11].
The local classification results for unreliable sensors can be given by figure 1.5(a) and 1.5(b), respectively, for the Bayesian and the credal classifiers.

The objective of the proposed multi-sensor approach is to enhance the local classifications by fusing them using the described Bayesian and credal rules of combination.

1.6.5 Global classification results

In this section, local classifications fusion issue is investigated. Two unreliable sensors are designed to observe the scenario depicted in figure 1.3. The Bayesian and credal local classification Mean Square Errors (MSE), for different measurements noises are given in figure 1.6.
Belief function based multi-sensor multi-target classification solution

FIGURE 1.6
Bayesian and credal local classifications with different levels of uncertainty.

The MSE represents the difference between the expected classes pignistic probabilities and the calculated ones, it is calculated by: \( \text{MSE} = (\hat{\text{Bet}}P - \text{Bet}P)'(\hat{\text{Bet}}P - \text{Bet}P) \), where \( \text{Bet}P \) is the theoretical expected pignistic probabilities for the credal case. For the Bayesian case, the MSE represents the difference between classes calculated probability distribution \( P \) and the expected probability density distribution noted \( \hat{P} \): \( \text{MSE} = (\hat{P} - P)'(\hat{P} - P) \).

Notice that all the results of this section are averaged over 30 Monte Carlo simulations.

Results presented in figure 1.6 confirm the result depicted in figure 1.4 about the outperforming of the credal classifier over the Bayesian classifier. Figure 1.6 shows also the increasing aspect of the classification errors when sensors uncertainty increases. The multi-sensor approach aims to enhance the classification results presented in figure 1.6. Bayesian and credal global classification results are given in figures 1.7(a) and 1.7(b) respectively.

FIGURE 1.7
Global classification results using different rules of combination.
Local and global Bayesian classification results are given in figure 1.7(a). As it can be seen, the best classification result is given by max operator. Concerning the credal classification results in figure 1.7(a), the best classification is given by the disjunctive rule of combination, and worst is given by the conjunctive combination rule. This is due to the nature of the conjunctive combination which favors singletons or specific subsets over the doubt. In fact, for example, if sensor 1’s belief is given by: $m_1(\{c_2, c_3\}) = 0.4$ and $m_1(\{c_1, c_2, c_3\}) = 0.6$, and sensor 2’s belief is given by: $m_2(\{c_2, c_3\}) = 0.5$ and $m_2(\{c_1, c_2, c_3\}) = 0.5$, the conjunctive combination gives: $m_1 \& m_2(\{c_2, c_3\}) = 0.7$ and $m_1 \& m_2(\{c_1, c_2, c_3\}) = 0.3$. This explains the accentuation of the the classification divergence caused by sensors noises during the first and second steps of movement where doubt is preferred. On the other hand, the disjunctive combination of the same mass functions gives: $m_1 \lor m_2(\{c_2, c_3\}) = 0.2$ and $m_1 \lor m_2(\{c_1, c_2, c_3\}) = 0.8$. This illustrates the cautious nature of the disjunctive combination which helps to obtain the best global classification result for the considered problem.

![Graph](image)

FIGURE 1.8
Multi-target scenario.

Figure 1.6.5 compares the best Bayesian classification result with the best credal classification result. It can be seen the the credal global classification is more robust that the Bayesian one.

1.7 Conclusion

This chapter recalls some previous results of kinematic data based classification [20, 11]. In a case of nested classes, the credal classifier was shown to outperform the classical Bayesian classifier. It is shown in this chapter that classification results can be deteriorated when it comes to manage high sensor
uncertainties. Therefore, a centralized multi-sensor approach was proposed, it aims to enhance the classification results by fusing sensors local classifications.

Multiple Bayesian and credal fusion rules are tested. It was shown that the best classification result is given by the credal disjunctive rule of combination. It is a cautious fusion rule which seems to be adapted to highly imprecise classification scenarios such is the case here for a nested classes situation.

Notice that before fusing the local classifications, a consensus on the tracked targets needs to be reached. This is done here using a new parameterless track-to-track matching method. Contrarily to the existing methods, like GNN algorithm for example, the proposed method does not need any parameter training which is appreciable for random tracking environments applications.

Through the results of this chapter, it can be seen that the application of the credal formalism in multiple target tracking and classification context is a challenging and a promising task. It is here applied to the association and the classification steps. In the future, filtering methods with belief function will be considered.


22Multi-sensor Data Fusion: From Algorithm and Architecture Design to Applications


