Sensing what matters

Wilbert van Norden

2010

Sensing what matters

Proefschrift

ter verkrijging van de graad van doctor

aan de Technische Universiteit Delft,

op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,

voorzitter van het College voor Promoties,

in het openbaar te verdedigen op dinsdag 16 februari 2010 om 10:00 uur

door Willem Leendert VAN NORDEN
master in Media and Knowledge Engineering

geboren te Katwijk

Dit proefschrift is goedgekeurd door de promotor:

prof.dr. C.M. Jonker

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. C.M. Jonker	Technische Universiteit Delft, promotor
Prof. dr. ir. L.C. van der Gaag	Universiteit Utrecht
Prof. dr. F.M.T. Brazier	Technische Universiteit Delft
Prof. dr.h.c. ir. P. van Genderen	Technische Universiteit Delft
Prof. drs. dr. L.J.M. Rothkrantz	Technische Universiteit Delft
Dr. J. Dezert	ONERA
Dr. ir. F. Bolderheij	CAMS – Force Vision

Dr. ir. Fok Bolderheij en prof. drs. dr. Léon Rothkrantz hebben als begeleider in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.

ISBN : 978-90-9024894-3

Force Vision
centre for automation of mission-critical systems

TU Delft

Talga Vassternich

(Deserve Victory)

Wizard's Eighth Rule – Naked Empire

This thesis was prepared using LaTeX typesetting. The research described in this thesis was executed primarily at CAMS Force Vision. The object of this research was to complete the objectives of the student. The conclusions and recommendations do not necessarily represent the opinion of CAMS Force Vision

Illustrations : W.L van Norden
Cover Design : W.L. van Norden & H. Groof

ISBN : 978-90-9024894-3

Preface

THIS research began directly after my Masters of Science thesis. After a short period aboard one of the Dutch frigates, I came in contact with my thesis advisor from the Royal Netherlands Naval College, commander dr.ir. Fok Bolderheij, who asked if I was interested in working at CAMS – Force Vision in the new Planning and Decision Support (PADS) department of which he was the head. One of the main aspects of my work there would be to start with the implementation of his PhD work and the schedulers I designed during my MSc research. Furthermore, a project would start together with Thales Netherlands and TNO on Management and Fusion of Sensors (MAFUSE) and I would be required to play some part in that research project.

Together with prof.drs.dr. Léon Rothkrantz, my MSc-thesis advisor from Delft, we began to outline a PhD project along the lines of MAFUSE focussing on sensor management. During that period, Léon and Fok both provided me with much support and help, for which I am very grateful. At Force Vision I was given the opportunity to work on this research for which I would like to thank the company for allowing me to use some time for research instead of production work. It is my nature to just want to solve a problem and I thank Léon for always scientifically questioning my proposed methods and pointing out other relevant work in various related fields.

After some time, the focus of the research shifted from sensor management towards reasoning and determining information requirements. It was in this stage that I came in contact with my supervisor prof.dr. Catholijn Jonker. Under her supervision my methodologies found their current form. While working on the concepts and running tests, Catholijn helped me to focus my reasoning. She also helped me with writing a good elucidation that emphasises all aspects of this research. Furthermore, I thank Catholijn for her trust in me and in my work and for the always positive and cheerful atmosphere.

Without the help of Catholijn, I would not have been so productive with publications. Despite the disappointment of some rejected papers, I do thank the various anonymous reviewers who, besides rejecting my papers, pointed out relevant literature that I had missed.

During my research I worked at CAMS – Force Vision with LTZE2OC Krispijn Scholte and LTZE2OC ir. Tanja van Haarst. Thank you both for the relaxed working atmosphere, the laughs and for your support. Furthermore, thank you Krispijn for your work on the reduction of time complexity in the algorithms and for your general computer help (e.g., with Matlab and LaTeX). Tanja, thank you for our many conversations and for the laughs that resulted from us annoying our colleagues.

During my research Fok was recommissioned at the Netherlands Defence Academy and was replaced by drs. Vincent van Leijen. Thank you Vincent for putting up with me, for maintaining the atmosphere of the PADS group, for your help with mathematics and for our discussions on various subjects.

During the final stages of writing this thesis, the PhD examination committee members have provided me with valuable comments. This has helped me tremendously in finalising and improving my work. I thank you all for your agreement to be part of my PhD examination committee and for the time you took to help me improve the thesis.

Last but not least, I would like to thank my family and friends and especially Miranda for putting up with my reaction to research related stress. It is underestimated how much work can be done if one rests enough and is given room to do nothing.

Wilbert van Norden

Den Helder, December 2009

Abstract

Sensing what matters

DECISION support functionalities are needed to support the human operators on board Royal Netherlands Navy (RNLN) ships since the missions are increasingly complex and they take place in increasingly complex environments. Furthermore, growing complexity in sensor systems requires more knowledge to utilise these sensor systems to their fullest potential. The available human knowledge on board RNLN ships however is decreasing due to a strive to reduce ship's complements and to reduce their training and education time. Where previously each individual sensor was assigned to a specialised operator, now one generic sensor operator is expected to control all sensors together. Automation is therefore needed to support that operator with that task.

It is critical to mission success to identify threats as soon as possible and sensors are needed to provide the necessary information. Based on timely observed threats, the appropriate actions can be chosen to ensure mission success and safety of the ship. Since not all objects in the environment can be observed at the same time, the deployment of the sensing capabilities should be optimised as best as possible. For mission success it is therefore essential to optimally deploy sensors to obtain the relevant information about objects to be able to identify threats.

In order to determine which objects pose the most risk for mission success, classification is essential. Based on the classification, objects can be ruled out as a threat or identified as one. The focus of this research therefore, is to automate the classification process. When different classification outcomes conflict with each other, sensor tasks should be requested to resolve those conflicts. Sensor management should try to reduce the uncertainty on the most dangerous objects as best as possible.

Reducing uncertainty to improve the classification process requires knowing what information to obtain and how this information may be obtained. To gain the best possible uncertainty reduction, sensor settings need to be adapted to the situation. Knowledge on how the sensors operate in different environments is required to optimise performance to fulfil the information requirements.

Due to the number of objects in today's complex missions, information requirements are substantial. A prioritisation mechanism for sensor tasks is therefore needed. Determining which information is considered to be more important than other information, is directly related to how much threat the object under consideration poses to the mission. For prioritisation purposes, the worst possible case is assumed which gives the upper boundary of the uncertainty in the risk estimation. This maximum possible risk an object poses is used as the priority for the information need that is determined for that object.

Sensor measurements always have a certain degree of uncertainty. Traditional classifiers are not designed to cope with this uncertain input. Furthermore, traditional classifiers require a training set containing different examples of all classes and such a dataset is not available. A model-based approach is therefore introduced in this thesis that uses models of possible objects. These model-based classifiers are also designed to handle uncertainties in the input. Due to this approach, the resulting classifiers can also determine which input uncertainty needs to be reduced in order to improve the classification solution.

The model-based classification approach introduced in this thesis may be used to construct a number of classifiers. The results of the individual classifiers need to be combined to a single classification result. A suitable combination rule for this is Dezert-Smarandache theory (DSmT). This combination rule is chosen since it can handle highly conflicting sources that express belief on solution sets with overlapping labels. Using DSmT to combine the different classifiers leads to a single classification solution.

Human operators are traditionally used for classification tasks in complex problem domains like the military domain. The classification system introduced in this thesis utilises the expertise of the operator. In the system the operator provides his knowledge of a specific situation by changing the labels used for classification or by providing a classification result. The original DSmT focusses on automating fusion problems without user interaction. Rules to enable the required user interaction are therefore developed in this thesis as well.

For any classification system, it is important to verify performance based on suitable evaluation criteria. The suitability of the criteria depends on the application domain in which the classification system needs to operate. In the case of naval missions, the overall system needs to deal with a large, hierarchical and changing label-set, with uncertain input, and it needs to indicate if there is confusion between labels. The latter characteristic is referred to as soft classification for which evaluation criteria are available in literature. These however, are not suitable for hierarchical label-sets. On the other hand, existing criteria that are suitable for hierarchical label sets cannot handle soft classifier output. This thesis therefore, introduces new criteria to appropriately evaluate the performance of the classification system based on the characteristics of the problem domain.

Tests with the new classification system show significant improvement of classification results over traditional systems. Furthermore, this thesis shows that the new system is capable of describing the information requirements that may be used as an input for sensor management. In a simulated environment, several classification systems have been tested and compared. This comparison is done with respect to existing criteria as well as the newly developed ones. We can conclude that the new classification system outperforms existing methodologies.

In short, the contribution of this thesis is threefold. Firstly, combining different model-based classifiers with DSmT leads to improved classification of objects as well as an improvement in sensor deployment and this system is therefore essential to successfully execute the missions the RNLN is faced with today and in the future. Secondly, the new model-based classification approach may be applied for any classification task that deals with uncertainty, multiple non-exclusive labels, and for which knowledge about the possible labels is available. Lastly, new criteria have been developed for the evaluation of soft classifiers operating on hierarchical solution spaces.

Wilbert van Norden

Propositions

Propositions belonging to the thesis "Sensing what matters" by Wilbert van Norden.

1. In sensor management it is important to determine which information one does not need.
2. Existing sensor management approaches are not implemented in current systems because they fail to address two key factors: 1. what should the sensors be doing and 2. which sensor is appropriate for which task.
3. Optimal sensor management does not garantee mission success.
4. Information-wise, disagreement is more valuable than agreement.
5. Certainty and specificity are mutually exclusive.
6. Sytem accuracy does not correspond with a micromort, so the value of life must be incoorperated in the system in another way.
7. Sensors will not perform according to their specifications in unforeseen circumstances. Unforeseen circumstances are normal.
8. Not being totally wrong for the right reasons is better than being exactly right for the wrong reasons.
9. Executing modern maritime military operations, such as peacekeeping and counter drugs, is an uncertain business.
10. When writing a PhD thesis you can easily run out of symbols.

These propositions are considered to be defensible and are approved as such by the supervisor prof. dr. C.M. Jonker.

Samenvatting

Waardevol waarnemen

Technologiën om de gebruiker aan boord van schepen van de Koninklijke Marine (km) te helpen met de besluitvorming zijn benodigd door de steeds complexere missies die plaats vinden in complexere omgevingen. Voorts is een hoger kennisniveau bij de gebruiker vereist vanwege de groeiende complexiteit van sensoren om deze op de juiste wijze te kunnen inzetten. De direct beschikbare menselijke kennis aan boord van de schepen van de km daarentegen is dalende vanwege het streven om bemanningen van schepen en de voor hen beschikbare trainings- en opleidingscapaciteit te verkleinen. Waar er vroeger voor elke afzonderlijke sensor een specifieke operator was moeten nu alle sensoren door één gebruiker bediend worden. Er is dus automatisering nodig om de gebruiker te ondersteunen bij het uitvoeren van die taak.

Het zo snel mogelijk identificeren van dreigingen is essentieel voor missiesucces en sensoren zijn benodigd om de hiervoor benodigde informatie te vergaren. Door dreigingen tijdig waar te nemen, kunnen tegenacties uitgevoerd worden om missiesucces en de veiligheid van het schip te waarborgen. Aangezien het ondoenlijk is alle objecten te allen tijde te bemeten, moet de sensorinzet geoptimaliseerd worden. Om missies veilig tot een goed einde te brengen is het essentieel dat de sensoren optimaal benut worden om de relevante informatie over objecten te vergaren om zo in staat te zijn dreigingen correct en tijdig te identificeren.

Classificatie van objecten is essentieel indien er bepaald moet worden welk van de objecten het meest dreigend is ten opzichte van de missiedoelstellingen. Gebaseerd op de classificatie kunnen doelen direct worden uitgesloten danwel aangemerkt worden als bedreiging. Het hoofddoel van dit onderzoek is daarom het automatiseren van het classificatieproces. Wanneer verschillende classificatie-uitkomsten met elkaar in conflict zijn, moeten sensortaken aangevraag worden om dit conflict op te lossen. De sensoraansturingsmethodiek moet streven naar een zo goed mogelijke onzekerheidsreductie van de meest gevaarlijke doelen.

Alleen weten welke informatie benodigd is om onzekerheid in classificatie te reduceren is niet afdoende, het is ook noodzakelijk te weten hoe deze informatie verkregen kan worden. Teneinde zoveel mogelijk onzekerheid te reduceren moeten de sensorinstellingen aangepast worden aan de situatie. Het bepalen van de instellingen om de benodigde informatie te verkrijgen vereist kennis over de effecten van omgevingsfactoren op de prestaties van de diverse sensoren.

Gezien het aantal objecten binnen de huidige missies is de informatiebehoefte dermate groot dat de diverse behoeftes geprioriseerd moeten worden. Bepalen welke informatiebehoefte belangrijker is dan een andere is gerelateerd aan de dreiging dat het object waarvoor de informatiebehoefte gesteld is met zich meebrengt. Voor het prioritiseren wordt de meest dreigende situatie aangenomen welke (nog) niet uitgesloten kan worden op grond van de beschikbare informatie. De prioriteit is dus de bovengrens van de onzekerheid bij de risico-inschatting. Dit maximale risico van een object geeft de prioriteit aan van de informatiebehoefte die is vastgesteld voor dat object.

Sensormetingen hebben altijd een bepaalde mate van onzekerheid. Traditionele classifiers zijn niet ontworpen om onzekere invoerwaardes te verwerken. Voorts vereisen deze traditionele classifiers een set met voorbeelden van alle mogelijke klasse voor training en een dergelijke set is niet beschikbaar. Dit proefschrift introduceert daarom een model-gebaseerde classificatie aanpak die gebruik maakt van modellen van mogelijke klassen. Deze model-gebaseerde classifiers zijn ontworpen om onzekere invoerwaardes te verwerken. Door deze aanpak kunnen de classifiers ook bepalen welke onzekerheid over een van de invoerwaardes verkleind moet worden teneinde de classificatie oplossing te verbeteren.

De in dit proefschrift geïntroduceerde model-gebaseerde classificatiemethodiek kan gebruikt worden om verschillende classifiers te maken. De uitkomsten van deze classifiers moeten gecombineerd worden. Een geschikte regel voor deze combinatie is Dezert-Smarandache Theorie (DSmT). Er is voor deze combinatieregel gekozen omdat deze om kan gaan met conflicterende informatie van bronnen die een mate van geloof uitdrukken op overlappende labels. Door gebruik te maken van DSmT kan er een enkele classificatie-oplossing bepaald worden.

De menselijke gebruiker voert traditioneel de classificatietaken uit in complexe probleemgebieden zoals het militaire domein. Het classificatiesysteem dat in dit proefschrift is ontwikkeld gebruikt de expertise van de gebruiker. In het system kan de gebruiker zijn kennis over de situatie aangeven door, óf de mogelijke labels aan te passen óf door zelf een classificatie aan te dragen. De oorsponkelijke DSmT richt zich voornamelijk op het automatiseren van combinatieproblemen zonder interactie met de gebruiker. Regels om deze interactie wel mogelijk te maken worden daarom ook ontwikkeld in dit proefschrift.

Voor elk willekeurig classificatiesysteem is het belangrijk om de werking te verifiëren met behulp van geschikte evaluatiekriteria. De geschiktheid van de kriteria wordt mede bepaald door het applicatiedomein van het classificatiesysteem. In het geval van militair maritieme missies, moet het systeem kunnen omgaan met een grote, hierarchische en veranderende set van labels, onzekere informatie, en het moet de eigen onzekerheid en/of verwarring tussen labels aangeven. Laatstgenoemde karakteristiek wordt ook zachte classificatie genoemd waarvoor kriteria beschikbaar zijn in de literatuur. Deze zijn echter niet in staat om te gaan met hierarchische labels. Aan de andere kant, bestaande kriteria die daar wel mee om kunnen gaan, kunnen niet omgaan met zachte classificatie oplossingen. Daarom introduceert dit proefschrift nieuwe kriteria om de classificatie resultaten te evalueren gebaseerd op de karakteristieken van het applicatiedomein.

Testen met het nieuwe classificatiesysteem tonen een significante prestatieverbetering aan ten opzichte van traditionele classifiers. Tevens toont dit proefschrift aan dat het nieuwe systeem in staat is de informatiebehoefte te omschrijven die gebruikt kan worden als invoer voor de sensoraansturingsautomatisering. In een gesimuleerde omgeving zijn diverse classificatiesystemen getest en vergeleken. De vergelijking is uitgevoerd met de bestaande en de nieuw ontwikkelde evaluatiekriteria. We kunnen concluderen dat het nieuwe classificatiesysteem beter presteert dan bestaande technieken.

Resumerend, de bijdrage van dit proefschrift is drieledig. Ten eerste, het combineren van verschillende model-gebaseerde classifiers met DSmT leidt tot een verbeterde classificatie van objecten alsmede tot een verbetering van de inzet van sensoren waardoor dit systeem essentieel is voor het succesvol uitvoeren van de missies die KM vandaag en in de toekomst opgedragen krijgt. Ten tweede, een nieuw model-gebaseerd classificatiesysteem is ontwikkeld dat gebruikt kan worden voor elke classificatietaak waarbij omgegaan moet worden met onzekerheid, overlappende labels en waarbij kennis beschikbaar is om de mogelijke klassen te modelleren. Ten slotte, nieuwe evaluatiekriteria zijn ontwikkeld om zachte classifiers te evalueren die gebruikt worden om oplossingen te vinden in een hierarchische oplossingsruimte.

Wilbert van Norden

Stellingen

Stellingen behorende bij het proefschrift "Sensing what matters" van Wilbert van Norden.

1. Binnen sensor management is het belangrijk te bepalen welke informatie niet benodigd is.
2. Bestaande sensoraansturingsmethodieken zijn niet geïmplementeerd in bestaande systemen omdat ze doorgaans twee belangrijke aspecten niet in ogenschouw nemen: 1. wat moeten de sensoren doen en 2. welke sensor is goed in welke taak.
3. Optimale sensoraansturing garandeert geen missiesucces.
4. Qua informatieve waarde is onenigheid meer waard dan consensus.
5. Zekerheid en specificiteit sluiten elkaar uit.
6. Systeemnauwkeurigheid komt niet overeen met een micromort, dus moet de waarde van een leven op een andere wijze ingebracht worden in het systeem.
7. Sensoren opereren niet volgens de specificatie in onvoorziene omstandigheden. Onvoorziene omstandigheden zijn normaal.
8. Het niet helemaal ongelijk hebben om de juiste redenen, is beter dan helemaal gelijk hebben om de verkeerde redenen.
9. Het uitvoeren van moderne maritiem militaire operaties, zoals *peacekeeping* en *counter drugs*, is een onzekere bezigheid.
10. Tijdens het schrijven van een proefschrift kunnen de symbolen snel op raken.

Deze stellingen worden verdedigbaar geacht en zijn als zodanig goedgekeurd door de promotor prof. dr. C.M. Jonker.

Contents

1 Introduction

The only sovereign I can allow to rule me is reason. The first law of reason is this: what exists, exists; what is, is. From this irreducible bedrock principle, all knowledge is built. This is the foundation from which life is embraced. Reason is a choice. Wishes and whims are not facts, nor are they a means to discovering them. Reason is our only way of grasping reality - its our basic tool of survival. We are free to evade the effort of thinking, to reject reason, but we are not free to avoid the penalty of the abyss we refuse to see.

Wizard's sixth rule – Faith of the Fallen (Ch.2)

When the first Radio Detection And Ranging (RADAR) systems became available aboard naval vessels, the human operators were responsible for all settings and management tasks for these RADARs. Today, technological developments and political choices have led to the necessity of making sensor systems more intelligent for modern military missions. This, in short, is the reason for the research that is done in the field of Command and Control (C2) and sensor management in particular.

This chapter will elaborate on this reason in the first section. In the two following sections previous work is discussed that has been the starting point for this research. The last two sections of this chapter state the problem definition and give an outline of this thesis.

1.1 Background

In the last few decades, three factors have caused the Royal Netherlands Navy (RNLN) to do research on automation within the Combat Management System (CMS) aboard her frigates. The first of these reasons can be found in the changing nature of the missions of the RNLN. During the Cold War, typical missions protected sea lines of communication against e.g., submarine threats on the Atlantic. Now, missions are more diverse in nature and location e.g., counter drug operations in the Caribbean, embargo enforcement in the Middle East and providing humanitarian help wherever needed. These missions are mostly executed in littoral waters, which are characterized by more rapidly changing meteorological conditions and more civil traffic in the vicinity of the ship compared to missions executed at open sea. Rapidly changing weather conditions make sensor performance hard to predict. As a result it is unknown to the operator where objects can or cannot be detected. The presence of dense civil traffic makes obtaining situation awareness more difficult since each object needs to be classified. Furthermore, being close to land means that enemy forces can stay undetected longer due to land clutter in the sensor systems[1] and that they can use landmasses to stay hidden from our sensor systems. This, combined with the less predictable sensor performance, makes that objects are detected much later than in open sea conditions. As a result the available time to reason on classification and intentions is shortened. This leaves dangerously little reaction time.

The second factor is financial in nature. Due to budget cuts the RNLN is striving to reduce ship complements and their available training and education time is reduced. Furthermore, where previously all specific sensors were operated by specialised operators, today one generic sensor operator is responsible for the entire sensor suite. Less people aboard who receive less training and specialised education means that the readily available human knowledge aboard ships is decreasing.

The third factor is technical. Developments in RADAR and Electro-Optical (EO) systems have given rise to complex sensor systems for which usually more than ten (technical) parameters need to be set and/or adjusted by the operator to optimise performance. This means that there are at least that many opportunities to improve the performance of these systems. However, the operator needs to have extensive knowledge of the sensor system to determine which settings are required to compensate for the consequence of environmental conditions on performance. Without adequate support for the operator, sensor performance is degraded to such an extent that threats are no longer being detected in time. Furthermore, the number of sensors and parameters makes it almost impossible to tune all sensors in time leading to degraded sensor performance. Not de-

[1]Land clutter in sensor systems is the term used for the returned energy of land masses. Since the detection threshold of sensors is typically set for the returned energy of the sea, this land clutter causes many false detections. Since the presence of land is known to the system and the operator a detection close to land may be mistaken for land clutter

tecting threats in time shortens reaction time which in turn threatens mission success.

In short, the RNLN is executing increasingly complex missions in increasingly complex environments with advanced sensor systems while less required human knowledge is available on board. This growing discrepancy gives rise to the need for automation to support the human operators. In order to make such automated systems, a good understanding is needed of the different C2 processes involved in military operations.

1.2 Command and Control

The need for automation is driven by the growing discrepancy between required and available human knowledge. This is partly due to reduced manning as well as reduced training and education time for personnel. Therefore, we decided to look at C2 from the operator's perspective. The starting point for this approach is the cognitive model of how human operators cope with C2 tasks, as proposed by Van Delft in [19]. That cognitive model divides C2 into four processes:

- Situation Assessment (SA);
- Threat Assessment (TA);
- Decision Making (DM);
- Direction and Control (DC).

These four different processes are similar to the well-known Observe-Orient-Decide-Act (OODA) loop from Boyd, [11], and can be used for automation of C2 processes. In SA all objectively measurable attributes, e.g., speed and position, of objects are determined (*observe*). Subjectively measurable attributes — such as the identity of an object: friendly or hostile, or the threat the object poses — are determined in the TA processes (*orient*). The combined SA and TA processes are called the picture compilation process. In DM decisions are made about appropriate (counter) actions which are allocated to available resources (*decide*). Resources assigned to tasks are controlled by DC processes (*act*). Combined the four, originally cognitive, C2 processes can be seen as a control cycle for arbitrary resources.

Information can follow different paths through the C2 model to come to actions. In the first path observations immediately lead to control actions (from SA directly to DC) which is called the primary level. The TA and DM processes together are called the secondary level. Both the primary and the secondary level of C2 are discussed in [19]. By training, more actions can be taken through the primary level and only the complex situations go through the secondary level. In that sense the primary level has parallels with Recognition Primed Decision-Making (RPDM), [53].

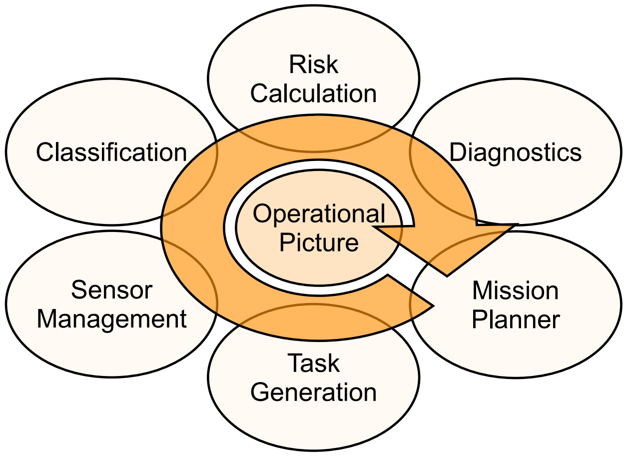


Figure 1.1: Processes involved in the Command & Control process that are vital to the picture compilation process

Bolderheij et al. [10] and De Greef et al. [33] use the cognitive C2 model as the basis for an object-oriented approach to C2 aboard RNLN frigates, more specifically they both describe object-oriented sensor management within the C2 concept. From the cognitive C2 model it already becomes apparent that the information about the world is vital for all processes within C2. In this research the representation of the world is referred to as the Operational Picture (OP) see figure 1.1. The object-oriented approach to C2 is used as the basis for this thesis.

The paradigm shift proposed by Bolderheij et al. in [10], results in a system with decoupled processes, each process using information from and/or giving information to the OP. This means that the processes can be implemented as concurrent or sequential processes or by a mix of both. This notion of concurrent processes fits well with the revision of the Joint Director of Laboratories (JDL) model for information (or data) fusion for which Llinas et al. propose a method for communication between the different fusion levels in [60].

This model with decoupled processes has similar functionalities as the Networked Adaptive Interactive Hybrid System (NAIHS) model proposed by Kester in [52]. Furthermore, in [85] Rasmussen describes how humans process information ton come to a decision. In this model, the information may be processed following a number of different paths through different processes. Bolderheij's object-oriented model provides similar processes that may run in parallel both models are thus similar. Since JDL, NAIHS, OODA, RPDM, and the cognitive model of Rasmussen can be mapped to the model of Bolderheij, which is both simple and flexible, this latter model is used throughout this work.

Each process, whether executed concurrent or sequential, depends on the information that is available about the world in which we operate. One of the main sources for this information are the sensors. In order for those sensors to deliver the right information at the right time a management methodology is required. Managing the sensor suite correctly is of vital importance since it determines the quality of information on which all command decisions are made.

1.3 Sensor Management

The purpose of having a sensor is quite straightforward: it provides information about the world in which we operate. The previous section already stated the importance of an accurate OP and since sensors are vital in obtaining and maintaining this picture, sensor management is an important process. This section focusses on two aspects of sensor management: firstly, the management considerations themselves, and secondly the prioritisation of different tasks that need to be executed.

1.3.1 Managing a sensor suite

Sensor management is vital for C2 and has therefore been the focus of studies in the past. In [97] Strömberg et al. conduct a survey of sensor management techniques that focus on the required technical settings to improve performance of individual sensors. The added value of sensors to achieve mission success has been described by McIntyre and Hintz in three papers, [64], [65], and [66], although they are not specific on how such a system should be developed. This section describes an approach to sensor management based on the picture compilation process which is vital for mission success. It provides the specifics that are required to create the sensor manager proposed by McIntyre and Hintz.

Operationally speaking, sensors should provide the best information possible on the most relevant objects given the mission objectives. In [10], Bolderheij et al. state that sensor management should support the picture compilation process — which is the combination of Situation Assessment (SA) and Threat Assessment (TA) — as good as possible, along the lines of the notions on sensor management of Hall in [35] and [36]. By representing objects in what is called the Operational Picture (OP), a data-store[2] is created in which all C2 processes may read and write information. In the model of Bolderheij et al., the OP not only consists of existing tracks (already detected objects) but it also contains the expected objects as derived from intelligence reports and mission statements.

[2]Depending on the implementation such a store could be viewed as a blackboard or in the case of an agent-system as a marketplace

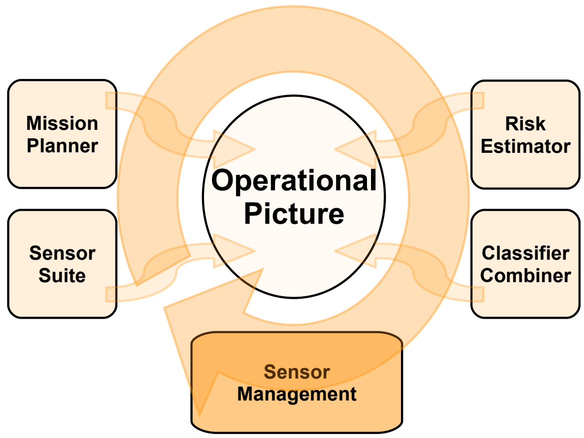


Figure 1.2: General model for sensor management

Each object in the OP comes with its available attribute information and the related uncertainty therein. The uncertainty on expected objects is used to determine the required surveillance tasks, see [104]. Through the representation of all detected and expected objects in the OP sensor management can determine which information should be obtained given the current situation and the mission statements.

Based on the OP, sensor management has to determine how to deploy the sensors to obtain the required information. The resulting sensor control cycle is depicted in figure 1.2. In order to determine which sensor to use and to calculate the required settings for that sensor, first the information need for each sensor function, similar to the approach suggested by Johansson and Suzić, [46], needs to be determined. In the naval warfare domain e.g., when searching for a sea skimming missile a minimum detection range determines the scanning frequency. Using such timing constraints for scanning areas is also discussed by Duron and Proth in [27] although Duron and Proth discretise the environment where we do not. The resulting control cycle is based on the uncertainty about detected objects in the OP that leads to information needs.

By adding expected objects and/or adjusting the information on available objects, the operator can direct the surveillance capabilities of the available sensor suite. This means that the operator no longer needs to think about which sensor to use in which mode but he can tell the system the expected threats and the system will take care of optimizing the sensor suite to search for these threats. This is in full agreement with recent developments on the JDL model like the Level *5* user refinements proposed by Blash and Plano in [6]. The operator is still able to interact with sensor systems on a more technical level if desired since those interfaces are still maintained, thus enabling the implementation of different levels of automation, [29]. While full automation might be achieved, the operator can still interact with the system at any desired level.

In other domains similar management schemes have been proposed for sensor management. In [12], Cai and Ferrari describe how a sensor's route may be determined to optimally classify a number of objects in an environment. Or in more generic terms, it is optimised for a treasure hunt problem. The principle of managing a sensor based on desired information is the same. The difference however is that they consider a mobile sensor with limited settings, whereas in this thesis the focus is on multiple sensors with a multitude of settings that are more or less stationary with respect to the environment; the stationary platform we consider cannot move through the surveillance area in a similarly short period of time as is the case in the treasure hunt problem. Krysander and Frisk [55] describe the problem of sensor placement for fault detection and isolation where similar differences exist. The principle of using expected objects for planning activities is discussed by Mohn in [70] for the Army domain. Although the required knowledge to come to task generation and task allocation is different, the principles are the same.

1.3.2 Prioritising sensor tasks

In order to schedule a sensor suite, the different sensor tasks that need to be executed are prioritised. This prioritisation is usually either left to the operator or the prioritisation is solved by assigning priorities to different types of sensor tasks like e.g., track or horizon search. An overview of sensor scheduling methodologies that use this approach can be found in e.g., [72].

In contrast to the approach of prioritisation of different sensor functions, Komorniczak et al. describe a process to prioritise detected targets in order to make a distinction between those tasks in [54]. Romberg describes the prioritisation for search areas in [86], whereas Miranda et al., [69], describe a prioritising methodology based on simulations.

Similar to these three approaches, Bolderheij proposes to use risk calculation — as described by Yellman in [113] as probability of occurrence multiplied by the cost of such occurrence — in [10], which is similar to Romberg, [86]. Other work on threat assessment can be found in e.g., [15], and [47], but they focus on a subset of air targets. The risk calculation of Bolderheij provides a more flexible model which may be used for any type of target and is therefore used in this thesis.

A similar approach for using risk as a prioritisation mechanism in sensor management can be found in [81], where it is applied in missile defence systems. In [79], Osadciw and Veeramachaneni discuss how risk calculation is used to construct a fitness function for managing a sensor network where the focus does not lie with the management problem itself, rather it tries to determine the information need for optimal picture compilation for C2.

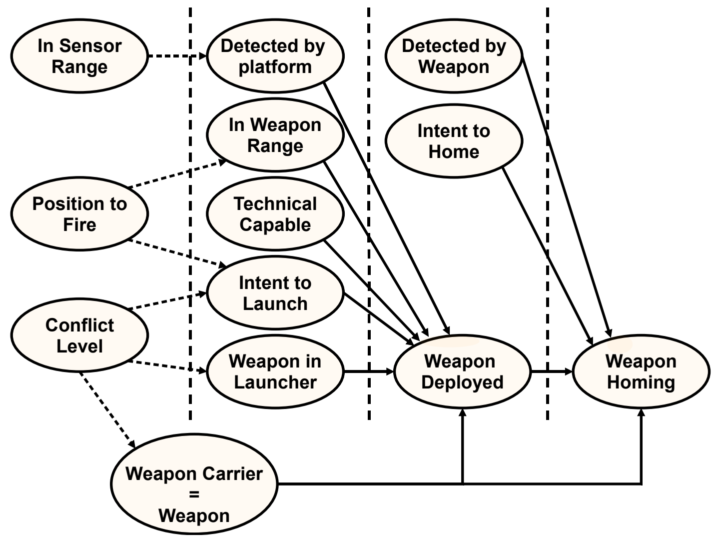


Figure 1.3: The DBN used for risk calculation, where a weapon (e.g., a missile) directed against the own platform or the unit that needs protection poses most risk

Risk calculation in management problems has also been used in the space exploration domain by Mehr and Tumer in [67]. It therefore seems reasonable to also adopt risk calculation for sensor management within the C2 framework described in this thesis for the maritime military domain.

The risk calculation approach of Bolderheij provides a uniform way to prioritise all types of sensor functions since the risk of expected objects and detected objects is determined using the same methodology. Risk calculation is done using a model like that shown in figure 1.3 which is described in [9]. In this Dynamic Bayesian Network (DBN) the relations between risk and the classification solution on an object becomes apparent. Not only does a better classification solution improve tracking performance, as shown in [2], it also enables a better risk estimation. Since many classifiers exist, each with their own pros and cons, here multiple classifiers are developed and combined.

1.4 Research questions

Sensors are used to provide a representation of the environment (also called the OP) as complete and as accurate as possible relevant to the goal of the system that is utilising the sensors. Tasks are prioritised based on the risk the object poses to the system's goal(s) as was discussed in the previous section. For maritime military missions, the risk can be calculated using the DBN proposed by Bolderheij in [9]. This method of risk calculation depends highly on the classification solution. In order to make the OP as accurate and complete as possible, classification is essential to recognise potential threats in a timely manner.

Van Haarst et al. propose to generate sensor task requests based on the uncertainty in risk, [104], where they use the methodology from Bolderheij to calculate risk. A problem occurs when trying to find a quantification of the uncertainty in the classification solution. Due to the nature of the application domain, maritime military operations, uncertainty in classification in even more important because of the potentially severe consequences.

The main goal of a sensor manager is to reduce uncertainty as best as possible. From the DBN for risk calculation it can be seen that classification is essential in assessing risk that objects pose. In order to automate sensor management, classification process needs to be addressed. This process is traditionally executed by the human operators in the military domain. Due to the shift in military missions however, it has become difficult for the human operator to execute that task parallel to the sensor management process.

The main research question this thesis addresses is:

How can operators be supported in their task of interpreting real-time data in complex environments?

The problem domain used in this thesis is the maritime military domain and the interpreting data in that domain starts with classification. Sensor are used to provide data and the sensors need to be optimally deployed in order to provide the relevant information for classification. More specific research question therefore are:

1. How should the class labels be modelled when the operators use different classification trees and require more specific or less specific answers?
2. How should classifiers cope with uncertain input from sensors and intelligence reports?
3. What conditions need to be met to combine classifiers that operate on uncertain input and that assign belief to labels on different hierarchical levels?
4. How should classifiers be evaluated taking the hierarchical levels of the class labels into account and that generic but correct answers are preferred over specific answers that may be wrong?
5. How can classification uncertainty be described and how should the classification process determine which information is needed to reduce that uncertainty?
6. How should sensor management get the required information in a complex environment?

A desirable classifiers should deal with uncertainty, multiple non-exclusive labels and incoorperate knowledge on the possible labels if this is available. The operator should be able to exert influence during the combination of the different classifiers. Furthermore, the classification system should be able to request specific information to come to a better classification. Sensor management in turn needs to deploy the sensors in such a way that it finds the requested information in a timely fashion.

Designing a system in general requires evaluating the performance of the proposed methodology. Naturally, the criteria on which performance is evaluated needs to match the application domain and it needs to be suitable for the output of the system. Besides the development of the actual system, there is a focus on evaluation criteria to estimate performance and make a comparison with other systems. These criteria have to be able to deal with non-exclusive hierarchical labels and soft classification results. They also need to provide a detailed insight in the hierarchical nature of the labels.

1.5 Thesis outline

In this thesis, the required steps to come to classification support and sensor management are discussed in the different chapters.

Chapter 2. Classification
This chapter describes the classification solution space and how such a space may be constructed for any classification problem. Based on this solution space, different classifiers are constructed that enable the system to determine which uncertainty reduction is required to improve the solution given for each individual classifier. These classifiers are based on knowledge about the possible classification solution. For any classification task that deals with uncertainty, multiple non-exclusive labels and for which knowledge on the possible labels is available the described methodology may be applied. Methods and models described in the chapter have been published at the International Conference on Information Fusion in 2008 as well as a chapter contribution, the references of those publications are:

[73] Wilbert L. van Norden, Fok Bolderheij, and Catholijn M. Jonker. Classification support using confidence intervals. In *Proceedings of the 11th International Conference on Information Fusion*, pages 295–301, Cologne, Germany, 30 June – 3 July 2008;

[74] Wilbert L. van Norden, Fok Bolderheij, and Catholijn M. Jonker. Combining system and user belief on classification using the DSmT. In *Proceedings of the 11th International Conference on Information Fusion*, pages 768–775, Cologne, Germany, 30 June – 3 July 2008;

[76] Wilbert L. van Norden and Catholijn M. Jonker. *Advances and Applications of DSmT for Information Fusion (collected works)*, volume 3, chapter 12. Utilizing classifier conflict for sensor management and user interaction, pages 371–386. American Research Press, Rehoboth (MA), May 2009.

Chapter 3. Combining classifier belief
Classifiers express a degree of belief that the object under consideration belongs to a certain labels. Since different classifiers may have conflicting results, a combination rule is used to deal with this conflict. Furthermore, the different labels need not be mutually exclusive and the combination rule needs to take that into account as well. A theory of combining information that satisfies both requirements is Dezert and Smarandache Theory (DSmT). This chapter explains how this theory is applied for combining classifiers. In Appendix A more information about DSmT is given for the interested reader.

This chapter shows how DSmT can be used to keep track of conflict between the different classifiers. By using the information from the classifier that causes most conflict, sensor function requests can be generated as an input for sensor management. In order to enable the required user interaction this chapter also introduces new rules that can be applied after the use of any combination rule.

The application of DSmT in this manner and the new rule for the addition of exerting user preferences has appeared in various publications:

[74] Wilbert L. van Norden, Fok Bolderheij, and Catholijn M. Jonker. Combining system and user belief on classification using the DSmT. In *Proceedings of the 11th International Conference on Information Fusion*, pages 768–775, Cologne, Germany, 30 June – 3 July 2008;

[76] Wilbert L. van Norden and Catholijn M. Jonker. *Advances and Applications of DSmT for Information Fusion (collected works)*, volume 3, chapter 12. Utilizing classifier conflict for sensor management and user interaction, pages 371–386. American Research Press, Rehoboth (MA), May 2009;

[78] Wilbert L. van Norden and Catholijn M. Jonker. User insisted redistribution of belief in hierarchical classification spaces. In *Proceedings of the 2009 IEEE/WIC/ACM international joint conference on Web Intelligence and Intelligent Agent Technology*, pages 115–122, Milan, Italy, 15–18 September 2009.

Chapter 4. Sensor deployment
In order to execute the required sensor functions as well as possible, different choices on sensor deployment are needed. Since this is considered a task of the human operator, there is little work done in the field of automating this. This chapter describes the required steps for an automated approach to sensor deployment. Firstly, the most suitable sensor needs to be selected for the task and secondly, that sensor needs to be scheduled and controlled during task execution.

This chapter describes how these steps may be achieved using sensor performance evaluation. Scheduling of sensors is achieved using a heuristic scheduling approach like the one used in [7]. These considerations on sensor deployment have lead to the publications below. This chapter may be seen as an extended summary of these publications:

[75] Wilbert L. van Norden, Jeroen L. de Jong, Fok Bolderheij, and Leon J.M. Rothkrantz. Intelligent task scheduling in sensor networks. In *Proceedings of the 8th International Conference on Information Fusion*, pages 1351–1358, Philadelphia (PA), USA, 25–29 July 2005;

[50] Jeroen L. de Jong and Wilbert L. van Norden. Application of metaheuristics in sensor management. In *Proceedings of the 1st international conference on Cognitive Systems with Interactive Sensors*, Paris, France, 15–17 March 2006;

[49] Jeroen L. de Jong and Wilbert L. van Norden. Application of hybrid metaheuristics in sensor management. In *Proceedings of the 18th BeNeLux Artificial Intelligence Conference*, Namur, Belgium, 5–6 October 2006. type B contribution;

[51] Jeroen L. de Jong and Wilbert L. van Norden. Application of hybrid metaheuristics in sensor management. *Aerospace Science and Technology*, 11(4):295–302, May 2007;

[104] Tanja Y.C. van Valkenburg-van Haarst, Wilbert L. van Norden, and Fok Bolderheij. Automatic sensor management: challenges and solutions. In *Proceedings of the SPIE Defense and Security Conference, Optonics and Photonics in Homeland Security (6945)*, pages 694511–1 – 694511–11, Orlando (FL), USA, 16–20 March 2008;

[58] A. Vincent van Leijen, Fok Bolderheij, and Wilbert L. van Norden. Unification of radar and sonar coverage modeling. In *Proceedings of the 12th International Conference on Information Fusion*, pages 1673–1678, Seattle (WA), USA, 6–9 July 2009.

Chapter 5. Performance issues
Evaluation of a system requires good criteria that are suitable for the problem domain. Besides verifying the functionality of a system, it should also be verified that the system set-up is feasible with respect to required computation time. This chapter focusses on both these performance issues by introducing new evaluation criteria. Firstly, by describing evaluation criteria that are suitable for non-exclusive classification tasks where belief is expressed on various class-labels. Secondly, time complexity of the combination rule is addressed and a proposition is made to reduce this complexity with minimal effect on the output. Work described in this chapter has appeared in:

[89] Krispijn A. Scholte and Wilbert L. van Norden. Applying the PCR6 rule of combination in real time classification systems. In *Proceedings of the 12th International Conference on Information Fusion*, pages 1665–1672, Seattle (WA), USA, 6–9 July 2009;

[77] Wilbert L. van Norden and Catholijn M. Jonker. Confusion and distance metrics as performance criteria for hierarchical classification spaces. In *Proceedings of the 2009 IEEE/WIC/ACM international joint conference on Web Intelligence and Intelligent Agent Technology*, pages 131–136, Milan, Italy, 15–18 September 2009.

Chapter 6. Test results
In different tests the resulting system set-up for reducing uncertainty in the OP by sensor deployment is evaluated. The classifiers are tested in a simulated environment that is representative of the maritime military domain. The performance is compared with traditional classifiers that are evaluated in the same environment. Furthermore, the performance of the new classifiers is compared to that of the traditional classifiers for a known theoretical classification task using traditional evaluation criteria. Results show that the new classifiers outperform the traditional ones. The generation of sensor functions is also shown. Finally, the reduction of time complexity is evaluated and the effects on the output are discussed and compared with simple but fast combination rules, showing that real-time application of the combination rule is possible. Results discussed in this chapter have appeared throughout the publications, [73], [74], [89], [78], and [77].

Chapter 7. Conclusions
The closing chapter of this thesis provides the conclusions of this research. These conclusions have appeared in the various publications that form the basis for Chapters *2–6*.

2 Classification

Mind what people do, not only what they say, for deeds will betray a lie.

Wizard's fifth rule – Soul of the Fire (Ch.28)

Classification is the process in which a label from a given set of labels is assigned to (a collection of) data. For the military domain it is described as the process in which one or more class-labels are assigned to a detected object. This assignment is done based on all relevant information that is available at the time. This chapter describes how the classification process can be automated and supported using Model-Based Classifiers (MBCs). Parts of this chapter appeared in conference proceedings, [73] and [74] and as a chapter contribution, [76].

A solution space containing all possible labels needs to be constructed since classifiers need to know which labels may be assigned to data. This solution space is discussed in the first section of this chapter. For each of the class-labels a membership field is defined, which is discussed in Section 2.3. Finally, Section 2.4 introduces the new MBCs. The resulting classifiers require knowledge to create the membership fields, without this the model-based approach will not work. Knowledge elicitation of the problem domain is therefore important when building MBCs.

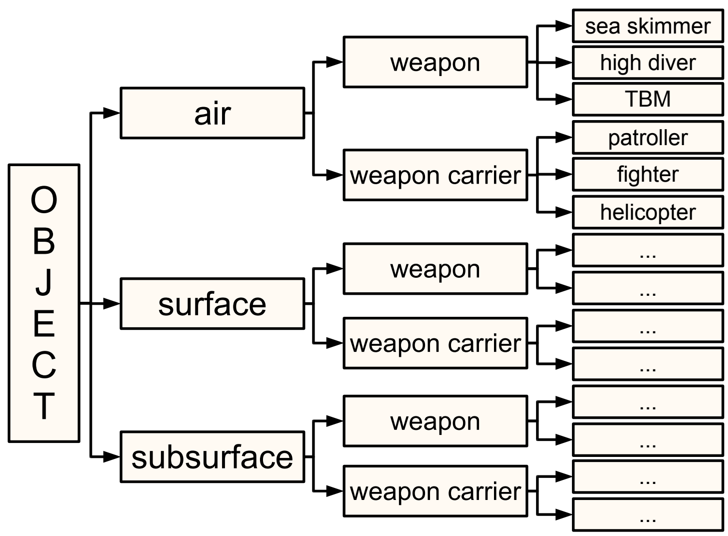


Figure 2.1: Traditional classification tree

2.1 Classification solution space

Traditional classification processes, such as described in [68] and [88], construct a classification tree as shown in e.g., figure 2.1. Constructing that classification tree is essential and requires knowledge elicitation. To avoid problems with knowledge elicitation from experts, Quinlan proposes a machine learning technique for the construction of classification trees, [84]. Other machine learning techniques for learning a classification solution space are described by Taylor et al. in [98]. The downside however is that it requires a lot of real data and processing time to learn a classification tree.

Besides the problems of constructing a classification tree there is another drawback. Each branching is done based on a characteristic of objects; the order in which this is done is rather arbitrary. As a consequence, the resulting classifier may provide only relatively high level labels when further branching is impossible given the tree structure. This can occur even though information might be available to facilitate more specific labels at a lower level in the tree. Branching on independent attributes in a pre-specified order furthermore means that certain branchings need to be done repeatedly. In figure 2.1 e.g., the first branching distinguishes between air, surface and subsurface objects. The second branching is based on whether the object is a weapon itself or if it is a weapon carrier. Both these attributes are unrelated and the same branching needs to be done in each separate branch.

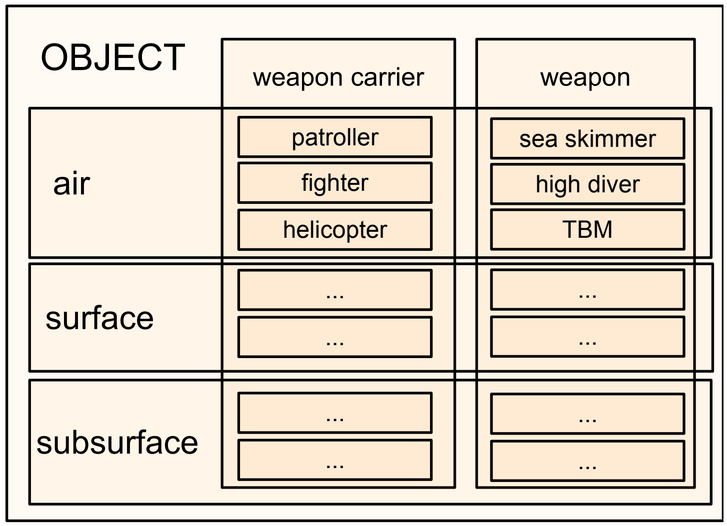


Figure 2.2: Example of a solution space shown in a Venn diagram

This section introduces a new way of modelling the classification space to solve above mentioned problems. In this model no tree is constructed. Each possible class is described based on possible attribute values. Together the attributes span a search space in which all classes — both generic classes and specific classes — are represented, see figure 2.2 where the example from figure 2.1 is shown in this new model. Classification of objects can be done by excluding parts of this search space based on available information.

In this search space several class-labels at various levels of specificity exist. In figure 2.2 the objects with highest specificity are *patroller*, TBM and such that were the final nodes in figure 2.1. At a higher specificity level the difference between *weapon* and *weapon carrier* is made. In the new model this distinction is done once whereas it had to be repeated in the tree structure approach. Using these different levels of specificity the entire classification space is modelled with this set notation in the multi-attribute space.

In figure 2.2 it can be seen that some classes overlap whereas others do not. For our model we say that all elements at the same level of specificity are mutually exclusive, meaning that although the classes may look similar with respect to attribute values, an object cannot belong to both classes. A sail boat e.g., can move at the same speed as a frigate, a detected object however cannot belong to the class *frigate* as well as the class *sail boat*.

The solution space can be represented as a collection of class labels at K different specificity levels, each containing N_k labels with $k \in \{1, 2, \ldots, K\}$,

$$\begin{Bmatrix} \{\theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,N_1}\} \\ \{\theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,N_2}\} \\ & & \vdots & \\ \{\theta_{K,1} & \theta_{K,2} & \cdots & \theta_{K,N_K}\} \end{Bmatrix}.$$

In this notation a label $\theta_{k,n}$ refers to all classes that belong to that label, for notational purposes we use only the label name in equations. The entire frame of discernment, denoted Θ, is defined as all the elements from the solution space joined:

$$\Theta = \{\theta_{1,1}, \theta_{1,2}, \cdots, \theta_{1,N_1}, \cdots, \theta_{K,1}, \theta_{K,2}, \cdots, \theta_{K,N_K}\}.$$

As mentioned earlier, all elements on the same specificity level are defined to be mutually exclusive:

$$\theta_{k,n} \cap \theta_{k,q} = \varnothing \text{ for } \begin{cases} q, n \in \{1, 2, \ldots, N_k\}; \\ n \neq q; \\ k \in \{1, 2, \ldots, K\}. \end{cases}$$

Though this distinction might seem odd, it is enforced to more accurately model the way operators view the frame of discernment in e.g., classification. On the same level of specificity an object may be classified as either a *helicopter* or an *air plane*, it cannot be both. At a different level however it may be classified as an *air* object which overlaps both of these. On yet another specificity level, the solution might be *fixed wing* or *rotary wing*. In itself these are mutually exclusive but they are used separately in operational systems. The model of the solution space is therefore chosen to enable these various overlapping class-labels simultaneously.

Classes at different levels of specificity are not mutually exclusive and can therefore overlap in the Venn diagram, as can be seen in figure 2.2. We distinguish two types of overlaps. Firstly, a class may be fully enclosed by a class at a higher specificity level. These fully overlapping elements in the branch are called child and parent classes. Since they may occur at different specificity levels, the a-th order ancestor label of label $\theta_{k,n}$ is defined as equation 2.1 and denoted $\theta_{k,n}^{\uparrow a}$.

$$\theta_{k,n}^{\uparrow a} = \bowtie \{\theta_{k-a,v} \in \Theta \mid \theta_{k,n} \cap \theta_{k-a,v} \neq \varnothing\} \tag{2.1}$$

For all $a > 0$ parent elements are found, for $a < 0$ child elements are obtained, and for $a = 0$ the element itself is found. These definitions assume that the rows in the model are ordered based on specificity.

The obtained model is a hierarchical one and each object may be assigned more than one of these labels. In literature, such types of classification tasks are

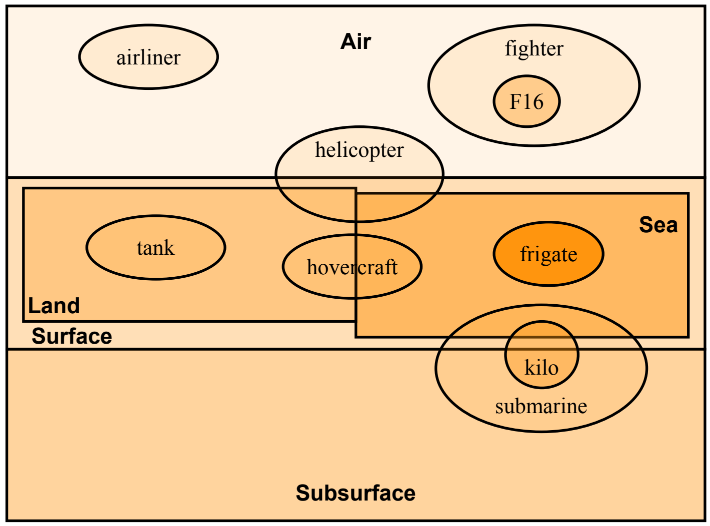


Figure 2.3: Venn Diagram of the classification solution space

referred to as multi-label learning, see e.g., [87], [18], and [14]. The modelling of the solution space is similar but differences exist. In multi-label learning each label has exactly one parent at the next higher level whereas in our approach each element may have multiple parents of the same order and furthermore, they may have no parents at all. A more detailed description of the parent-child relations follows in Section 3.3.

In figure 2.3 an example of a Venn Diagram is shown using classes at different specificity levels. At the lowest specificity level ($k = 1$), the domains are represented, *surface*, *air*, and *subsurface*. At the next level ($k = 2$) only two child elements are represented, namely the sub-domains *sea* and *land*. Two more specificity levels are represented. One representing generic objects ($k = 3$) like e.g., the *helicopter*, and one representing specific classes ($k = 4$) like e.g., the *F-16 Fighting Falcon* fighter.

Throughout this thesis, class labels are referred to as an element X_i from the frame of discernment Θ, where $i \in \{1, 2, \dots I\}$ and $I = \sum_{k=1}^{K} N_k$. A mapping $\Omega : \mathbb{N}^2 \to \mathbb{N}$ is defined by equation 2.2 to map each element $\theta_{k,n}$ on a label X_i. The reverse is obtained by the inverse mapping, $(k, n) = \Omega^{-1}(i)$.

$$i = \Omega(k, n) = \left(\sum_{u=1}^{n-1} N_u \right) + n \tag{2.2}$$

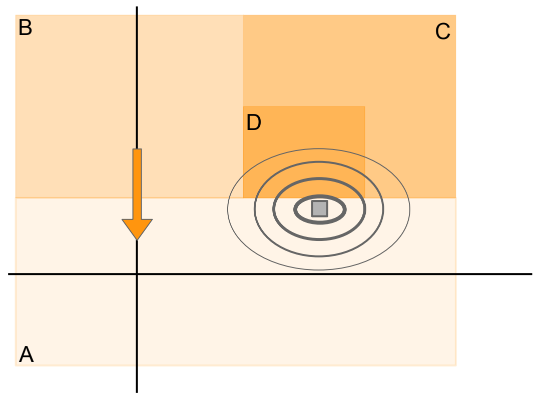


Figure 2.4: A measurement (grey square) and its uncertainty (grey lines) in any attribute space can be compared to regions in that space that are typical of a class; in this space it can also be determined which attribute needs to be measured more accurately

2.2 The MBC principle

The principle behind model-based classification builds on set notation of the classification solution space. Measurements can be plotted in the Venn diagram of the classes to determine which classes match this measurement best. The uncertainty in the measurement is taken into account as well by drawing the uncertainty lines, or confidence intervals, around the mean measurement. By comparing which parts of those lines match with which classes, a value can be assigned that quantifies how well the entire measurement fits the different classes.

Consider a four class problem with exclusive classes A, B, C and D. Figure 2.4 shows the Venn diagram of those classes and a measurement in the same attribute space. The lines around the mean of the measurement (the grey square) are the uncertainty lines. By looking at which part of which line corresponds to the different classes, a relative comparison between the fitness on the different classes can be determined. The overall solution is found by taking a weighed average of the outcomes for each line, the closest to the mean measurement the higher the weight.

In figure 2.4 the information requirements for new measurements are also visible. Most of the lines correspond to class A. The uncertainty lines however, match some of the other classes as well. Reducing uncertainty on the vertical axis could exclude those three possibilities, whereas reducing uncertainty on the horizontal axis would most likely only exclude classes B and C. From the Venn diagram can also be determined how much uncertainty would need to be reduced to exclude classes.

The principles behind model-based classification require a good model of possible attribute values of the different classes. The knowledge elicitation process is therefore important. Not only is knowledge needed about possible attribute values, also knowledge about dependencies between attribute values is needed. E.g., in figure 2.4 it can be seen that the combination of values on the horizontal axis and vertical axis is needed to distinguish between classes B, C, and D. This knowledge is expressed in membership fields which need to be constructed for all classes.

Knowledge elicitation is therefore just as important as was the case for a classification tree approach. Major advantages over the tree approach are the fact that the MBC approach cannot get stuck on a high level node, the model can be adjusted quite easily, and it provides a way to determine which attribute information improves the classification solution best.

2.3 Membership fields

The goal of classification is to assign one (or more) of the labels, X from the frame of discernment, to a detected object. To assign a label to an object, each class-label is described based on possible behaviour in some subset of observable attributes. When L different membership fields can be made for class X and each membership field is based on J_ℓ attributes with $\ell \in \{1, 2, \ldots, L\}$, the membership field is given by $\Gamma_{\ell,X}(\vec{y_1}, \ldots, \vec{y_{J_\ell}})$, where y_j with $j \in \{1, 2, \ldots, J_\ell\}$ denotes the value of attribute A_j. Figure 2.5 e.g., presents the membership for a generic object within the air domain for the subset of attributes speed and altitude.

Each object has its specific characteristics in flight. An airliner e.g. cannot fly with low airspeeds at a high altitude. Another example would be objects that can maintain altitude without airspeed. The possible combination of altitude and speed can therefore be used for classification purposes for air targets. In figure 2.5 an example for a generic air-bound class is shown where the envelope is given by the physical constraints. The area for which the membership equals *1*, gives the usual cruising speed and the cruising altitude. Where the membership equals zero the object cannot operate because of the physical constraints e.g., a flying object requires a minimum airspeed at a given altitude given its structure and weight.

Knowledge about possible objects, like presented in figure 2.5, is required for all elements X for a number of subsets of attributes. Only based on that knowledge, classifiers can assign a certain belief to each of the elements within the frame of discernment. How classifiers assign such belief is discussed in Section 2.4. The required knowledge to create these membership functions is expert user generated and formulated mostly during the planning phase of a mission.

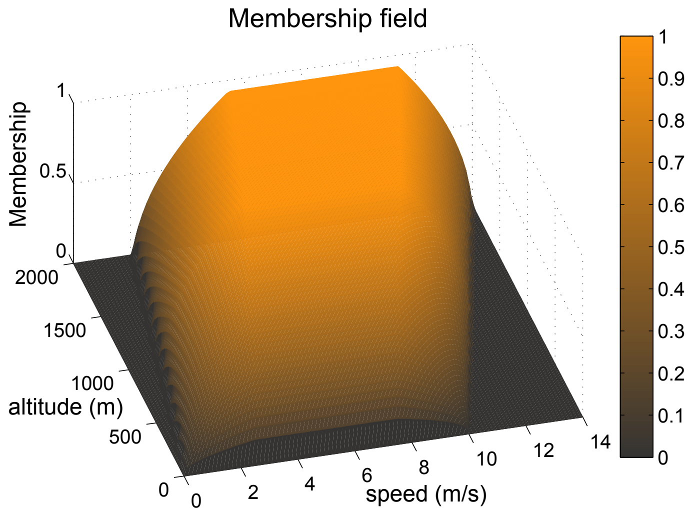


Figure 2.5: Membership of a class for attributes speed and altitude

These memberships can describe physical manoeuvring capabilities — e.g., the maximum speed of a ship given the water depth, [32]— or physical dimensions of the different classes.

Other information that might be used in the planning phase of a mission could be prior probabilities for certain classes in certain areas. In order to make this information available to the system a mission planner is required along the lines of [70]. This mission information, as well as the membership fields of classes, may be inserted, deleted, and edited during mission execution when new or additional information becomes available. The difference between this model and the model from [80] is that here multiple memberships on subsets of attributes are defined, thus obtaining multiple relations between classes that can be changed during mission execution whereas in [80] the relations are predefined and fixed. Some of this mission information is freely available like e.g., air lanes or the density of commercial shipping in [37].

2.4 Model-based classification

The previous section discussed how the classification solution space and the different classes are modelled. In order to complete the construction of a MBC, measurements are combined to match the modelling of classes and a description of the resulting classifier output is needed.

2.4.1 Combining Measurements

On each subset of attributes for which a membership field is constructed, a MBC can be run obtaining L different MBCs. Each MBC works on the same principle: determine a Confidence Interval (denoted CI) — a CI constitutes an uncertainty area as represented in figure 2.4 — based on known information and see how well this interval fits the membership function, [73]. In figure 2.4, this means seeing how well each grey uncertainty uncertainty line fits the membership fields of the various classes. The different orange areas represent the different class membership fields.

A membership field is denoted $\Gamma_{\ell,X}(y_1, y_2, \ldots, y_{J_\ell})$, and it is a function of attribute values y_j for attribute A_j with $j \in \{1, \ldots, J_\ell\}$ for class $X \in \Theta$. Measurements on attributes A_j by sensors are given by a mean value μ_{A_j} and variance ${\sigma_{A_j}}^2$. For notational ease, the subscript A_j is shortened to j. The possible values for attribute A_j may then be described by μ_j plus a fraction ξ_j of the square root of the variance σ_j, thus obtaining $y_j = \mu_j + \xi_j \sigma_j$ for all possible attribute values. Defining y_j this way is done for notational purposes later one since Probability Density Functions (PDFs) are mostly defined using means and variances. The PDF of the measurement of attribute A_j is denoted $p_j(.)$.

So far, the membership field — which is given to the system — is described in terms of attribute values and those attribute values can be written based on available information. The next step is to find the attribute values that lie within the desired CI. In figure 2.4, this means finding all values that lie within a chosen grey line which represented an uncertainty region.

To find all values with a closed region, the boundary of that closed space needs to be determined. The first in doing that is to find a single point on that boundary. Equation (2.3) describes such a boundary for CI. For each attribute values we define that point as $y_j = \mu_j + \alpha\sigma_j$, where α represents the boundary value. Due to the properties of the PDFs used in this thesis, α is the same for all attribute values.

Assuming all Gaussian distributed and independent measurements, equation (2.3) is re-written as equation (2.4) which in turn simplifies to equation (2.5) using basic mathematical operations.

$$CI = \int \cdots \int_{\mu_j - \alpha\sigma_j}^{\mu_j + \alpha\sigma_j} \left(\prod_{j=1}^{J_\ell} p_j(y_j) \right) \mathrm{d}y_1 \cdots \mathrm{d}y_{J_\ell} \tag{2.3}$$

$$CI = \prod_{j=1}^{J_\ell} \left[\int_{\mu_j - \alpha\sigma_j}^{\mu_j + \alpha\sigma_j} \frac{1}{\sigma_j \cdot \sqrt{2\pi}} \cdot \mathrm{e}^{\frac{-(y_j - \mu_j)^2}{2{\sigma_j}^2}} \, \mathrm{d}y_j \right] \tag{2.4}$$

$$CI = \prod_{j=1}^{J_\ell} \operatorname{erf}\left(\frac{\alpha}{\sqrt{2}} \right) \tag{2.5}$$

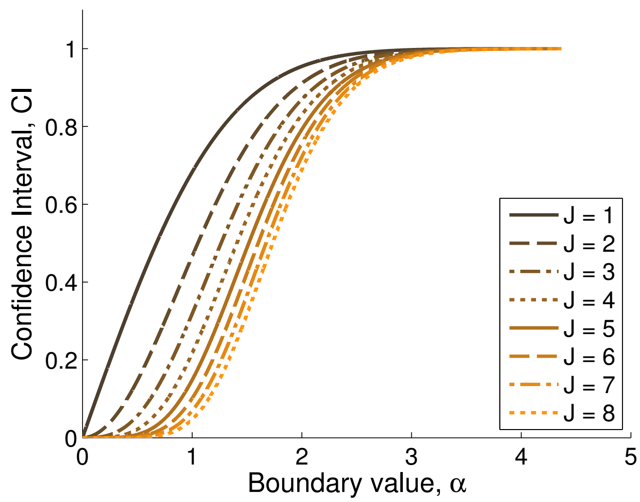


Figure 2.6: The boundary value α against the confidence interval CI for a various number of attributes, $J_\ell \in \{1, 2, \ldots, 8\}$

The error function (denoted erf) is given by equation (2.6), see e.g., [108], and is used in solving the integral of normal distributions. In order to calculate α given CI with equation (2.5) the inverse error function, denoted erf^{-1}, is needed. For the inverse error function equation 2.7 holds for $-1 \leq z \leq 1$ and equation (2.8) holds for $z \in \mathbb{R}$. The relation between the boundary value α and the confidence interval for the combination of J_ℓ attributes is shown in figure 2.6.

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_o^z e^{-t^2} dt \tag{2.6}$$

$$\mathrm{erf}\left(\mathrm{erf}^{-1}(z)\right) = z \tag{2.7}$$

$$\mathrm{erf}^{-1}\left(\mathrm{erf}(z)\right) = z \tag{2.8}$$

A single point is now found that lies on the boundary of the uncertainty area. The value of the combined PDFs should the same at each point on this boundary. All other attribute values on the boundary can be found based on α using equation (2.9). Replacing the equal sign by the $\geq$ operator finally finds all attribute values in the CI.

$$\prod_{j=1}^{J_\ell} p_j\left(\mu_j + \xi_j \sigma_j\right) = \prod_{j=1}^{J_\ell} p_j\left(\mu_j + \alpha \sigma_j\right) \tag{2.9}$$

2.4.2 Classifier solution

In order to map the entire confidence interval on membership functions of targets equation (2.9) — with the $\geq$ operator — is re-written as equation (2.10) for Gaussian distributions. With this equation all combinations of values can be found that fall within a certain CI. Calculating the combined membership for various values of CI quantifies how well a measurement fits the membership field of a certain class. Another application of calculating which attribute values constitute a contour line for a CI is reasoning on troop movements in urban environments over longer periods of time, [48].

$$\sum_{j=1}^{J_\ell} {\xi_j}^2 \leq J_\ell \cdot \alpha^2 \tag{2.10}$$

For each value of the CI the boundary value α, which is the same for all attributes as shown in [73], is determined. When all information sources provide Gaussian distributed measurements, α is given by equation (2.5). Which combinations of attribute values constitute a contour line for a given CI (or α) may then be determined with equation (2.10). Solving equation (2.10) can be done using a spherical notation due to the assumption of Gaussian distributed measurements. The contour line of the CI becomes a circle with angle $\vec{\gamma_j}$ and radius $\alpha\sqrt{J_\ell}$. By integrating over all angles the summed membership for a given CI is obtained, equation (2.11).

$$\Phi_{\ell,X}(\alpha) = \underbrace{\oint \cdots \oint}_{j\,=\,1,\ldots,J_\ell} \Gamma_{\ell,X}(\vec{\gamma_j})\,\mathrm{d}\vec{\gamma_j} \tag{2.11}$$

Function $\Phi_{\ell,X}(\alpha)$ is found, which is given in equation (2.11) and sums the membership for a given α by integrating over all possible attribute values given the different $\vec{\gamma_j}$ for label $X \in \Theta$. Again, this function is obtained for Gaussian measurements, when a different PDF for the measurements is assumed equation (2.9) does not reduce to equation (2.10) which in turn changes equation (2.11). The methodology however stays the same as is shown in Section 2.6

The total fitness of a measurement to a membership field, denoted $m_\ell(X)$, is defined as the integral over α of the summed membership values weighed by factor $W_\ell(\alpha)$. For the boundaries of this integral is known that $\alpha \in [0, \infty\rangle$ from equation (2.5) since $CI \in [0, \ldots, 1]$. This weight factor is given in equation (2.12), it reduces to equation (2.13) for Gaussian distributions and it is shown in figure 2.7.

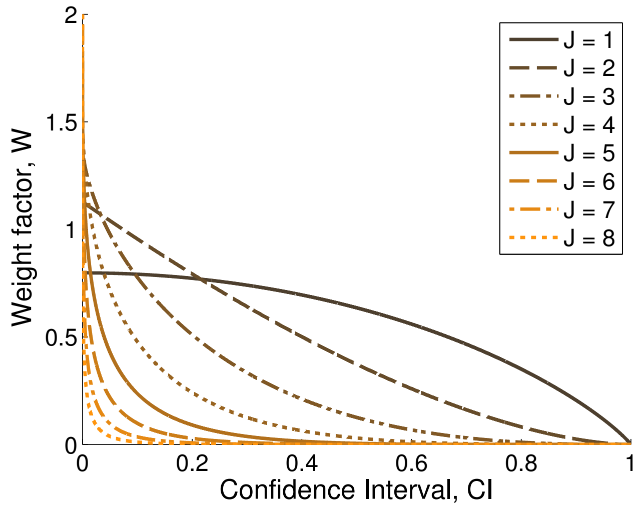


Figure 2.7: Weight factor for the various values of CI and $J_\ell \in \{1, 2, \ldots, 8\}$

$$W_\ell(\alpha) = \frac{\displaystyle\prod_{j=1}^{J_\ell} p_j\left(\mu_j + \alpha\sigma_j\right)}{\displaystyle\int_0^\infty \left(\prod_{j=1}^{J_\ell} p_j\left(\mu_j + \alpha\sigma_j\right)\right)\,\mathrm{d}\alpha} \tag{2.12}$$

$$W_\ell(\alpha) = \sqrt{\frac{2J_\ell}{\pi}} \cdot \mathrm{e}^{-\frac{1}{2}J_\ell\alpha^2} \tag{2.13}$$

The purpose of the weight factor is to be able to use it for a weighed average. It is therefore necessary to check whether

$$\int_0^\infty W_\ell(\alpha)\,\mathrm{d}\alpha \;=\; 1,$$

since that should be the case. Again, the error function — as given in equations (2.6), (2.7) and (2.8) — is needed to calculate this. In the weight factor, define t by

$$t^2 = \frac{J_\ell \cdot \alpha^2}{2} \rightarrow \mathrm{d}\alpha = \frac{2}{\sqrt{2J_\ell}}\,\mathrm{d}t$$

which leads to

$$\begin{aligned}\int_0^\infty W_\ell(\alpha)\,\mathrm{d}\alpha &= \int_0^\infty \sqrt{\frac{2J_\ell}{\pi}}\cdot \mathrm{e}^{-\frac{1}{2}J_\ell\alpha^2}\,\mathrm{d}\alpha = \\ &= \sqrt{\frac{2J_\ell}{\pi}}\cdot\frac{2}{\sqrt{2J_\ell}}\cdot\int_0^\infty \mathrm{e}^{-t^2}\,\mathrm{d}t = \\ &= \frac{2}{\sqrt{\pi}}\int_0^\infty \mathrm{e}^{-t^2}\,\mathrm{d}t = \mathrm{erf}(\infty)\equiv 1 \quad \text{QED.}\end{aligned}$$

The resulting fitness function, given in equation (2.14), is simply the weighed average value of the summed membership over the different CI values.

$$m_\ell(X) = \int_0^\infty W_\ell(\alpha)\Phi_{\ell,X}(\alpha)\,\mathrm{d}\alpha \tag{2.14}$$

2.5 Feedback possibilities

The added value of calculating how well a measurement fits the membership function is twofold. Firstly, it can be used as classifier output. Secondly, equation (2.11) may be used to investigate if more accurate information on attribute values will reduce classification uncertainty and if so, how much more accurate the attribute value needs to be. Consider e.g., an object that is measured on two attributes ($J_1 = 2$), namely speed and altitude in table 2.1, and the membership field from figure 2.5.

Figure 2.8(a) shows how the membership is distributed over the various values of CI for the example from table 2.1. The best fit occurs at the boundaries of CI values between *0.5* and *0.85*. For larger values of CI the summed membership decreases, meaning that the combined mean values on both attributes do not fit the membership field.

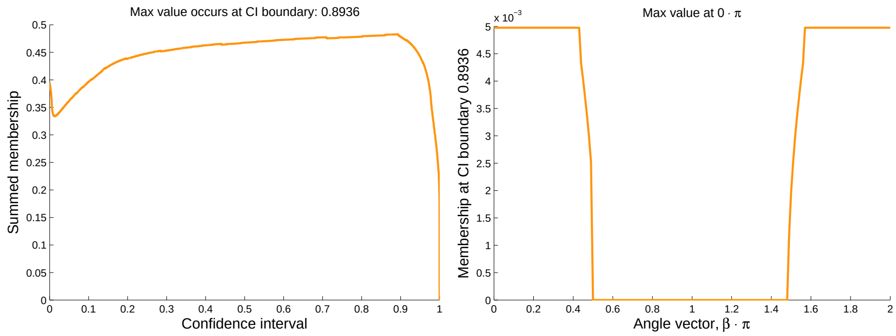
It is therefore interesting to see how the membership is distributed over possible attribute values on that contour line. This means that equation (2.10) needs solving for $J_1 = 2$ resulting in a description of a circle when $\leq$ is replaced by $=$, ${\xi_1}^2 + {\xi_2}^2 = 2\alpha^2$. Thus, the altitude at the outer edge of the CI becomes:

$$y_1(\alpha) = \mu_1 + \sigma_1\alpha\sqrt{2}\cdot\sin(\beta)$$

with $\beta \in [0, 2\pi]$.

Table 2.1: Measurements on Speed and Altitude

A_j	j	μ_j	$2\sigma_j$
Altitude	1	*1200*	*200*
Speed	2	*12*	*6*



(a) Membership for various CI values

(b) Membership values at $CI = 0.8936$ for the various $\beta \in [0, 2\pi]$

Figure 2.8: Membership value against CI and against β

The speed at the outer edge of the CI becomes:

$$y_2(\alpha) = \mu_2 + \sigma_2\alpha\sqrt{2} \cdot \cos(\beta)$$

with $\beta \in [0, 2\pi]$. Figure 2.8(b) shows the results using these descriptions where it becomes apparent that in this example the membership peaks around $\beta = 0$. Since $|\sin(0)| < |\cos(0)|$, a reduction in uncertainty on the speed is required most. From the CI value with maximum summed membership, the required uncertainty reduction can be calculated using equation (2.5) and the definition of ξ_j.

Using both these steps, the impact of the different attributes to how well the confidence interval fits the membership is determined. Similar work on attribute impact on results is found in e.g. [103]. In this manner the system learns which information is needed and the priority (through risk calculation) of that sensor task. This information can be fed to a sensor scheduler designed according to [104] or [51] to close the sensor control cycle. In this sense, the information needs are mapped on the information gathering capabilities, [46].

2.6 Extensions

The formulas to determine the CI in this chapter mostly assume Gaussian distributions on measurements and change when a different PDF is assumed on the measurement (or sensor) as was already stated in Section 2.4.2. Looking back at equation (2.3), now assume a Laplace distribution — which is given in equation (2.15) with mean value μ_j and variance $2{b_j}^2$ — on the independent measurements. In this case, the relation between the boundary value and CI from equation (2.3) turns to equation (2.16) for independent, Laplace distributed measurements.

$$p_j(y_j) = \frac{1}{2b_j}\,\mathrm{e}^{-\frac{|y_j-\mu_j|}{b_j}} \tag{2.15}$$

$$CI = \prod_{j=1}^{J_\ell}\left[\int_{\mu_j-\alpha b_j\sqrt{2}}^{\mu_j+\alpha b_j\sqrt{2}} \frac{1}{2b_j}\,\mathrm{e}^{-\frac{|y_j-\mu_j|}{b_j}}\,\mathrm{d}y_j\right] \tag{2.16}$$

Using basic mathematical operations equation (2.16) is reduced to equation (2.17). Note, the absolute value in the definition of the Laplace distribution may be left out due to the fact that it is a symmetrical function around the mean value. In order to do this the lower boundary in the interval is changed from $\mu_j - \alpha b_j\sqrt{2}$ to μ_j. Since this change reduces the results exactly in half, the integral is multiplied by *2*.

$$\begin{aligned} CI = \prod_{j=1}^{J_\ell}\left[\int_{\mu_j}^{\mu_j+\alpha b_j\sqrt{2}} \frac{1}{b_j}\mathrm{e}^{-\frac{y_j-\mu_j}{b_j}}\,\mathrm{d}y_j\right] = \\ = \prod_{j=1}^{J_\ell}\left[\int_0^{\alpha\sqrt{2}} \mathrm{e}^{-t_j^2}\,\mathrm{d}t_j\right] = \left(1-\mathrm{e}^{-\alpha\sqrt{2}}\right)^{J_\ell} \end{aligned} \tag{2.17}$$

Any combination of distributions could be used, e.g., for five independent sources of which two are Gaussian distributed and three are Laplace distributed, the CI is given by:

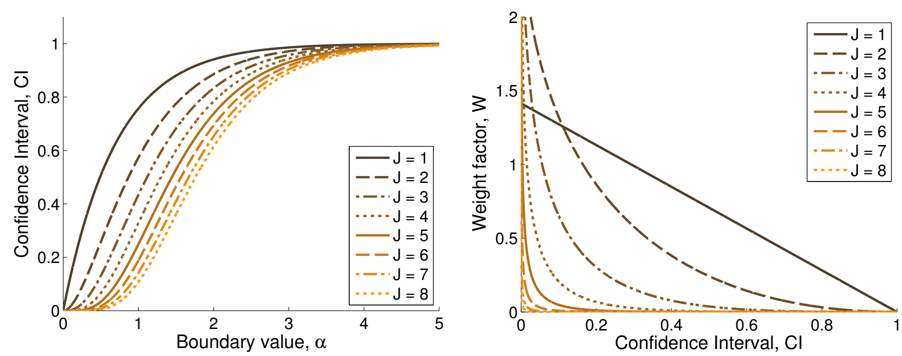
$$CI = \left(1-\mathrm{e}^{-\alpha\sqrt{2}}\right)^3\left(\mathrm{erf}\left(\frac{\alpha}{\sqrt{2}}\right)\right)^2 .$$

When all sources give Laplace distributed measurements, equation (2.9) changes to equation (2.18) which in turn simplifies to equation (2.19) based on $\alpha \in [0,\infty\rangle$. Possible attribute values are still defined as the mean and a fraction of the square root of the variance, for Laplace distributions $y_j = \mu_j + \xi_j b_j\sqrt{2}$.

The weight factor $W_\ell(\alpha)$, changes to equation (2.20) for Laplace PDFs , for which the property $\int_0^\infty W_\ell(\alpha) = 1$ can be easily shown. In figure 2.9, both the relation between Confidence Interval and the boundary value α and the weight factor and α are shown.

$$\prod_{j=1}^{J_\ell}\frac{1}{2b_j}\mathrm{e}^{-|\xi_j\sqrt{2}|} \geq \prod_{j=1}^{J_\ell}\frac{1}{2b_j}\mathrm{e}^{-\alpha\sqrt{2}} \tag{2.18}$$

$$\sum_{j=1}^{J_\ell}|\xi_j| \leq \alpha\cdot J_\ell \tag{2.19}$$



(a) Relation between CI and α for Laplace distributions with $J_\ell = \{1, 2, \ldots, 8\}$

(b) Weight factor against α for Laplace distributions with $J_\ell = \{1, 2, \ldots, 8\}$

Figure 2.9: Different values for Laplace distributions for CI and the weight factor

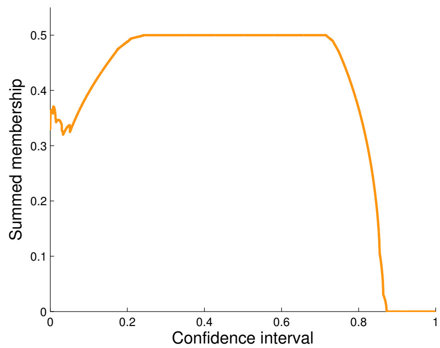
$$W_\ell(\alpha) = J_\ell\sqrt{2} \cdot \mathrm{e}^{-\alpha J_\ell \sqrt{2}} \tag{2.20}$$

From figure 2.9(b) it can be seen that more attributes combined, leads to less influence of uncertainty in those attribute values. This effect was also visible in Gaussian distributed measurements only not as strong. This can be explained by the excess kurtosis of both distributions. Gaussian distributions have an excess kurtosis equal to *0* whereas the excess kurtosis of a Laplace distribution is *3*, see [44]. This means the Laplace distributions have a stronger peak around the mean and fatter tails. This fact explains these effects. In general, more excess kurtosis in the underlying distribution of sensor measurements means less influence of uncertainty when combining enough sources.

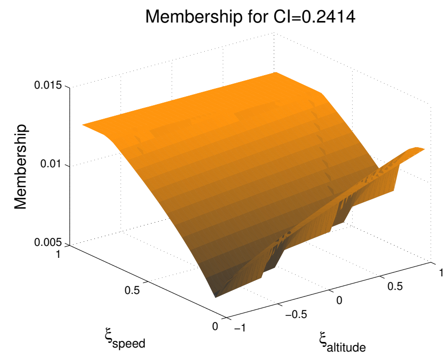
For the example of table 2.1 let us assume Laplace distributions instead of Gaussian ones. The b_j parameter is obtained by the variance, ${\sigma_j}^2 = 2{b_j}^2$. The results from figure 2.8 change to those shown in figure 2.10. These results are obtained by reducing equation (2.19) for two sources to

$$|\xi_1| + |\xi_2| = 2\alpha$$

for the relation between the fraction on altitude and that of speed. In the case of Laplace distributions, measurements are given by $y_j = \mu_j + \xi_j b_j \sqrt{2}$. This distribution leads to similar results. For both the Gaussian distributed measurements as well as the Laplace distributed measurements, a reduction in speed uncertainty reduces the most classification uncertainty. This becomes apparent when comparing the relation between the fractions ξ_1 and ξ_2 as shown in figure 2.11 for the example from table 2.1 and a confidence interval of *85%* for both cases.



(a) Membership for various CI values for Laplace distribution



(b) Membership values at $CI = 0.2414$ for the various fractions

Figure 2.10: Membership value against CI and against various fractions

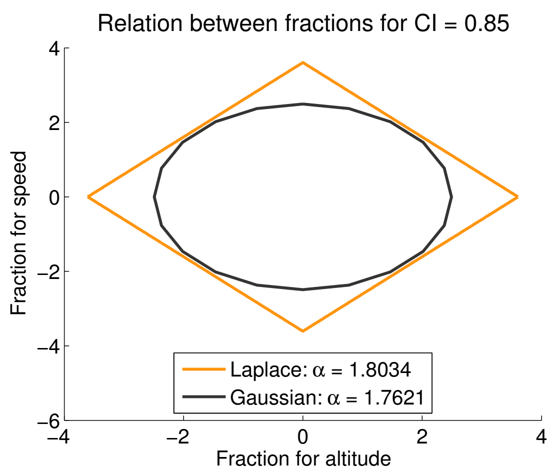


Figure 2.11: Relation between fractions for $J_\ell = 2$ and $CI = 0.85$ for Gaussian and Laplace distributions

From figure 2.11 can also be seen that two Gaussian distributed measurements lead to a circle and two Laplace distributed measurements lead to a rhombus (or diamond) for the fractions. Section 2.5 gave the equations for the circle. For the rhombus in the case of two attributes can be said that both polygon diagonals are perpendicular and that both are equal to 2α. Thus the opening angle of the rhombus is $\pi/4$ rad.

3 Combining classification belief

To believe in a contradiction is to abdicate your belief in the existence of the world around you and the nature of the things in it, to instead embrace any random impulse that strikes your fancy — to image something is real simply because you wish it were. A thing is what it is, it is itself. There can be no contradictions.

In reality, contradictions cannot exist. To believe in them you must abandon the most important thing you possess: your rational mind. The wager for such a bargain is your life. In such an exchange, you always lose what you have at stake.

Wizard's ninth rule – Chainfire (Ch.48)

DIFFERENT sources will disagree to some extent. Dealing with this disagreement — or conflict — is needed to determine a combined system belief. Besides reducing the conflict, it is also used as a feedback mechanism. In this sense, conflict is a valuable contribution for assessing tasks that are currently executed, anomaly detection and user interaction. The approach discussed in this chapter has appeared in [74], [76], and [78].

3.1 Choosing a combination rule

The previous chapter showed that for any subset of attributes a weighted average of membership can be calculated. Thus, L different classifiers express to what extent they believe that the object under consideration belongs to a certain classification label. Since there are now multiple classifiers expressing their belief, these beliefs need to be combined using a combination rule. Furthermore, these classifiers may provide different results and the chosen combination rule should therefore be able to cope with conflicting information.

Hunter and Liu describe a methodology to combine uncertain information from conflicting sources using possibility theory, [41]. Their approach only communicates labels to the operator after combining the information, whereas the approach using DSmT can communicate quantified solutions, or in this case: the soft classification results. Sekkas et al. solve a similar problem of combining context sensitive information using Fuzzy Logic, [90], with soft classification results. In their work fuzzy logic is used to assign confidence degrees to different sources but the combination of information of those sources is done probabilistically. Possibility theory requires additional information in the form of a-priori probabilities or a training phase. Both are therefore not directly applicable in the military domain.

Another possible combination rule is Dempster-Shafer theory, [20] and [91]. The Dempster-Shafer (DS) theory assume all elements in the solution space to be mutually exclusive and in Section 2.1 was shown that the classification solution space in this domain does not fulfil this assumption. Such a frame could be achieved by refining the solution space but this will have to be done repeatedly which is costly in terms of computation time. Furthermore, DS is created for independent and non-conflicting sources. The latter property does not apply. Different rules exist to overcome this problem but the question then becomes which of these is more suited. For now, the question which rule is the best to do this is which scenario is not answered.

Although any combination rule might be adapted to be applicable for combining classifier belief, here the choice is to use DSmT, [92] and [24], based on the conflict resolution possibilities within this theory and because it does not assume a mutually exclusive frame of discernment. Furthermore, besides the fusion model no additional rules are needed in DSmT when adding sources, whereas e.g., fuzzy logic, would requires no rules to take a new source into account.

3.2 Dezert-Smarandache Theory

The DSmT may be seen as an extension of DS theory, [20] and [91], since DS theory assumes an exclusive and exhaustive frame of discernment and DSmT assumes only exhaustiveness. The DS theory has been used in Artificial Intelligence (AI) applications since the early eighties of the 20th century. Besides assuming a non-exclusive frame of discernment, DS theory is not designed for combining highly conflicting data as can be seen in Zadeh's example, [114]. Since DSmT deals with non-exclusive classes and it can deal with conflicting information this framework is used throughout this work.

3.2.1 Free DSmT model

In the most general case, DSmT assumes that all possible combinations of labels from Θ using the $\cup$ and $\cap$ operators are possible. In order to combine the beliefs expressed by L different classifiers, a set is constructed that contains all labels themselves, all combinations of labels with both operators, and the classical empty set. This set is called the hyper-power set in DSmT and is denoted as D^{Θ}. This notation refers to Dedekind lattices, [17] and [16], since the construction of the hyper-power set D^{Θ} is closely related to Dedekind's problem, [21], [22], and [17].

From a certain frame of discernment Θ with I labels, the hyper-power set, D^{Θ}, can be constructed by three rules:

1. $\varnothing, \theta_1, \theta_2, \ldots, \theta_I \in D^{\Theta}$;
2. If A, $B \in D^{\Theta}$ then $A \cup B \in D^{\Theta}$ and $A \cap B \in D^{\Theta}$;
3. No other labels belong to D^{Θ} except those obtained using rules 1 and 2.

Using D^{Θ} the combination rule in equation (3.1) is defined to combine the beliefs from the L different classifiers where the belief held by classifier ℓ on a specific label is denoted $m_\ell(X_\ell)$ with $\ell \in \{1, 2, \ldots, L\}$. This combined belief is denoted $m_c^f(X)$ where the f refers to the free fusion model where all possible combinations from D^{Θ} are taken into account, [92]. Note that for the beliefs expressed by the classifiers $\sum_{X_i \in D^{\Theta}} m_c^f(X_i) = 1$ should hold, a normalisation of each classifier output is therefore required.

$$m_c^f(X) = \sum_{\substack{X_1, X_2, \ldots, X_L \in D^{\Theta} \\ X_1 \cap X_2 \cap \ldots \cap X_L = X}} \prod_{\ell=1}^{L} m_\ell(X_\ell) \tag{3.1}$$

Operators however, usually do not think in terms of generalised belief assignments. Plausibility and the credibility as defined in DS theory may be used instead. Both these quantities are also defined in the DSmT framework, [23]. Recently, Dezert and Smarandache proposed to expand the power set to a super-power set that also contains all combinations with the operators $\cap$, $\cup$, and the complement operator in [24]. This extension provides more possibilities for the application of DSmT. Due to the added computational complexity however we do not consider this new extension in this thesis.

Using the free fusion model implies that all possible combinations of labels are valid. Section 2.1 however stated that labels at the same specificity level are mutually exclusive, an object cannot belong to both the *helicopter* class and the *submarine* class. Based on the ancestral relations, equation 2.1, and based on this mutual exclusiveness within the same specificity level, a set of constraints, denoted $\varnothing_{\mathcal{M}}$, is constructed containing all combinations represented in D^{Θ} that are constrained given the application domain. The resulting fusion model $\mathcal{M}$

is defined as a *2*-tuple containing the frame of discernment Θ and the model constraints $\varnothing_{\mathcal{M}}$, $\mathcal{M} = \{\Theta, \varnothing_{\mathcal{M}}\}$. The free fusion model assumes $\varnothing_{\mathcal{M}} = \varnothing$ which is not the case in our application domain. Additional rules are required when the free model does not hold, $\varnothing_{\mathcal{M}} \neq \varnothing$.

3.2.2 Proportional conflict redistribution

Smarandache and Dezert have introduced several Proportional Conflict Redistribution (PCR) rules in [93] to deal with constraints placed on the free model. In [62], Martin and Oswald describe the generic PCR6 rule of combination, given in equation (3.2), for the DSmT framework. This rule enables us to impose the model constraints $\varnothing_{\mathcal{M}}$ from the fusion model $\mathcal{M}$.

$$m_c^{\text{pcr6}}(X) = m_c^f(X) + \sum_{\ell=1}^{L} \mathcal{G}_\ell(X) \cdot m_\ell(X)^2 \tag{3.2}$$

In equation (3.2) the term $\mathcal{G}_\ell(X)$ denotes the factor that makes the redistribution of the conflicting mass of source ℓ to a label $X \in D^\Theta$ proportional to the original contribution to the conflict. It is given by equation (3.3) and explained in more detail with a small example in Appendix A.3.

$$\mathcal{G}_\ell(X) = \sum_{\substack{\bigcup_{u=1}^{L-1} X_{\varphi_\ell(u)} \cap X \in \varnothing \\ X_{\varphi_\ell(1)},\ldots,X_{\varphi_\ell(L-1)} \in \left(D^\Theta\right)^{L-1}}} \frac{\prod_{w=1}^{L-1} m_{\varphi_\ell(w)}\left(X_{\varphi_\ell(w)}\right)}{m_\ell(X) + \sum_{w=1}^{L-1} m_{\varphi_\ell(w)}\left(X_{\varphi_\ell(w)}\right)} \tag{3.3}$$

The term $\varphi_\ell(l)$ ensures that all labels from the hyper-power set are used except label ℓ — the label under consideration — and it is given by equation (3.4). In [62] this function is denoted $\sigma_\ell(l)$, whereas here $\varphi_\ell(l)$ is used to avoid confusion with the notation of the standard deviation for Gaussian measurements.

$$\varphi_\ell(l) = \begin{cases} \varphi_\ell(l) = l & \text{if } \; l < \ell \\ \varphi_\ell(l) = l+1 & \text{if } \; l \geq \ell \end{cases} \tag{3.4}$$

This combination rule can be implemented using Algorithm *3* from [62] which is freely available on-line and given in algorithm 1. Note that in algorithm 1 the operator $\times$ is used to denote the Cartesian product, [107]. Furthermore, the PCR6 rule only works when $\varnothing_{\mathcal{M}} \cap \Theta = \varnothing$ holds.

Algorithm 1: Pseudo code for the PCR6 rule of combination from [62]

```
Data   : k sources S : S[i],...,S[n]
Result : Fusion of S by PCR6, ep

for i = 1 to k do
    foreach c in S[i] do
        APPEND c to cl[i]
foreach ind in [1,size(cl[1])] × ... × [size(cl[k])] do
    c ← s ∩ ind
    if s ≡ ∅ then
        lconf ← 1; sum ← 0
        for i=1 to k do
            lconf ← lconf * S[i](cl[i][ind[i]])
            sum ← sum + S[i](cl[i][ind[i]])
        for i=1 to k do
            ep(S[i][ind[i]]) ← ep(S[i][ind[i]]) + S[i](cl[i][ind[i]]) * lconf/sum
    else
        lconf ← 1
        for i = 1 to k do
            lconf ← lconf * S[i](cl[i][ind[i]])
            ep(s) ← ep(s) + lconf
```

3.2.3 Belief metrics

The plausibility of X is the sum of all masses from labels that have partial or full agreement with X and it is given by equation (3.5). The credibility of X is the sum of masses from all the sub propositions (full agreement) of X and is given in equation (3.6). From the definition of the credibility follows that for all classes X with highest specificity $m_c(X) = \mathrm{Bel}(X)$ holds, since they have no sub propositions in the frame of discernment.

$$\mathrm{Pl}(X) = \sum_{\substack{X_i \cap X \neq \varnothing \\ X_i \in D^\Theta}} m_c(X_i) \tag{3.5}$$

$$\mathrm{Bel}(X) = \sum_{\substack{X_i \subseteq X \\ X_i \in D^\Theta}} m_c(X_i) \tag{3.6}$$

Plausibility sums all belief in partial agreement with X and $1 - \mathrm{Pl}(X)$ therefore gives all evidence in total disagreement with X. This value can be used e.g., as an indication for how much uncertainty resides in a label. For classes at a lower specificity level the difference between the credibility and plausibility gives more information on the confusion that still exists.

A single value for the possibility that the object under consideration belongs to a certain class instead of a range of values like Bel(X)–Pl(X) is desired by the operators. Note however, that during interviews operators indicate that a visualisation of combined generalized belief assignments on labels is hard for them to interpret. Another visualisation might therefore needed in practical implementations.

$$\mathrm{BetP}(X) = \sum_{X_i \in D^\Theta} \frac{\mathcal{C}_\mathcal{M}(X \cap X_i)}{\mathcal{C}_\mathcal{M}(X_i)} \cdot m_c(X_i) \tag{3.7}$$

In [25], the probabilistic transform as defined in DS is re-written for the DSmT framework. In DSmT the resulting quatity is called the pignistic[1] probability, following the notation of Smets [96]. It is denoted *BetP* and given in equation (3.7), where $\mathcal{C}_\mathcal{M}(X)$ denotes the DSm cardinality of label X given the fusion model $\mathcal{M}$ under consideration. The notion of DSm cardinality is discussed in [21].

In general

$$m_c(X) \leq \mathrm{Bel}(X) \leq \mathrm{BetP}(X) \leq \mathrm{Pl}(X)$$

holds $\forall X \in \Theta$. The pignistic probability might be a suitable quantity to visualise as a means to communicate with the operator, although we note that more testing is required to determine if this is the case.

$$\mathrm{DSmP}_\epsilon(X) = \sum_{X_i \in D^\Theta} \left[\frac{\sum\limits_{\substack{X_l \subseteq X \cap X_i \\ \mathcal{C}_\mathcal{M}(X_l)=1}} m_c(X_l) + \epsilon\, \mathcal{C}_\mathcal{M}(X \cap X_i)}{\sum\limits_{\substack{X_l \subseteq X_i \\ \mathcal{C}_\mathcal{M}(X_l)=1}} m_c(X_l) + \epsilon\, \mathcal{C}_\mathcal{M}(X_i)} \cdot m_c(X_i) \right] \tag{3.8}$$

Dezert and Smarandache introduce a different probabilistic transform in [23] which they denote DSmP_ϵ, it is given in equation (3.8). This DSmP_ϵ value uses the assigned masses on labels with a DSm cardinal of one and only slightly takes the cardinal itself into account since they propose to use either $\epsilon = 0$ or in case that is not possible due to the fusion model $\epsilon = \frac{1}{1000}$.

The conversion to probabilities facilitates communication with Bayesian approaches. Where the system itself operates on belief masses, which is referred to as the credal level, the operator and/or other subsystems may view the calculated probabilities on the pignistic level (values derived of values on the credal level such as credibility, plausibility and probabilities). Both the credal and pignistic levels are described in the Transferable Belief Model (TBM) as described in [95] by Smets and Kennes, where they use the pignistic level for decision making.

[1] from the Latin *pignus* for *bet*

Since the system works only on the credal level and the functions to switch to the pignistic level are given the rest of this chapter only addresses belief at the credal level.

3.2.4 Conditioning of belief

Additional information about the frame of discernment may lead to the desire to redistribute belief. This may be achieved by Shafer's conditioning rules or the belief conditioning rules as defined for DSmT. Both methods are briefly discussed.

Shafer's conditioning rules

In [91], a conditioning scheme is proposed to deal with additional information. When a new source indicates that the true solution lies in $X_{\text{true}} \in \Theta$, a new credibility and plausibility are determined $\forall X \in \Theta$ according to equations (3.9) and (3.10) respectively where $\overline{X_{\text{true}}}$ denotes not X_{true}. Although this approach is simple to implement and to understand, it is not considered to be objective enough as discussed in [94].

$$\text{Bel}_{\text{new}}(X \mid X_{\text{true}}) = \frac{\text{Bel}(X \cup \overline{X_{\text{true}}}) - \text{Bel}(\overline{X_{\text{true}}})}{1 - \text{Bel}(\overline{X_{\text{true}}})} \tag{3.9}$$

$$\text{Pl}_{\text{new}}(X \mid X_{\text{true}}) = \frac{\text{Pl}(X \cap X_{\text{true}})}{\text{Pl}(X_{\text{true}})} \tag{3.10}$$

In short, this condition rule uses a new source with $m_{\text{new}}(X) = 1$ and then uses Dempster's rule of combination to combine it with the previous held belief. The advantage of this approach is that the new source may even indicate that belief should not be held in a region of Θ.

Belief conditioning rules

In DSmT additional information on where the truth is, can be enforced using the Belief Conditioning Rules (BCRs) as explained in [94]. The numerous variations are all based on three subsets of D^Θ that are constructed using three rules. These three subsets are denoted D_1, D_2, and D_3.

The subset D_1 contains the combination of all labels that are used in the description of where the truth lies. To denote this set of labels, [94] defines the function $s(X_{\text{true}})$ when the truth lies in X_{true}. E.g., when the truth lies in element $X_{\text{true}} = X_1 \cap X_2 \cup X_5$, then $s(X_{\text{true}}) = \{X_1, X_2, X_5\}$. The subset D_1 contains all combinations of the involved labels that are returned by function $s(X_{\text{true}}))$ using the $\cap$ and $\cup$ operators as well as the those labels themselves.

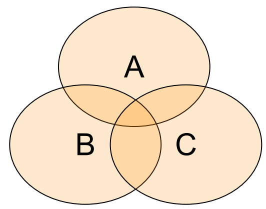


Figure 3.1: A generic free world fusion model common in DSmT

The second subset, D_2, is the sub-hyper-power set generated with all labels from $\Theta \backslash s(X_{\text{true}})$ and the $\cap$ and $\cup$ operators when the truth lies in X_{true}.

Finally, the third subset contains all elements from $D^{\Theta} \backslash \varnothing$ that are not represented in D_1 and D_2. This set is defined as $D_3 = (D^{\Theta}) \backslash (D_1 \cup D_2)$. All three subsets have no element in common two by two and their union is $D^{\Theta} \backslash \varnothing$.

Consider e.g., the free model of figure 3.1 and let the truth be in $A \cup B$. The different disjoint subsets then become:

- $D_1 = \{A, B, A \cup B, A \cap B,\}$ and all combinations contained in these elements like e.g., $A \cap B \cap C$;
- $D_2 = \{C\}$ since $s(A \cup B) = \{A, B\}$ and therefore $\Theta \backslash s(A \cup B) = C$;
- $D_3 = \{A \cup C, B \cup C, A \cup B \cup C, C \cup (A \cap B)\}$.

All BCRs are based on redistribution of the masses in D_2 and D_3 to elements in D_1. For BCR1 this is done by proportionally redistributing the combined mass from D_2 and D_2 to the elements in D_1. For the other rules, BCR2–31, redistribution is done directly to particular elements in D_1 or it is done from disjoint subsets of D_2 or D_3 to D_1 and variations thereof, for details see [94].

3.3 User preferences

Classifier beliefs can be combined using PCR6 as shown in [63] for a fully automated classification problem. When the operator however plays a part in the classification process additional rules are required. The operator (or user) may exert his influence in two ways:

1. the operator is an information source and
2. the operator can place additional constraints.

The first type of interaction can be handled by PCR6. The constraints placed by the user, denoted $\varnothing_{\mathcal{U}}$, however necessitate additional rules since $\varnothing_{\mathcal{U}} \cap \Theta \neq \varnothing$ and PCR6 does not facilitate these types of constraints, see Section 3.2.2. The

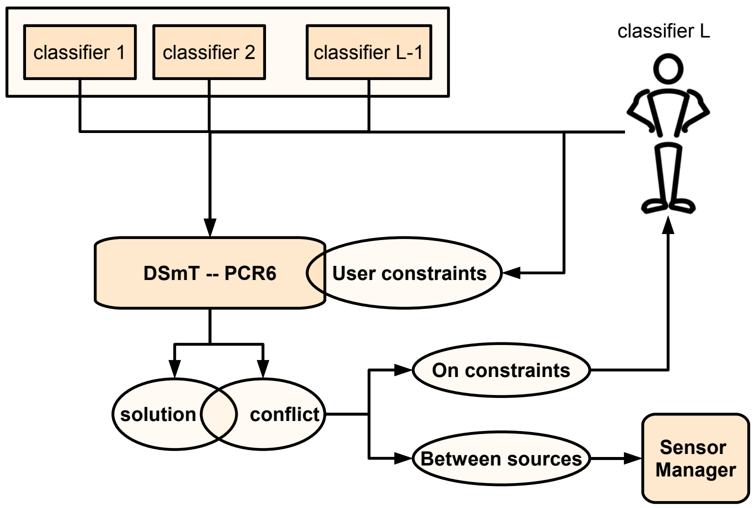


Figure 3.2: System architecture for automated classification and classification support

main difference between $\varnothing_{\mathcal{M}}$ and $\varnothing_{\mathcal{U}}$ is that $\varnothing_{\mathcal{M}} \cap \Theta = \varnothing$ whereas $\varnothing_{\mathcal{U}} \cap \Theta \neq \varnothing$ when $\varnothing_{\mathcal{U}} \neq \varnothing$. Figure 3.2 depicts the resulting system architecture to achieve the required user interaction.

Besides the user and the MBCs, any classification solution may be incoorperated in the system architecture from figure 3.2. Van der Broek et al. e.g., describe a classification system for sea based objects in [105] based on information received from electro-optical sensors. Although their work focusses on a three class problem with a jet-ski, a water-taxi, and a RHIB[2], it can be used by adding those three labels in the model as described in Section 2.1 at the appropriate specificity level(s).

3.3.1 Parents and bridges

In [94], several BCRs are proposed to deal with additional information, irrespective of the source of that information. However, these BCRs deal with the situation where a source indicates where belief should be held. Here, a different conditioning rule is introduced that deals with the situation where labels from the frame of discernment are excluded. The excluded labels are contained in $\varnothing_{\mathcal{U}}$. The conflict $\mathcal{K}$ that an element $X \in \varnothing_{\mathcal{U}}$ introduces, is the amount of belief

[2]Rubber Hull Inflatable Boat

it was originally assigned to by the used combination rule, equation 3.11. The total amount of conflict that is introduced by the operator is determined by summing all mass originally assigned to $X \in \varnothing_{\mathcal{U}}$ which has a value on $[0,1]$ and this value can be used to inform the operator how much conflict is introduced by $\varnothing_{\mathcal{U}}$, equation 3.12.

$$\mathcal{K}(X) = m_c(X) \tag{3.11}$$

$$\mathcal{K}_{\text{total}} = \sum_{X_i \in \varnothing_{\mathcal{U}}} \mathcal{K}(X_i) \tag{3.12}$$

Since the operator says that these labels should be excluded from the model, the new User Preference Redistribution (UPR) rule states that all these constrained labels, and of course their children, are assigned zero belief, i.e., $m_c^{\text{upr}}(X) = 0,\ \forall X \in \varnothing_{\mathcal{U}}$. In order to maintain validity in the operator constraints, $\forall X \in \varnothing_{\mathcal{U}}$ there is no $X_i \in \Theta \backslash \varnothing_{\mathcal{U}}$ for which $X_i \cap X = X_i$ holds. Since

$$\sum_{\forall X_l \in D^{\Theta}} m_c^{\text{upr}}(X_l) = 1$$

should still hold, the masses that are discarded need to be redistributed. The question becomes, where should it be redistributed to?

The first choice is to redistribute the conflicting mass to one of the parent labels of the excluded one with smallest DSm cardinality that is not excluded itself. In order to find the set of parent labels, $X^{\uparrow}$ is defined as the set containing all full parent labels of X. This set joins all labels $X_i \in \Theta$ for which $X \cap X_i = X$ holds, see equation (3.13) where the join operator is denoted $\bowtie$. Note that the difference with the definition of parent elements from Section 2.1 where all elements with lower specificity with a non-empty intersection were considered parent elements.

$$X^{\uparrow} = \bigotimes\{X_i \in \Theta \mid X_i \cap X = X\} \tag{3.13}$$

$$X^{\sqcap} = \bigotimes\{X_i \in \Theta \mid X_i \cap X \neq \varnothing \ \wedge\ X_i \cap X \neq X\} \tag{3.14}$$

$$X_i^{\sqcap} \bowtie X_i^{\uparrow} = \bigotimes_{a=1}^{K} \theta_{\Omega^{-1}(i)}^{\uparrow a} \tag{3.15}$$

$$X^{\rightarrow} = \bigotimes\{X_i \in \Theta \mid X_i \cap X^{\sqcap} \neq \varnothing \ \wedge\ X_i \cap X = \varnothing\} \tag{3.16}$$

The set of bridging labels of X, denoted $X^{\sqcap}$, is determined by joining the labels that have a non-empty intersection but that are not fully enclosed by X, equation (3.14). Now, all parent elements as defined in Section 2.1 are accounted for since equation (3.15) holds $\forall X_i \in \Theta$.

It could be that all these bridges are constrained by $\varnothing_{\mathcal{U}}$ as well, $X^{\sqcap}\backslash\varnothing_{\mathcal{U}} = \varnothing$. In this case an unconstrained label, $X^{\rightarrow}$, is found by using constrained bridges, equation (3.16). Should it occur that even this $X^{\rightarrow}\backslash\varnothing_{\mathcal{U}} = \varnothing$, $\mathcal{K}(X)$ is then redistributed to $\Theta\backslash\varnothing_{\mathcal{U}}$.

Four possible areas where the mass could be redistributed to are identified. The area that will be used for the redistribution of the belief assigned to the now constrained label X, denoted X^*, is determined by equation (3.17). In essence, the first choice is to use the smallest full parent. If all full parents receive negative confirmation from the user, bridges are chosen. Should these also receive negative confirmation, the bridges are used to find generic labels that have no intersection with the constrained label. Finally, should these labels also receive negative confirmation, the mass is redistributed to all labels from Θ that are not in $\varnothing_{\mathcal{U}}$.

Equation (3.17) determines where conflicting mass is to be redistributed. Additionally, the most computationally intensive part — i.e., calculating $X^{\uparrow}$, $X^{\sqcap}$, and $X^{\rightarrow}$ — can be done off-line $\forall X \in \Theta$ in contrast to BCR where the disjoint sets need to be constructed based on $\varnothing_{\mathcal{U}}$ and thus at run-time. When UPR is used in an on-line system and $\varnothing_{\mathcal{U}} \neq \varnothing$, equation (3.17) can be run with low computational costs.

Having identified where conflict should be redistributed to, the next step is to determine how to redistribute it.

$$X^* = \begin{cases} \min\limits_{X_i \in (X^{\uparrow}\backslash\varnothing_{\mathcal{U}})} \mathcal{C}_{\mathcal{M}}(X_i) & \text{if } X^{\uparrow}\backslash\varnothing_{\mathcal{U}} \neq \varnothing \\ \bigcup\{X_i \mid X_i \in (X^{\sqcap}\backslash\varnothing_{\mathcal{U}})\} & \text{if} \begin{cases} X^{\uparrow}\backslash\varnothing_{\mathcal{U}} = \varnothing \\ X^{\sqcap}\backslash\varnothing_{\mathcal{U}} \neq \varnothing \end{cases} \\ \max\limits_{X_i \in (X^{\rightarrow}\backslash\varnothing_{\mathcal{U}})} \mathcal{C}_{\mathcal{M}}(X_i) & \text{if} \begin{cases} X^{\uparrow}\backslash\varnothing_{\mathcal{U}} = \varnothing \\ X^{\sqcap}\backslash\varnothing_{\mathcal{U}} = \varnothing \\ X^{\rightarrow}\backslash\varnothing_{\mathcal{U}} \neq \varnothing \end{cases} \\ \Theta\backslash\varnothing_{\mathcal{U}} & \text{otherwise} \end{cases} \tag{3.17}$$

3.3.2 User preference redistribution

Only conflict needs to be redistributed so each element that is unconstrained keeps the belief value that was assigned to it by PCR6. Belief mass is added to the belief assignments on those labels obtained by equation (3.17). Conflicting mass is redistributed to all elements within that area proportional to their assignment based on PCR6. This proportionality is applied to ensure that initial differences between elements are kept after redistribution. If e.g., two elements were assigned *0.01* and *0.02* belief mass and *0.1* is to be redistributed to them. If this would be done by given both labels the same amount of the conflict it would result in *0.06* and *0.07* belief mass. Since this is a distortion of the original difference in belief assignment it is redistributed proportionally to their initial proportions, obtaining *0.043* and *0.086*. In this way, the fact that one element was assigned twice as much belief as the other is maintained after applying UPR.

The redistribution is done $\forall Y \in \varnothing_{\mathcal{U}}$, obtaining equation (3.18) for the UPR rule which is defined $\forall X \in D^{\Theta} \backslash \varnothing_{\mathcal{U}}$.

$$m_c^{\mathrm{upr}}(X) = m_c(X) + \sum_{\substack{Y^* \cap X = X \\ \forall Y \in \varnothing_{\mathcal{U}}}} \left[\mathcal{K}(Y) \cdot \frac{m_c(X)}{\sum\limits_{\substack{X_i \cap Y^* = X_i \\ \forall X_i \in D^{\Theta}}} m_c(X_i)} \right] \quad (3.18)$$

3.4 Describing information need

Through PCR6, the user and the sensor manager can be notified of conflicts between sources, [76]. If a conflict exists between automated classifiers, results from Section 2.5 can be used to determine on which attribute the uncertainty needs to be reduced and by how much this uncertainty needs to be reduced. A sensor function request is made based on the desired measurement of an attribute which is sent to the sensor manager. Additional sensor functions are then started to reduce the uncertainty on the desired attribute, resulting in a system as shown in figure 3.2.

New sensor measurements might not reduce uncertainty to the extent expected by the system based on the available sensor models. This is an indication that the sensor performance is degraded and can be used to trigger the maintenance crew and/or automated diagnostic functions.

Systems where sensor performance is monitored on-line are not uncommon. Wei et al. describe a similar system in [106] where the sensor measurements are used to reason on the sensor state using particle filters. To model different sensor states they use a Markov model with discrete states. Our approach however is able to deal with sensor performance degradation in a more continuous quantified manner.

In [30], Erdnic et al. also reason on sensor performance by comparing the information from multiple sensors that measure similar attributes to reason on sensor degradation in terms of *pass/fail*. The main difference with the approach described in this thesis is that Erdnic et al. describe a system with multiple similar sensors whereas here a system with fewer, and more dissimilar sensors is assumed.

When a conflict remains after initial on-line system diagnostic checks, other possibilities need to be considered. Unlike [34] and [90] where a conflict is always assumed to be caused by the unreliability of sources, here we consider that there might be either:

1. Something wrong with the automated classifiers;
2. The operator is mistaken;
3. An object does belong to a certain class but it is behaving unaccordingly.

Which one of these possible causes is actually the case needs to be investigated. In current systems there is no indication at all on this type of conflict. In interviews operators indicate that the trigger that a conflict occurs is considered a desirable new feature in the CMS. Especially the third possible option is interesting for anomaly detection. This is becoming important when considering the new mission types with many a-symmetrical threats.

3.5 Anomaly detection

Anomaly detection can be achieved in two different ways. The first method for anomaly detection is conflict between two — or more — classification information sources. E.g., an operator classifies an object as a fishing vessel but the automated classifier based on course and speed classifies it as a fast patrol boat. This can indicate that although the vessel is a regular fishing boat, its behaviour is atypical as also explained in section 3.4. Using conflict between sources for anomaly detection places constraints on the combination rules that may be applied for fusion. Approaches like the one from Xin et al. in [111], where DS rule is adapted for highly conflicting sources by throwing away the most conflicting source data e.g., become less usable, whereas using PCR6 is still an option.

Secondly, it can be done within a single classifier by taking temporal aspects into consideration. E.g., an airliner will adhere to given air lanes and ferries have dedicated routes they follow. If objects, classified as such, suddenly no longer adhere to these expected areas the system is triggered. Of course, this principle is not restricted to position but can be expanded to any subset of attributes.

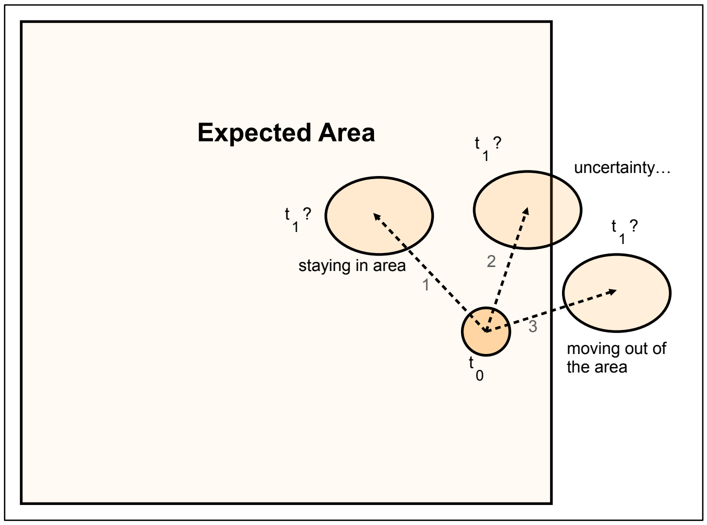


Figure 3.3: An object may be staying in, moving in, or moving out off the expected area but sometimes you cannot be sure

In figure (3.3) the principle of adherence to expected areas in shown. At a given time the position of an object is within the expected area with a given course and speed. We describe three different situations in this scenario for the estimated position at the next time step:

1. it completely falls within the expected area;
2. it partly falls within the expected area; or
3. it falls outside the expected area.

The first case is the most simple: the object is staying within the expected area and therefore the initial classification solution is supported. The third case indicates that the initial solution could very well be wrong since the object is certainly moving outside the expected area. The second case brings more uncertainty: since it is uncertain if the object is staying within the area or not thus increasing classification uncertainty. A practical example of this type of reasoning is the adherence to air lanes in the classification of air targets. Objects moving out of the air lane without an apparent cause are immediately indicated as *suspect* and investigated further. The same principles can also be used on e.g., known fishing grounds, sea lanes and tourist attractions.

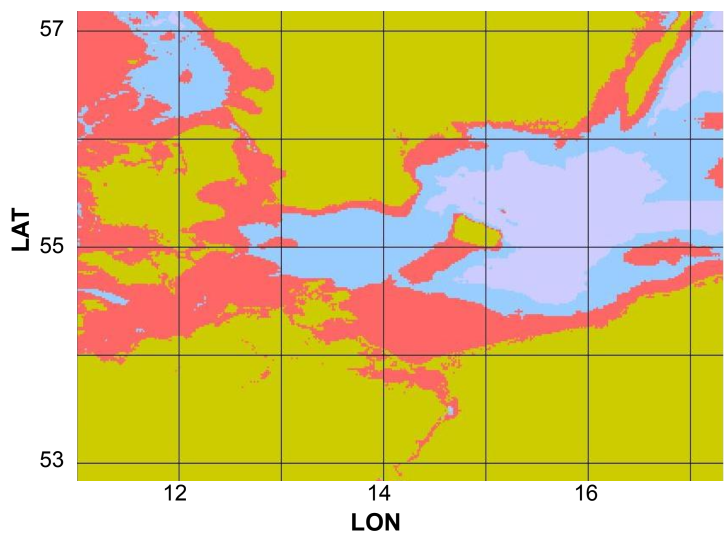


Figure 3.4: Mission area in which the waters with a water depth $\leq$ 25m has been highlighted in red

Another example of normal behaviour is the relation between speed and water depth for ships. The maximum speed of a ship in shallow waters is discussed in [32]. Knowing where a ship is sailing given its speed therefore excludes possible classification solutions. For mission planning, specific areas can be highlighted based on the water depths where certain ships would typically not occur. Figure 3.4 shows a region in which the waters with a water depth $\leq$ 25m are highlighted in red. Ships sailing at high speeds in those areas are therefore most likely small. Furthermore, the course and speed of ships are related to each other in certain areas with a lot of traffic, [57], helping the classification process as well as enabling anomaly detection.

Lensen et al. describe a system for electro-optical sensors that might be used for anomaly detection, [59]. A low resolution system with a wide field of view is used for general picture compilation and a high resolution narrow field of view camera is used for interesting objects to obtain more information. This high resolution image is fused into the lower resolution picture. The principle used in [59] is similar to the approach used in this thesis. Lensen et al. propose to use the most accurate sensor for objects that seem most interesting where we propose to use sensors to obtain the most relevant information first.

4 Sensor deployment

The past can teach us, through experience, how to accomplish things in the future, comfort us with cherished memories, and provide the foundation of what has already been accomplished. But only the future holds life. To live in the past is to embrace what is dead. To live life to it's fullest, each day must be created anew. As rational, thinking beings we must use our intellect, not a blind devotion to what has come before, to make rational choices.

Wizard's seventh rule – Pillars of Creation (Ch.60)

Based on the expected threat, surveillance tasks are executed by appropriate sensors from the sensor suite. Executing surveillance tasks may lead to detections, that in turn require execution of more sensor tasks like tracking, classification and such. The more dangerous an object might be, the less uncertainty about the information about that object is acceptable in order to take counter measures. More sensor tasks are then needed to reduce that uncertainty. The process of *generating sensor task requests* (or task generation for short) is therefore vital to sensor management.

Once the system knows which tasks to perform and what their priorities are, the most suitable of the available sensors is chosen. Sensor performance needs to be predicted based on the task and the mission in order to schedule the tasks for the available sensors. This chapter discusses the principles of sensor management and proposes a new sensor manager. The work presented in this chapter has appeared in the following publications: [75], [50], [51], [104], and [58].

4.1 Sensor management

Many methodologies have been proposed for sensor management, see e.g., [97], [79] and [4]. In general they can be divided in three categories:

1. Multi Function RADAR (MFR), see e.g., [39], [27], [99], and [110];
2. Strategies to optimally use multiple sensors, see e.g., [3] and [42];
3. Selection of resources, see e.g., [115], [45], and [26].

Besides the management of sensor systems, research is also done in general planning and scheduling techniques, see e.g., [101] and [5]. The introduction of the MFR has led to specific management strategies, see e.g., [97], [39], [27]. In [72] a comparison is made between these methodologies and proposes a sensor scheduler based on fuzzy Lyapunov synthesis as proposed by Margialot and Langholz for a different problem domain in [61]. This scheduler can be used for all three categories that can be distinguished in sensor management. It does however still depend on a system to define which tasks to perform and to describe the constraints under which to perform those tasks.

In [8] and [10] Bolderheij et al. describe a sensor management concept that is based on mission information. Based on this mission and the sensor observation new considerations about the sensor deployment can be made. Especially the notion that the expected threats determine how surveillance capabilities need to be deployed is interesting for closing the sensor control cycle. Bolderheij et al. also describe a mechanism to prioritise different sensor tasks in [9] using risk assessment based on risk (R) as defined by Yellman in [113]. Here, this approach is used as a basis to close the sensor control cycle.

The previous chapters already indicated how the available sensor information can be used to determine how much uncertainty needs to be reduced. Sensor task requests can be composed based on the information need. This maximises situation awareness since the most relevant information is being requested from the sensor suite. Figure 4.1 shows the sensor manager that is based on this principle.

As soon as the sensor manager knows:

- which information is needed;
- which sensor tasks can be used to obtain that information;
- which sensor settings are required to execute the sensor tasks; and
- which sensor settings are constrained by the mission;

the sensor manager can start to allocate tasks to sensors and create a schedule for each sensor. In the final stage the settings of the sensor suite will have to be

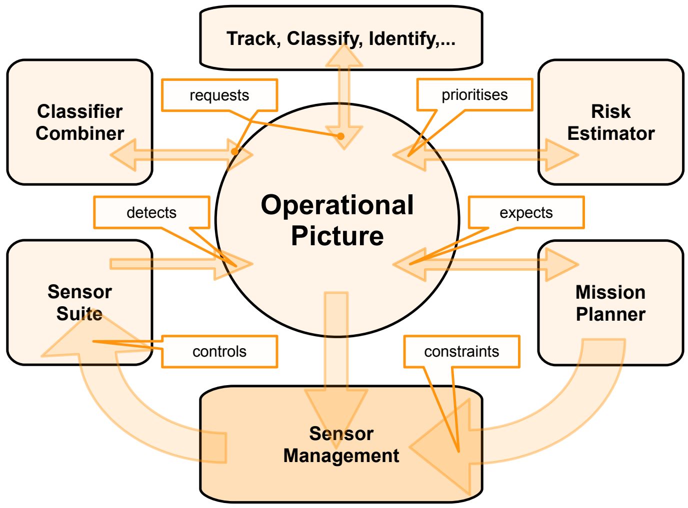


Figure 4.1: Mission planning defines constraints and surveillance, whereas other C2 processes determine the information requirements and risk estimation assigns priorities for the sensor management

adjusted to execute the assigned tasks as well as possible given the information requirements and constraints.

Sensor management requires various kinds of knowledge to appropriately allocate tasks and sensor settings to each sensor: knowledge about the sensor systems, about the mission, and about the environment, both meteorological and geographical. Combined with knowledge about the task at hand, sensor performance can be predicted. This sensor performance prediction is used as a measure of suitability of the sensor for the task at hand. With measures for sensor suitability, each task can be allocated to a sensor after which the sensor scheduling can start. Sensor controls are determined in the final step.

In general, the sensor management approach used in this thesis consists of three steps: allocating, scheduling, and controlling. This in accordance with [8] and [36] that both propose this three-stage sensor manager.

In the allocation process it is important to map the specific characteristics of the sensor task to the capabilities of the available sensors. We therefore first look at process where sensor task requests are generated after which we will look at the suitability of sensors for specific sensor tasks.

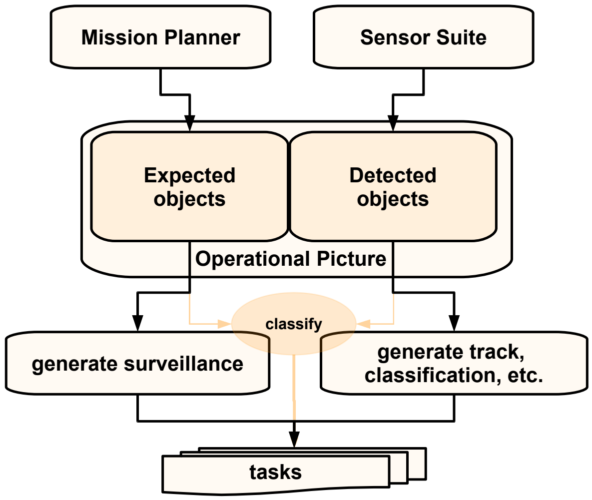


Figure 4.2: In task generation a distinction is made between task requests based on expected objects and those based on already detected objects; for the latter, results from the classification process are taken into account

4.2 Task request generation

Sensor tasks need to be executed to reduce the uncertainty on attribute information of objects in the operational picture as presented to the operators. Generation of sensor task requests can be split into two processes. The first being the generation of tasks for expected objects: being the surveillance capabilities. Second is the generation of tasks for already detected objects: track, classify and identify capabilities. Figure 4.2 shows this principle.

4.2.1 Expected objects

How surveillance capabilities are to be deployed depends on the expected threat in the mission area. The risk an expected threat poses to the mission naturally depends on the mission success criteria. Currently, the primary warfare officers make a sensor plan to detect manifestions of the expected threat. This is done by setting search sectors and choosing the modes of operation for the various sensors. Mission-driven sensor management as proposed in [7] can be used to support this process. As a part of this management system, a mission planning tool is required. Using that planning tool, the command team on-board only inserts the threat they expect. A sensor deployment plan is then determined using the determination of information needs as described in this thesis and [104].

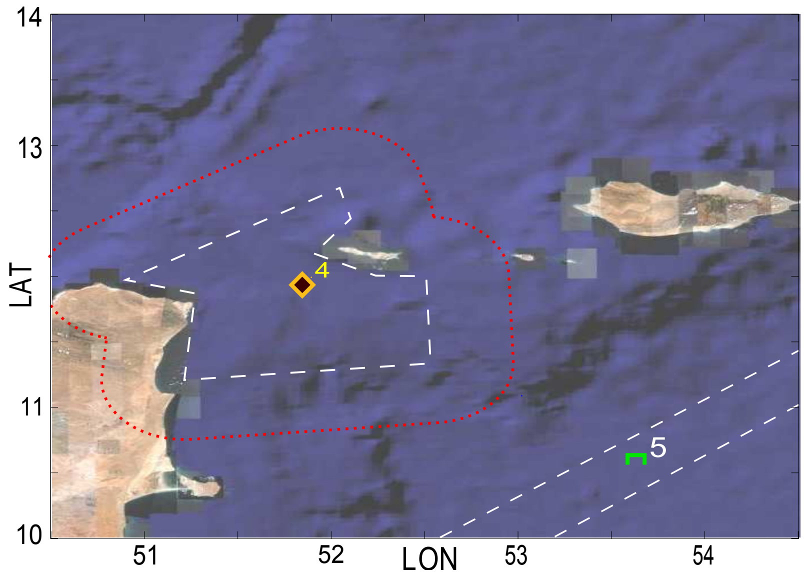


Figure 4.3: Mission planner with an expected fast patrol boat (object *4*) and an indicated air lane (object *5*)

Such a mission planner needs not to be very complex. A graphical interface showing geographical information and interface to insert objects and areas combined with the own navigation plans should suffice for the purpose of sensor management. Functionality may be added later to play what-if scenarios to offer more support in planning the deployment of assets. It is however the authors view, that such functionality should be developed based on the (simulated) operational picture and that it should not be integrated in the mission planner itself to ensure portability of the various software components.

Figure 4.3 shows a screenshot of the mission planner that was built during this research. In the scenario shown the the screenshot, a fast patrol boat (object number *4*) is expected in a certain area, which is indicated with the white lines. The weapon range (the red line) of the fast patrol boat can be visualised since this classification is known to the system. The weapon range is used as a trigger to start searching for a fast patrol boat in that region when the area where it is expected comes within sensor range. Furthermore, surveillance directed against the type of weapon that is used by the fast patrol boat is started when the own ship comes within possible weapon range. Added value of this visualisation is that during the planning phase the navigation plan could be changed based on such weapon ranges. Altering navigation plans based on weapon range information is shown for Unmanned Aerial Vehicles (UAVs) in e.g., [100].

The surveillance task to detect a fast patrol boat is based on the size of the area where it is expected to determine the search area. Furthermore, the size of the object and its manoeuverability indicate the refresh rate for the surveillance task. The surveillance area for the fast patrol boat has an increased elevation in order to detect possible weapon launches.

An air lane is also indicated in the area where airliners are expected. This expected object does not directly trigger a surveillance task due to the usual neutrality of airliners, rather it supports the classification process, see Section 2.3. Adding other, more threatening, expected air objects in the vicinity of the own ship will lead to sensor function requests for search areas.

Instead of indicating areas where certain types of objects are expected, the operator can also define e.g., a missile threat that occurs all around the ship thus triggering a horizon search task directed against sea-skimming missiles. The update rate for such a surveillance task is based on the minimum required detection range for such missiles.

4.2.2 Detected objects

For detected objects, sensor task requests are generated following the principles as explained in [104]. Tasks are generated based on the risk calculation mentioned before. More specifically, tasks are generated when the uncertainty that still exists on the risk an object poses exceeds an operator defined threshold.

Consider an object on which information about J attributes is available. The available information on the j-th attribute is given by several sensor measurements over time and is given by a mean value μ_j and a uncertainty given by σ_j with $j \in \{1, 2, \ldots, J_\ell\}$. The risk can be calculated using the DBN of Bolderheij, [9], and the risk object t poses to the mission given the current information is denoted $R(t)$. This risk calculation is done using the mean values of the attributes only. When sequentially varying each attribute with its uncertainty the system can determine which attribute uncertainty causes most variation in the risk, [104].

It is likely that the uncertainty about classification caused most uncertainty in the risk as was shown in Section 1.3.2 and the DBN shown in figure 1.3. When this is the case, the approach introduced in Section 2.5 is used to determine the information need on which the sensor task request is based. The priority assigned to the resulting sensor task request is determined by the maximum value of the risk given the uncertainties in the information about the attributes.

4.3 Sensor allocation

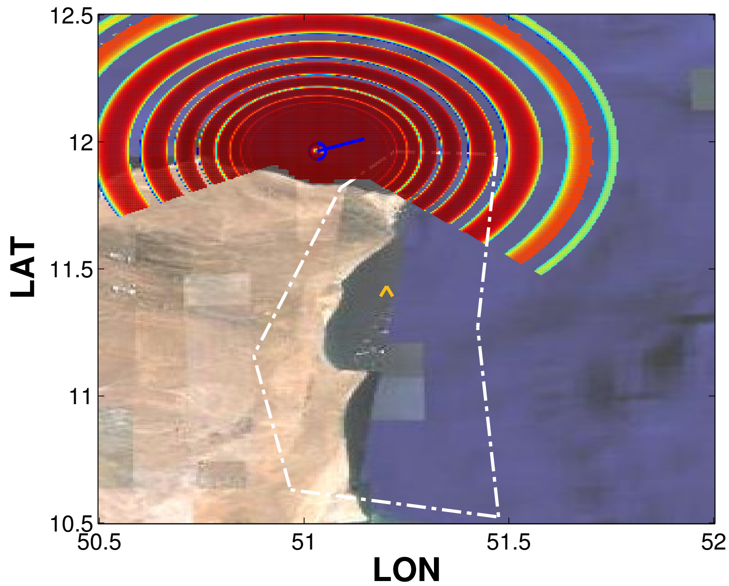
Sensor allocation is the process of assigning tasks to different sensor systems. This needs to be done in such a manner that the most important tasks are executed in time and preferably by the most suitable sensor. The choice that needs to be made is which sensor to use when. This depends on the sensor suitability and whether or not the sensor system is made available to the sensor manager by the operator.

4.3.1 Sensor suitability

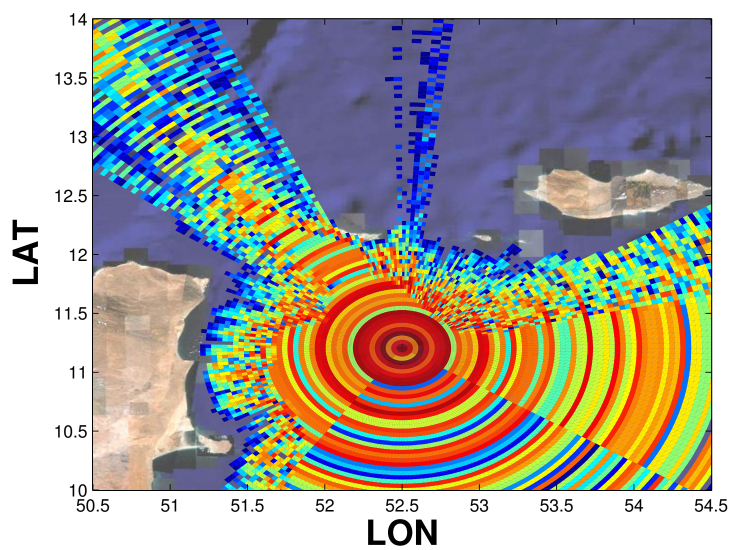
The suitability of a sensor for a given task depends on a number of things. Firstly, it depends on the type of object at which the sensor task is directed. Important factors are its position, speed and relative — or visible — size. Secondly, it depends on the sensor itself. For instance, a RADAR has different characteristics than a Sound Navigation and Ranging (SONAR) and will perform differently when directed at the same object. Finally, the suitability depends on the environmental conditions, both meteorological and geographical.

The environmental conditions play a different role for different types of sensors. For RADAR systems a program like Computer Aided RADAR Performance and Evaluation Tool (CARPET), see [40], might be used and for the propagation Advanced Refractive Effects Prediction System (AREPS), [83], could be used. For electro-optical systems a program like Electro-Optical Signal Transmission And Ranging (EOSTAR), [56], predicts the performance against different objects. For SONAR finally, an overview of propagation and performance prediction models can be found in [31]. Besides using the environmental conditions only for sensor performance prediction it can also be used in the more general purpose of mission planning, see e.g., [1].

Such performance prediction tools are combined with altimetry (land height) and bathymetry (water depth) information and are used to determine detection probabilities (P_D) which can be visualised in an overlay on the mission area as shown in figure 4.4. Figure 4.4(a) e.g., shows the detection probability of an S-band RADAR against an object at *350* m altitude with a RADAR-cross-section of *10* m^2 in the area of figure 4.3. In figure 4.4(b) the coverage of passive SONAR is shown based on the bathymetry of the environment. In figure 4.4 the colour red is used for $P_D = 1$ and blue for $P_D = 0$. Using these measures, the system can determine whether or not a sensor system fulfils the accuracy demands of the task at hand. Since the operator usually does not know the exact altitude of the expected threat, the vertical coverage is shown in a secondary screen for an input direction. Figure 4.5 shows this coverage for the example from figure 4.4(a) looking due south, in heading *180*. The altimetry information is added, while all bathymetry information is set to zero since the water depth does not directly influence RADAR performance.



(a) Coverage of an S-band RADAR against a helicopter at *350* m altitude



(b) Coverage of passive SONAR against a surface target

Figure 4.4: Embedding performance prediction tools into the CMS enables displaying the coverage diagrams directly on the tactical area

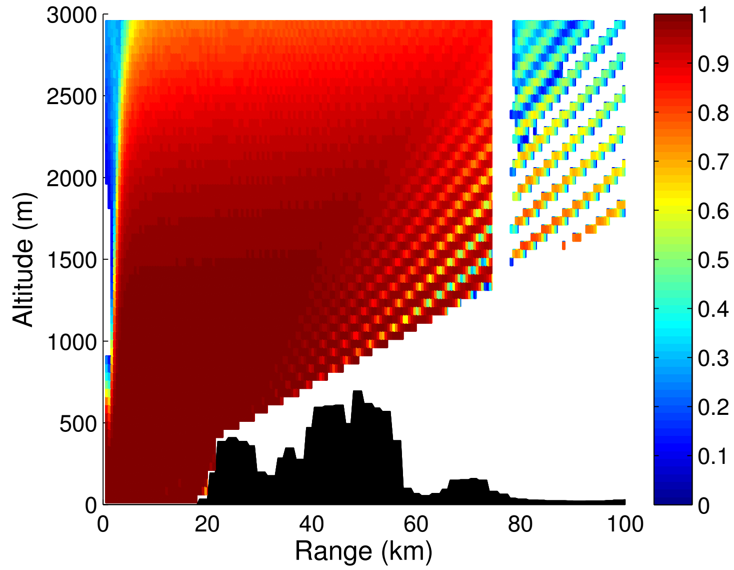


Figure 4.5: Vertical coverage of an S-band RADAR

In [58] these principles are explained in more detail for RADAR and SONAR for both the horizontal and vertical coverages.

Bathymetry and altimetry in figure 4.4 are taken from the public National Oceanic and Atmospheric Administration (NOAA) databases, [71]. This data has a resolution of 1 nautical mile. Interpolation is used to obtain data for the required range intervals in the mission areas. The RADAR performance is calculated using CARPET and SONAR performance is predicted using a range dependent loss model called Range dependent Acoustic Model (RAM)geo, [31].

Based on these prediction tools, the optimal settings — stage three of the sensor manager — for a sensor can also be determined for certain tasks given the environment. Since this operation can be done e.g., once every *15* minutes the computational complexity of the optimisation algorithm is not a problem.

4.3.2 Scheduling

The scheduling algorithm that is used for scheduling tasks has been discussed in previous studies, see [72], [75], [7], and [51]. The focus of this thesis is to show the validity of describing the information needs and determining sensor suitability. Since scheduling is necessary to complete the sensor control cycle, this section briefly discusses a scheduling algorithm which can be used in the system described in this thesis.

Margialot and Langholz introduced a scheduler based on fuzzy Lyapunov synthesis in [61]. It was designed for generic Job Shop Scheduling Problems, but has been adapted for sensor management, [72] and [75]. This fuzzy heuristic sorts all tasks in different buffers, one buffer (or queue, denoted Q_f) for each type of sensor task. Whenever a sensor becomes available, the weight for each buffer is calculated for sensor Ψ using equation (4.1) and is denoted $w_\Psi(Q_f)$. The sensor then takes the heaviest buffer and picks the task that adds most to the calculated weight.

$$w_\Psi(Q_f) = O_{\Psi,f}\sqrt{|Q_f|} \cdot \sum_{\forall t \in Q_f} S(\Psi, t) \cdot R(b) \tag{4.1}$$

In equation (4.1), $O_{\Psi,f}$ denotes the operator preference to use a certain sensor Ψ for task type f, $|Q_f|$ denotes the number of tasks in buffer Q_f, t denotes a task in a buffer and $R(t)$ denotes the risk object t, at which this task is directed, poses to our current mission. Finally, $S(\Psi, t)$ denotes the suitability of sensor Ψ for task t and can be determined using the mechanisms explained in section 4.3.1.

5

Performance issues

The greatest harm can result from the best intentions.

[...]

It sounds a paradox, but kindness and good intentions can be an insidious path to destruction. Sometimes doing what seems right is wrong, and can cause harm. The only counter to it is knowledge, wisdom, forethought, and understanding the First Rule. Even then, that is not always enough. [...] *Violation can cause anything from discomfort, to disaster, to death.*

Wizard's second rule – Stone of Tears (Ch. 63)

Most classifiers assume exclusive classes from a classification tree. Based on exclusiveness of classes, combination rules have no need for knowledge of the classification model. In chapter 2 of this thesis, a new approach to model the classification space is introduced together with classifiers that use available prior knowledge about the different classes. Chapter 3 discussed how the results from these classifiers can be fused while taking into account that class labels are not necessarily exclusive. This new classification approach needs to be compared to existing ones. Comparing two approaches requires evaluation criteria suitable for classification problems with non-exclusive classes.

Evaluation criteria for classifiers exist that either deal with soft classification results in problems with exclusive classes, see e.g., [38]; or that deal with hard classification results with non-exclusive classes, see e.g., [13], [18], and [112].

Evaluation criteria for non-exclusive labels assume a tree structure where each label has only one parent label. The new classification approach from Chapter 2 however provides soft classification results for non-exclusive labels where labels may have multiple parent labels. New evaluation criteria are presented in this chapter which can be applied in domains where soft classifiers are used for multi-label classification with non-exclusive, hierarchical labels. Chapter 6 discusses results obtained with both the existing criteria as well as the new criteria.

Required computational power for new approaches is important in implementations. This chapter therefore also discusses the computational complexity of the PCR6 rule for combining the information from the different classifiers. Steps are introduced to reduce the required computation time in order to make an implementation feasible.

The work presented in this chapter has appeared in [89] and [77].

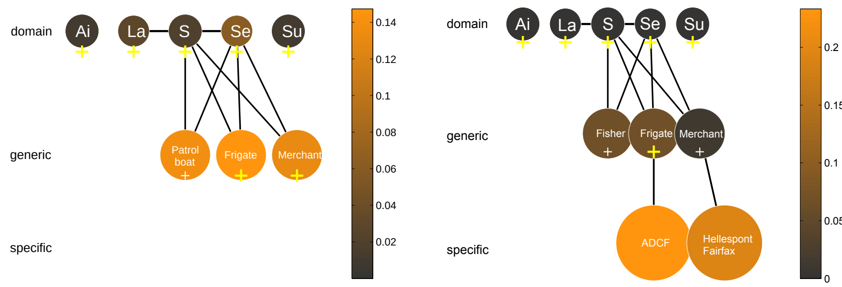
5.1 Traditional evaluation

The goal of a classifier is simple: assign a label to data. To determine how well a classifier performs, a number of performance criteria have been introduced in the literature. In this section we discuss the existing criteria and show their shortcomings.

5.1.1 Error estimation

The error-estimation criterion, denoted E, counts how many times a hard classifier wrongly classifies an example from the test data. The error estimation therefore states a percentage of how often a classifier is wrong based on test data. The criterion E is suitable for problems with exclusive classes but it is not suitable for problems with non-exclusive classes since it is unclear which labels should be counted as wrong. Furthermore, E requires hard classifier output whereas the proposed classifiers from Chapter 2 express belief on all labels (soft classification). Using E therefore, means that the classifier output is the label that was assigned most belief.

In figure 5.1 a possible classification solution from two different soft classifiers is shown for an example where the true classification is Air Defence and Command Frigate (ADCF). Although classifier *2*, figure 5.1(b), assigned much belief, namely *0.233*, to the correct solution (an ADCF), it also assigned relatively much belief (*0.193*) to a wrong label from a different branch. In contrast, classifier *1* in figure 5.1(a) has spread its belief evenly over more generic labels, but all of them ships. In that sense, classifier *1* indicates that it does not have enough information for a more specific description, whereas classifier *2* suggests to have a more definitive answer since it assigns belief to more specific classes.



(a) Solution with classifier *1* (b) Solution with classifier *2*

Figure 5.1: Possible classification solutions of different classifiers for an object with true classification *Air Defence and Command Frigate*

To compare the soft classifiers using E, the output of classifier *1* becomes *Frigate* and that of classifier *2* becomes ADCF based on the label that received most belief. This leads to the conclusion that classifier *2* has best performance since it finds the right solution. In contrast, in interviews the operators indicated that the output of classifier *1* is more desirable since it admits having uncertainty on various types of ships. This keeps a worst-case scenario open that would be (wrongfully) neglected using the output of classifier *2*. Test criteria for classifiers operating on non-exclusive classes should therefore take into account how classifiers spread their belief over classes in the entire solution space.

5.1.2 Recall and precision

In multi-label learning applications the F_1 measure as proposed by Yang, [112], is used. This criterion is based on recall and precision. Recall on object o, denoted r_o and given in equation (5.1), in multi-label applications is defined as the number of found labels that were correct divided by the total number of correct labels. The precision on object o, denoted pr_o and given in equation (5.2), is defined as the number of found labels that were correct divided by the total number of labels that were found. The F_1 measure is based on the mean recall and precision, denoted $\widehat{r}$ and $\widehat{pr}$ respectively, over all objects and is given in equation (5.3).

$$r_o = \frac{\left|X \in \mathcal{S}_o \wedge X \in \hat{\mathcal{S}}_o\right|}{\left|X \in \hat{\mathcal{S}}_o\right|} \tag{5.1}$$

$$pr_o = \frac{\left| X \in \mathcal{S}_o \wedge X \in \hat{\mathcal{S}}_o \right|}{|X \in \mathcal{S}_o|} \tag{5.2}$$

$$F_1 = \frac{2\widehat{r}\,\widehat{pr}}{\widehat{r} + \widehat{pr}} \tag{5.3}$$

In equations (5.1)–(5.3) X denotes a label, $\mathcal{S}_o$ denotes the set of labels that are correct for the o-th object, and $\hat{\mathcal{S}}_o$ denotes the set of assigned labels to the o-th object. These criteria are based on hard classifier output where the MBCs produce a normalised soft classification result. Re-writing these quantities for soft normalised classifier output means that for both recall and precision the denominator equals *1* due to the normalisation of the classifiers. In this case, $|r| = |pr|$ and thus the F_1 metric equals the precision (or the recall), $F_1 = \widehat{pr} = \widehat{r}$. Furthermore, the numerator of precision and recall, which for hard classifiers is defined as the number of correctly found labels, shows similarities with credibility and plausibility i.e., summing all masses assigned to (partly) correct labels.

Elkan proposes to use a cost function for recall and precision to weigh different classification solutions on importance during the training on labels in [28]. The DBN for risk calculation from Section 1.3.2 may be used as this weight function since it can couple the classification performance to mission success. Although this mission dependence is important, it is also a drawback of using the DBN as a weighing function in classifier learning. Since the risk calculation is mission dependent, each classifier would need training for all different specific missions in order for this to work. Another possible implementation would be to train the classifiers using the DBN during mission execution. Operationally speaking this is unacceptable since classification should also be conducted directly at the start of a mission.

5.1.3 The loss function

For multi-label learning the loss function is also used as an evaluation criterion. This function counts the number of labels that are found by a classifier that are in the right branch of the classification tree. This metric may be used in various ways. For the use of the loss function in uni-category classification tasks see e.g., [13] and [18]. Other ways of using the loss function are also possible. Dekel et al. use the length of the correct path in the branch in the loss function, [18], whereas Cai and Hofmann use a weighed version of this path to distinguish between the different levels in the tree, [13].

Cesa-Bianchi et al. [14] use the parent, child relation in the loss function so that a wrongly classified child label is not penalised if the parent was already misclassified. In [87], Rousu et al. build upon this by using a sibling and subtree scaling to re-weigh individual prediction errors on individual nodes.

For similar problems the loss function may be applied differently depending on which literature one follows. Which specific application of the loss function is better than another for different domains is unknown. Since no unified application is known of this criterion it is not used in this thesis. Furthermore, although the loss function can deal with multiple parent labels in the classification tree it still assumes hard classification results leading to similar problems as encountered with the error estimation criterion in Section 5.1.1.

5.1.4 Confusion matrix

All criteria discussed so far operate on hard classification results. The traditional criterion for soft classifiers is the confusion matrix (see e.g., [38]), which is denoted M. The element $M_{i,j}$ in this matrix denotes the mean value of belief assigned to label j when the ground truth was label i. The focus when using this criterion is on the diagonal of M, the mean values of belief assigned to the correct label. In the case of non-exclusive labels however, more areas in M need to be taken into account. Section 5.2.1 will therefore introduce new evaluation criteria which are based on those areas in M.

Criteria that take the distribution of belief into account are not only relevant for military applications. All processes where a relatively small difference in classification leads to a completely different action, such as crisis response and financial markets, would benefit from a good insight in classifier uncertainty for non-exclusive, multi-class problems.

5.2 New metrics

The previous section shows that the existing criteria in literature for multi-label learning are not suitable as evaluation criteria for the type of domain discussed in this thesis. Two new criteria are therefore introduced. Both criteria, the confusion metrics and the distance metrics, are based on the confusion matrix while taking the model of the classification space into account. These new criteria are inspired by the loss function from e.g., [18] and [14], but do not have the same drawbacks as the loss function itself.

5.2.1 Confusion metrics

All criteria for multi-label learning described in Section 5.1 use knowledge about parent, child relations and count the number of the correct labels found and distinguish based on the level in the tree. For soft classification results it therefore seems logical to look at the values in the confusion matrix and sum values based on the model knowledge. These *confusion metrics* sum the confusion between non-exclusive classes with a different level of specificity.

For example, when the ground truth solution is *Seahawk* the confusion with all types of *Helicopter* with exception of the *Seahawk* itself is summed. This may be repeated for each level of specificity obtaining multiple values. In contrast to the loss function approaches, these confusion values are not summed over all the different orders of parents. Instead, the a-th order ancestor confusion (or confusion in the branch denoted $B_a(X_i)$) on label X_i is calculated by equation (5.4).

$$B_a(X_i) = \sum_{\forall X_j} \{M_{i,j} \mid X_j \in \mathcal{H}_a(X_i)\} \tag{5.4}$$

In these confusion metrics, the ancestral labels are determined by equation (2.1) from Section 2.1 where $\theta_{k,n}$ is replaced by X_i using the mapping function Ω, equation (2.2). Furthermore, the set $\mathcal{H}_a(X)$ denotes the set that is constructed by joining all a-th order ancestor labels of label X that are not included in the other ancestral label sets, equation (5.5). The labels in set $\mathcal{H}_a(X)$ are referred to as the unique a-th order ancestral labels of X.

$$\mathcal{H}_a(X) = X^{\uparrow a} \setminus \left(\bigotimes_{p=-K}^{a-1} X^{\uparrow p} \right) \tag{5.5}$$

In equation (5.4) the confusion matrix is denoted M and single values $M_{i,j}$ represent the mean value of belief a classifier assigns to label X_j when the correct label is X_i. For overall classifier evaluation the mean value of the a-th order branch confusion over all labels is examined, $\widehat{B_a}$. For hard classifiers with exclusive classes the mean value of the diagonal of the confusion matrix is often used, [38], this value is produced by $\widehat{B_0}$.

5.2.2 Distance metrics

Equation (5.4) indicates the total amount of confusion with the a-th order parents and all its children. Figure 5.1 however shows, that the total amount in those areas does not give enough information. When a classifier is wrong, it would be preferable that its confusion is evenly (or uniformly) spread over the child elements in the right branch. Our second criteria type, the distance metric, therefore determines the Root-Mean-Square (RMS) distance of confusion values in the branch to the mean value in that area. This is also done for each specificity level.

$$\delta_a(X_i) = \sqrt{\frac{1}{|\mathcal{H}_a(X_i)|} \cdot \sum_{X_j \in \mathcal{H}_a(X_i)} \left(M_{i,j} - \frac{B_a(X_i)}{|\mathcal{H}_a(X_i)|} \right)^2} \tag{5.6}$$

The RMS distances for the a-th order ancestor confusion is denoted δ_a and it is given by equation (5.6) for each label X_i. The number of unique a-th order ancestor labels of label X_i are needed to calculate the mean value and it is denoted by $|\mathcal{H}_a(X_i)|$. To determine overall classifier performance the mean value over the class labels is determined, $\widehat{\delta_a}$.

5.3 Computational complexity

The size of the hyper-power set used in DSmT follows the Dedekind numbers given the size of the frame of discernment, [92] and [102]. In [102] a proof is given that the n-th Dedekind number may be calculated by equation (5.7).

$$\prod_{k=1}^{2^{2^n}} \sum_{j=1}^{2^n-1} \sum_{i=0}^{j-1} \left(1 - b_i^k b_j^k \prod_{m=0}^{\log_2(i)} 1 - b_m^i + b_m^i b_m^j \right) \tag{5.7}$$

In equation (5.7),the term b_i^k is determined by equation (5.8). Sufficient to say, the number of required computations increases dramatically when the frame of discernment grows. For practical implementations of DSmT and PCR6 this is a problem that needs to be addressed.

$$b_i^k = \frac{k}{2^i} - \frac{2k}{2^{i+1}} \tag{5.8}$$

Recent developments in DSmT show a shift from using the hyper-power set to using the so-called super-power set, [24]. The difference between the two is that the hyper-power set is constructed using the $\cap$ and $\cup$ operators, the super-power set is constructed using the $\cap$, $\cup$, complement, and exclusion operators. The size of the super-power set that needs consideration will therefore increase much faster than the Dedekind numbers given the size of the frame of discernment. Although this new super-power set is not used in this thesis, it does becomes apparent that reducing the required computation time becomes even more important should this work be expanded with the super-power set.

5.3.1 Filtering

Although the required number of computations in the case of classification is reduced as a result of the model constraints, this reduction is not enough for practical implementation purposes. In this section, another approach is introduced to reduce the computational complexity for practical implementations.

The MBCs assign belief to all classes from the frame of discernment, Section 2.4. This implies that a lot of elements from this frame receive a non-zero value. By filtering the output of all classifiers before combining them, the number of non-zero values is increased, which in turn reduces the number of possible combinations of elements that need to be evaluated. For a threshold value λ, the filtered output of source ℓ can be determined by equation (5.9). In order to maintain validity, the resulting masses are normalised.

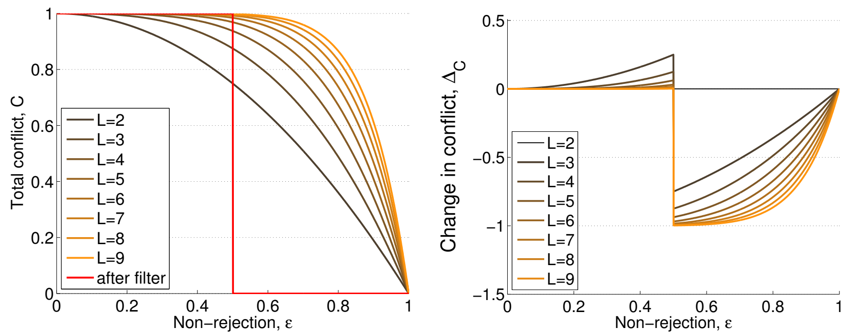
$$m_\ell^{\text{filt}}(X) = \begin{cases} m_\ell(X) & \text{if } m_\ell(X) \geq \lambda \cdot (\max_{X_i} \; m_\ell(X_i)) \\ \\ 0 & \text{if } m_\ell(X) < \lambda \cdot (\max_{X_i} \; m_\ell(X_i)) \end{cases} \tag{5.9}$$

5.3.2 Zadeh's example

To get an idea of the impact of the proposed filtering on the output, it's effect on Zadeh's example, [114], is studied. Zadeh's example addresses the case where all sources disagree on what is the correct label. All L sources however, do not fully reject an additional label X_{L+1} in Zadeh's example. This additional label is therefore assigned a small belief, ε, of non-rejection. Zadeh's example usually assumes small values for this non-rejection belief, $\varepsilon \leq 0.1$. Since our goal is to study the effects of filtering, we consider $0 \leq \varepsilon \leq 1$. Table 5.1 shows Zadeh's example for L-sources where belief expressed by a classifier (or source) on label X is denoted by $m_\ell(X)$ with $\ell \in \{1, 2, \ldots, L\}$.

Table 5.1: Zadeh's example for L-sources

	X_1	X_2	$\ldots$	X_L	X_{L+1}
$m_1(.)$	$1-\varepsilon_1$	0	$\ldots$	0	ε_1
$m_2(.)$	0	$1-\varepsilon_2$	$\ldots$	0	ε_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$m_L(.)$	0	0	$\ldots$	$1-\varepsilon_L$	ε_L



(a) Conflicting mass in Zadeh's example (b) Added conflict due to filtering

Figure 5.2: The influence of filtering on the conflict for Zadeh's example when using DSmT

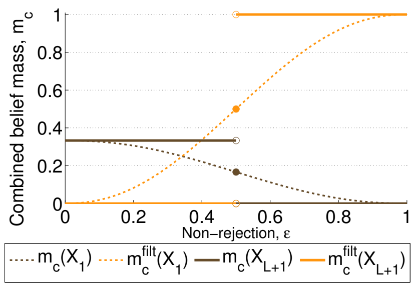
Effect on conflict

The amount of conflict, denoted $\mathcal{C}$, in Zadeh's example is given by equation (5.10) since all labels are mutually exclusive. For the effect of filtering we look at a simplified case where all ε_ℓ are equal in which case we use ε for notational ease. In this simplified example, the product in equation (5.10) is reduced to ε^L.

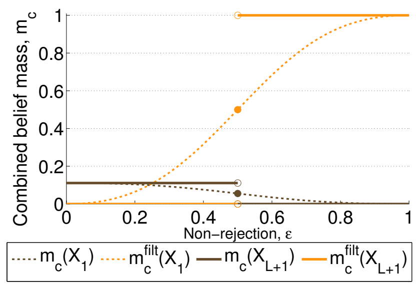
Figure 5.2(a) shows $\mathcal{C}$ for different values of ε and L using equation (5.10). The proposed filtering from equation (5.9) entails that the conflict becomes *1* when $\varepsilon \leq 0.5$ becomes *0* otherwise, see table 5.1. This means that the filtering either increases conflict (when $\varepsilon \leq 0.5$) or it decreases conflict (when $\varepsilon > 0.5$). Applying a filtering threshold λ satisfying equation (5.11) produces figure 5.2(b).

$$\mathcal{C} = 1 - \prod_{\ell=1}^{L} \varepsilon_\ell \tag{5.10}$$

$$\lambda \geq \begin{cases} \frac{\varepsilon}{1-\varepsilon} & \text{for } \varepsilon \leq 0.5 \\ \\ \frac{1-\varepsilon}{\varepsilon} & \text{for } \varepsilon > 0.5 \end{cases} \tag{5.11}$$



(a) Belief mass for $L = 3$



(b) Belief mass for $L = 9$

Figure 5.3: From the belief masses for $L = 3$ and $L = 9$ can be seen that with more sources combined belief assigned label X_{L+1} exceeds that of other labels for smaller values of ε

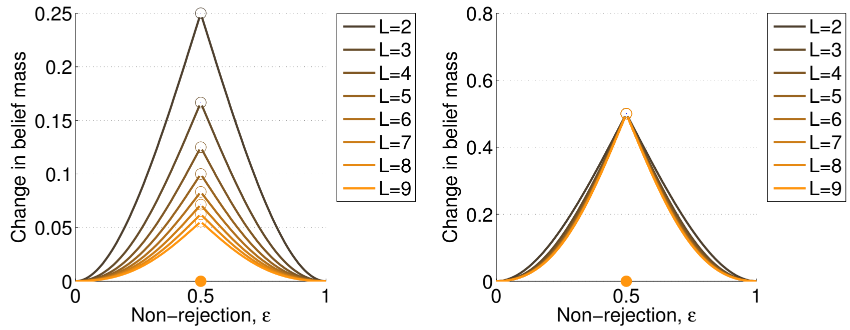
Effect on belief mass

Figure 5.2(b) shows that the impact of the input filtering on the conflict decreases as the size of the frame of discernment and the number of sources increases for $\varepsilon \leq 0.5$. The impact on the belief masses themselves (and on subsequent decisions based on those beliefs) is something different. The combined belief on label X_1 and on X_{L+1} for the unfiltered and the filtered case are shown for $L = 3$ and $L = 9$ in figures 5.3(a) and 5.3(b) respectively.

Figures 5.3(a) and 5.3(b) show that filtering changes the combined belief to such an extent that it alters the decision significantly. This effect is stronger when more sources are combined. More sources influence the decision so strong due to Zadeh's problem definition. In figure 5.4(a) the absolute difference between $m_c(X_1)$ and $m_c^{\text{filt}}(X_1)$ is shown. Figure 5.4(b) shows the absolute difference for $m_c(X_{L+1})$. Note that for $i \in \{2, 3, \ldots, L\}$, $m_c(X_1) = m_c(X_i)$ holds and that all combined belief sums to one.

Figures 5.3(a), 5.3(b), and 5.4(b) show that the unfiltered combined belief for label X_{L+1} hardly changes when a different amount of sources are combined. The values for the other labels however do vary, figures 5.3(a), 5.3(b), and 5.4(a). For all X_i with $i \in \{1, 2, \ldots, L\}$ the belief masses are the same and depend on L. The label X_{L+1} accumulates all parts of the conflict it was involved in according to PCR6. When more sources assign a small ε to label X_{L+1}, this label therefore receives more combined belief whereas the other labels need to divide the conflict amongst each other.

$$\frac{1}{L+1} = \sum_{i=0}^{L-1} \left[\binom{L-1}{i} \cdot \frac{(1-\varepsilon)^{L+1-i} \cdot \varepsilon^i}{(L-1)(1-\varepsilon) + i\varepsilon} \right] \tag{5.12}$$



(a) Changes for $m_c(X_1)$ (b) Changes for $m_c(X_{L+1})$

Figure 5.4: Belief mass in Zadeh's example for $L = \{2, 3, \ldots, 9\}$ where can be seen that combined belief assigned to label X_{L+1} is less sensitive to the number of sources compared to the other labels

Filtering negates this accumulation effect. The decision to choose for X_{L+1} therefore is only made when $\varepsilon > 0.5$. The point where all combined belief masses are equal for the labels occurs at $\varepsilon = 0.5$ in the filtered case. When no filtering is applied, this value of ε where all combined belief masses are equal depends on the value of L.

This dependency of ε on L in the unfiltered case is given by equation (5.12) for $L \geq 2$ and $0 \leq \varepsilon \leq 1$ when PCR6 is applied. When voting is used as a combination mechanism, all combined belief masses are equal for $\varepsilon = \frac{1}{L+1}$. Figure 5.5 shows a numerical estimation (using Newton-Raphson, [109]) of equation (5.12) for $L \in \{2, 3, \ldots, 30\}$. For comparison reasons, figure 5.5 also shows when all combined belief masses are equal for voting and the filtered case.

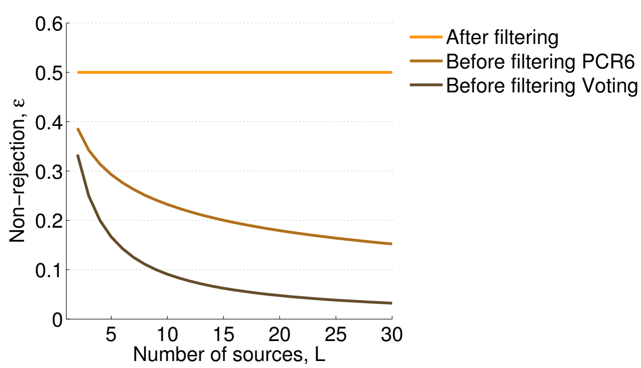


Figure 5.5: Value of ε against L where $m_c(X_1) = m_c(X_{L+1})$ in Zadeh's example

In figure 5.5 the lines indicate the value of ε where all combined belief masses are equal for three different cases. The area above a line indicates that label X_{L+1} receives more combined belief mass than the individual labels X_i with $i \in \{1, 2, \ldots, L\}$ for that particular case. The effect of filtering on combined belief masses is less when PCR6 is used to combine the sources.

5.3.3 Applicability of filtering

Decisions based on combined belief after filtering can differ from the unfiltered case in Zadeh's example when $\varepsilon \leq 0.5$. Before using the proposed filter, it is therefore vital to check if beliefs that are set to *0* in the various sources are not all assigned to the same labels. Should this be the case, the accumulation effect is negated by the filter which causes a different decision.

Negation of the accumulation effect is not unexpected for Zadeh's example since it is usually described for $\varepsilon \leq 0.1$ and in those regions the effects of filtering are negligible. When applying the filter to other examples it is important to first check if the negation effect does not influence the decision.

The change in decision caused by filtering occurs strongly in Zadeh's example. It is expected that this effect is less for situations with non-exclusive labels. Belief masses may be unfiltered that have a partial overlap with labels of which belief has been filtered out. This way, a combined belief is still assigned to that label when applying combination rules that take the non-exclusiveness of labels into account like PCR6 thus overcoming the negation effect.

6 Evaluation

People can be made to believe any lie, either because they want to believe it's true, or because they are afraid it's true.

Wizard's first rule – Wizards first rule (Ch.36)

HAVING an idea is one thing, coming to a fully operational system is quite another. The last important phase to come to such a system is evaluation. Various tests have been done based on the same classification solution space in different areas using traditional as well as the new evaluation criteria. For these tests, training and test data are used for traditional classification approaches. The test data has also been used to evaluate the new MBCs for which no training is required.

The first section of this chapter discusses the training- and test data as well as different scenarios that have been tested. Section 6.2 discusses the evaluation of various classifiers as well as their computational complexity. The user interaction required for the military domain is discussed in Section 6.3. Section 6.4 shows how sensor task requests are constructed in a simulated environment. Section 6.5 closes with the application of the new MBCs in a theoretical problem to show the general applicability of the classifiers proposed in this thesis.

6.1 Scenario

In order to validate the concepts introduced in the previous chapters of this thesis, different scenarios have been implemented for testing. Each of these scenarios has been constructed using the same database of classes to generate data. This data has also been used as a knowledge source for the MBC. The scenarios are either placed at a geographical location or in a simulated environment depending on the type of test.

6.1.1 Classes

Using the set notation from Section 2.1, a database consisting of *14* specific classes and *13* more generic classes as presented in table 6.1 is used. Additionally, at the lowest level of specificity ($k = 1$) the three domains are represented: *air*, *surface*, and *subsurface*. The *surface* domain has two child elements at a next specificity level, namely *sea* and *land*. Generic objects may be a child element to one of these five low specificity objects or it may be a bridge between two or more of them.

The objects implemented have a variation in military and civilian platforms as well as in weapons. Platforms may be related to each other to bring weapons and weapon carriers together. A generic *speedboat* may be a recreational vehicle but it could also be used to fire a Rocket Propelled Grenade (RPG). A generic *speedboat* has no weapon range, but if the *speedboat* is related to an RPG based on the expected threat in a region it does have a weapon range.

Using these relations many more sub-classes can be created. Different frigates can be based on the same platform but carry different weaponry. A Fast Patrol Boat (FPB) may be fitted with only a gun but it might also carry short range missiles. Relating more generic classes in this manner can expand the dataset more. These extensions are relevant to threat- and risk evaluation and not directly for classification. The extensions are therefore not explored in more detail in this thesis.

Section 3.2.1 introduced the model $\mathcal{M}$ as a *2*-tuple containing the frame of discernment Θ and the set $\varnothing_{\mathcal{M}}$ containing all model constraints. For the frame presented in table 6.1 the model constraints need to be described. Section 2.1 stated that all frames using the set notation contain mutually exclusive labels at the same specificity level. All combinations of labels in the hyper-power set (D^{Θ}) that include such intersections are therefore included in $\varnothing_{\mathcal{M}}$.

More labels are mutually exclusive than only those on the same level of specificity: an object cannot belong to both the *Air* class and the *Submarine* class. The ancestral relations between the labels in table 6.1 are used to determine which combinations of labels in D^{Θ} are valid and which should be included in $\varnothing_{\mathcal{M}}$. Table 6.2 shows the parent labels at the different ancestral levels for the elements at the most specific level, $k = 4$, for the classes shown in table 6.1.

Table 6.1: Classes at Various Levels

n	**Specific** ($k = 4$)	**Generic** ($k = 3$)	**Sub-Domain** ($k = 2$)	**Domain** ($k = 1$)
1	Boeing 747	Helicopter	Land	Air
2	Air Defence and Command Frigate (ADCF)	Airliner	Sea	Surface
3	F-16 Fighting Falcon	Fighter jet		Subsurface
4	Hellespont Fairfax	Missile site		
5	Seahawk	Fast patrol boat (FPB)		
6	Walrus class submarine	Fisher		
7	Jumbojet	Submarine		
8	Exocet	Frigate		
9	Harpoon	Merchant		
10	Leopard II tank	Missile		
11	Apache	Tank		
12	F-14 Tomcat	Speedboat		
13	Multi-purpose Frigate (MFF)	Rocket propelled grenade (RPG)		
14	Kilo class			

Table 6.2: Ancestral relations for the specific classes, $k = 4$

$\theta_{4,n}$	$\theta_{4,n}^{\uparrow 1}$	$\theta_{4,n}^{\uparrow 2}$	$\theta_{4,n}^{\uparrow 3}$
Boeing 747	Airliner		Air
Jumbojet	Airliner		Air
F-16	Fighter		Air
F-14	Fighter		Air
Exocet	Missile		Air
Harpoon	Missile		Air
Seahawk	Helicopter		Air, Surface
Apache	Helicopter		Air, Surface
Leopard II	Tank	Land	Surface
ADCF	Frigate	Sea	Surface
MFF	Frigate	Sea	Surface
Hellespont Fairfax	Merchant	Sea	Surface
Walrus class	Submarine	Sea	Surface, Subsurface
Kilo class	Submarine	Sea	Surface, Subsurface

Table 6.3: Ancestral relations for generic classes, $k = 3$

$\theta_{3,n}$	$\theta_{3,n}^{\uparrow 1}$	$\theta_{3,n}^{\uparrow 2}$
RPG		Surface
Missile site	Land	Surface
FPB	Sea	Surface
Fisher	Sea	Surface
Speedboat	Sea	Surface

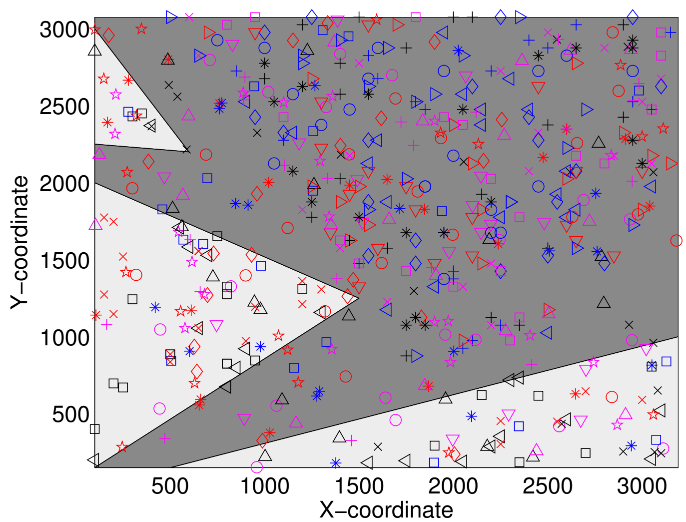


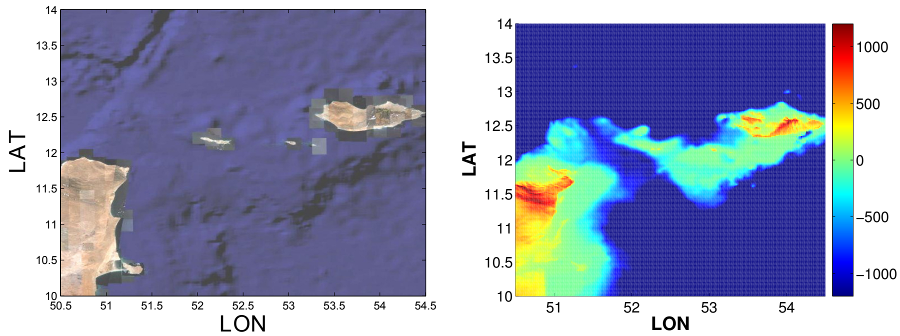
Figure 6.1: Train data plotted on the chart of the environment based on Cartesian coordinates

Based on these relations, other valid combinations can be deduced and those that cannot be deduced are given in table 6.3.

A frame of discernment is obtained with $I = 32$ and it is therefore a multi-class problem. Membership functions that describe possible behaviour for generic objects and normal behaviour for specific objects have been implemented for all classes in the frame of discernment. Each class membership is described on the following attributes:

- Position;
- Speed and altitude combination, where depth is seen as a negative altitude value; and
- Size.

Based on these functions, train- and test data has been generated. Since the membership functions were chosen to realistically model the specific classes, the resulting dataset is considered to be realistic enough for the purpose of this evaluation. Each object in the train- and test data contains *9* attributes: X-position, X-position uncertainty, Y-position, Y-position uncertainty, altitude, altitude uncertainty, speed, speed uncertainty, and size.



(a) Chart for the area

(b) Bathymetric and altimetric chart for the area

Figure 6.2: Area for the sensor management simulation

6.1.2 Data

Train- and test data for classification have been generated based on the membership functions that describe possible and normal behaviour. For each specific and generic class in the database *60* objects are randomly generated, leading to a total of *1620* objects. From this data *33%* is used for training and the rest for testing. Figure 6.1 shows the train data with their Cartesian position in the area. In this figure, the dark grey is sea and the lighter grey is land and each coloured shape represents a different class.

6.1.3 Area

Charts of the geographical areas are used for constructing various scenarios. For sensor management and classification purposes, the bathymetry (water depth) and altimetry (land height) information about the region are important. One of the areas is shown in figure 6.2. The satellite image of the region is shown in figure 6.2(a) and the bathymetry and altimetry information is visualised in figure 6.2(b). For the creation of the scenarios at geographical locations, coordinates are set in latitude and longitude, for simulated environments Cartesian X-Y coordinates are used.

Bathymetry and altimetry information is taken from the public NOAA databases, [71]. This data has a resolution of *1* nautical mile. Linear interpolation is used to obtain data for the required range intervals in the mission areas. The focus of the evaluation in this thesis is to validate the principle of how such information may be used in a system. The resolution is therefore considered to be sufficient. Higher resolution information can be obtained from e.g., NASA's Shuttle RADAR Topography Missions at a *90* meter resolution, [43].

6.2 Classification results

Using the train- and test data explained in the previous section, all classification set-ups have been evaluated with respect to both traditional criteria and the new criteria from Chapter *5*. The different classification set-ups that have been tested, are discussed after which the results are given and discussed.

6.2.1 Classification set-ups

To compare the different classifier evaluation criteria, three types of classifiers are compared based on the data explained in Section 6.1.2. The three types of classifiers are:

1. trained classifiers,
2. Model-Based (MB) trained classifiers, and
3. Model-Based Classifiers (MBCs).

Multiple classifiers are trained for each classifier type — except for the MBC since they do not require training — and combined using either PCR6 or a voting algorithm, leading to six classification system set-ups that are evaluated. Implementations of the trained classifiers are taken from the *Pattern Recognition (*PR*) Toolbox*[1]. More information on these different classifiers can be found in e.g., [38]. The PR *Toolbox* also provides the implementations of the training routines and the traditional evaluation functions.

The PCR6 combination rule from DSmT is explained in Section 3.2.2 and in more detail in Appendix A. The voting algorithm takes the mean of all beliefs that were expressed to a label by the different classifiers.

Trained classifiers

Classifiers that are trained on (a subset of) attributes are referred to as trained classifiers. Here, three different kind of trained classifiers — explained in e.g., [38] — are used:

- a 3-Nearest Neighbour (NN) classifier – when plotted in the attribute space, the class that has three training examples closest to the object under consideration is the classification solution;
- a Linear Distance Classifier (LDC) – the class with lowest linear distance to the central point of the training examples of that class is the classification solution; and

[1] This toolbox is developed by the Pattern Recognition Group of Delft University of Technology and available via www.prtools.org

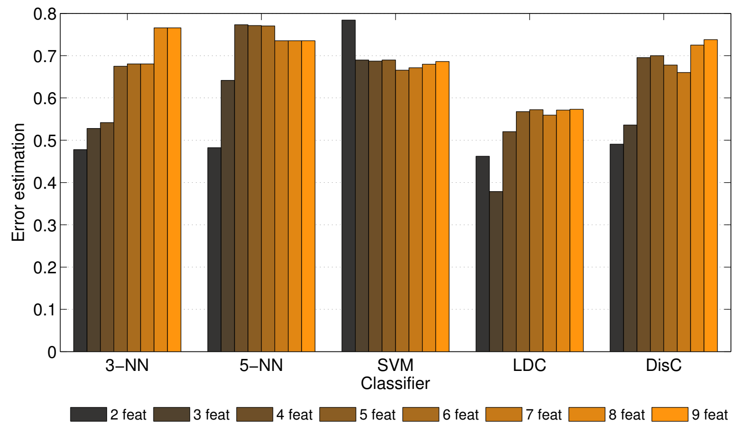


Figure 6.3: Error estimates of different classifiers trained on different amounts of attributes of the train data

- a Dissimilarity Classifier (DisC) – defines a kernel to maximise dissimilarities and then trains a linear classifier on this new attribute space.

The classifiers were chosen based on initial test results which are shown in figure 6.3. These three classifiers were preferred to a Support Vector Machine (SVM) and a 5-NN classifier since they outperform them. When classifiers are trained on a selected number of attributes, a feature evaluation is executed to determine which attributes are used. Based on these initial results, the 3-NN and the DisC, are trained on two features and the LDCs are trained on three features.

All of the chosen classifiers can be used to directly give a single label (hard classification) but can also be used as soft classifier. The latter option is chosen in this thesis because the MBCs are soft classifiers as well. Furthermore, the combination rules are designed to combine soft information.

MB-trained classifiers

Based on three specificity levels, namely domain ($k = [1, 2]$), generic classes ($k = 3$), and specific classes ($k = 4$), classifiers have been trained. These classifiers are referred to as MB-trained classifiers since the training data is split over different classifiers based on the model. For these classifiers we use LDCs based on their performance during initial tests. By themselves, these classifiers are not expected to perform well since they are only trained on subsets of the class labels. When combined however, they complement each other. This should lead to a classification set-up that outperforms the trained classifiers since each individual classifier has its own speciality whereas the trained classifiers are all trained on all labels.

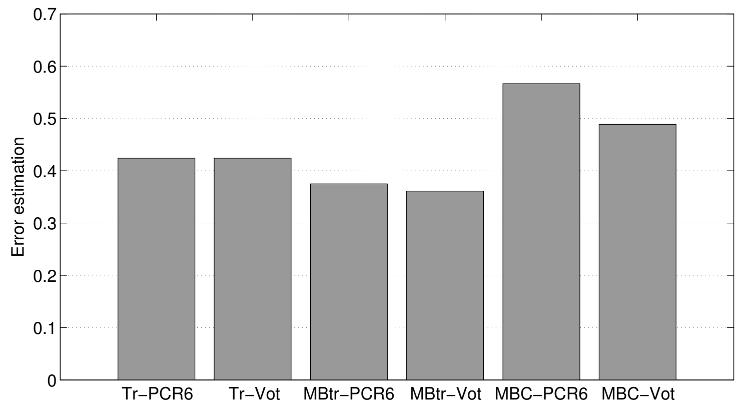


Figure 6.4: Error estimations of different combination rules and different classifiers

Model-Based classification

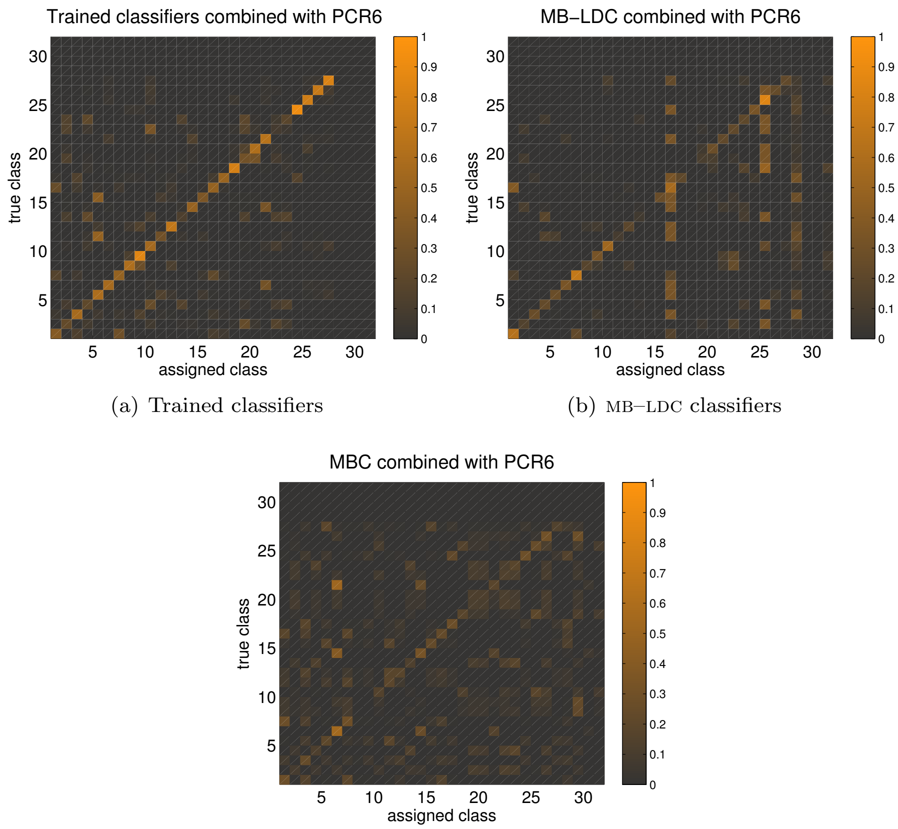
Using the membership functions belonging to each class, MBCs have been implemented. The MBC approach can be used since the test data contains uncertainty information on various attributes. Based on the similarity between the different classes it is expected that the MBC approach is especially suited to spread belief over different classes rather than choose a single classification solution with certainty.

6.2.2 Error estimation

The error estimate, E, is determined by counting how many test objects are not classified correctly based on the hard classification output. Note, only an exactly correct classification solution counts as a successful classification. The error estimations are shown in figure 6.4 for the different classifier set-ups. All different set-ups show a high value for E. This is not unexpected since the class labels are very specific and the error estimation criterion does not take less specific answers into account. Based on E, the conclusion is that the MB-trained classifiers combined with the simple voting algorithm give the best result.

6.2.3 Confusion

Traditionally, the confusion matrix is examined when comparing classifiers in more detail. The values $M_{i,j}$ in a confusion matrix M, are obtained by calculating the mean belief a classifier assigns to label X_j when the correct label is X_i. Figure 6.5 shows the confusion matrices for the three types of classifiers when combined with PCR6. In figures 6.5(a) and 6.5(b) the downside of only



(a) Trained classifiers (b) MB–LDC classifiers (c) MBC

Figure 6.5: Confusion matrices for various classifiers combined with PCR6

considering the mean value of the diagonal of M becomes apparent. This metric might be high, because some classes are classified with a high precision whereas others are never classified correctly. In figure 6.5(c) the overall mean on the diagonal is low, but roughly the same for all classes, which might be desirable when robustness is desired in multi-class problems.

For the MB-trained classifiers an additional downside is visible in the confusion matrix. Distinct vertical lines show up in the visualisation of the confusion matrix. This means that despite the information, the classifiers have a certain bias for a small amount of classes.

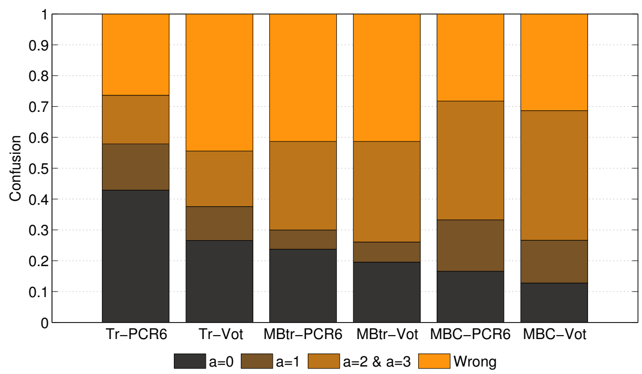


Figure 6.6: Confusion distribution for different combinations of classifiers

New confusion metrics

In contrast to looking at the entire confusion matrix or the diagonal of M, Section 5.2.1 introduced new evaluation metrics based on confusion denoted $\widehat{B_a}$. These metrics sum the confusion values at the different ancestral orders a. The mean value of the diagonal of M is obtained for $a = 0$, since each class label is its own zero-th order parent. Summing all ancestral confusion for all values of a a value smaller than or equal to *1* is found, $\sum_{\forall a} \widehat{B_a} \leq 1$. The difference between the summed ancestral confusion and *1* is the mean amount of belief that has been assigned to completely wrong labels. This could be seen as the soft classification evaluation criterion version of the error estimation criterion which is more suitable for hard classification.

The results of the mean ancestral confusion, $\widehat{B_a}$, are shown in figure 6.6, where the exact results are obtained for $a = 0$, the branch results are for $a = 1$, and the domain indicates $\widehat{B_2} + \widehat{B_3}$. The amount of wrongly assigned belief is obtained by $1 - \sum_{a=0}^{3} \widehat{B_a}$. This figure shows some interesting results when looking at the mean amount of wrongly labelled data from this confusion-matrix approach. Where in figure 6.4 the highest error-rate was for the MBC combined with PCR6, the highest mean value in the confusion matrix on wrong labels is assigned by trained classifiers combined with the voting algorithm. Second worst on this criterion are the MB-trained classifiers. Remarkable, since these performed best based on the error-estimation criterion. This already shows that conclusions on classifier performance change when different criteria are used.

In figure 6.6 the advantage of PCR6 over the simple voting strategy from Section 6.2.1 is also visible. That this effect occurs most in the trained classifiers is expected. The trained classifiers do not take the interrelations between classes into account whereas the MBC and the MB-trained classifiers do. The knowledge of the solution space that PCR6 uses therefore has most effect on the trained classifiers. In general, the more knowledge of the fusion model is used by the classifier, the less difference between the voting algorithm and PCR6 occurs.

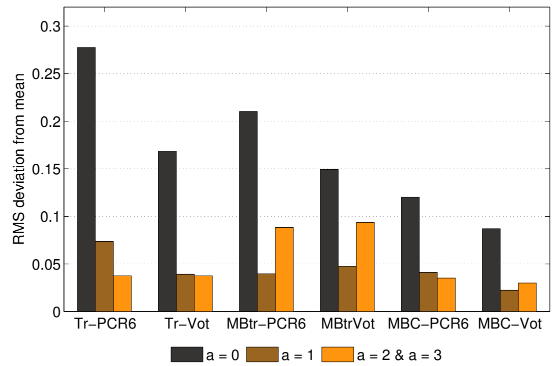


Figure 6.7: The RMS deviations for different combinations of classifiers

New distance metrics

Section 5.2.2 introduces new distance metrics that express a degree of robustness a classifier has in assigning belief at the various ancestral labels, which are denoted $\widehat{\delta_a}$. This metric describes how robust the classifier classifies all the exactly correct labels for $a = 0$. When this value is equal to zero it means that all classes are assigned the same amount of belief namely $\widehat{B_0}$. Similar reasoning follows for different values of a. For high values of $\widehat{\delta_a}$ it means that some classes are found and some are not.

In figure 6.7, the results are shown for the RMS deviations for the classifiers, where *solution* indicates $a = 0$, *branch* indicates $a = 1$, and *domain* indicates the sum for $a = 2$ and $a = 3$. These results support the conclusion that the MB-trained classifiers do not give the best results. It also supports the conclusion that the MBCs yield the best results looking at how belief is spread over less specific classes. Furthermore, they show a smaller RMS deviation on the diagonal of the confusion matrix. This means that the classification set-up has a stable performance on all classes. In contrast, the trained classifiers are good at classifying a certain number of classes, while being bad at classifying others.

Combining the results from figure 6.6 and 6.7 leads to the conclusion that the MBC combined with PCR6 has best performance for this domain where multiple non-exclusive labels have similar attribute values. This fits well with the expectations since trained classifiers are trained on dissimilarities, which would be there if the labels had been mutually exclusiveness. This is not the case in the train data for this type of domain. The MB approaches however do not search for similarities but use pre-knowledge about classes: an approach that utilises the non-exclusiveness of classes.

6.2.4 Reducing computation time

Applying filtering techniques on the classifier output prior to applying the combination rules influences the required computation time. When using the filter given by equation (5.9), the reduction is shown in figure 6.8(a). In this figure the reduction in computation time is given for the trained classifiers as well as the MBCs when combining them using PCR6.

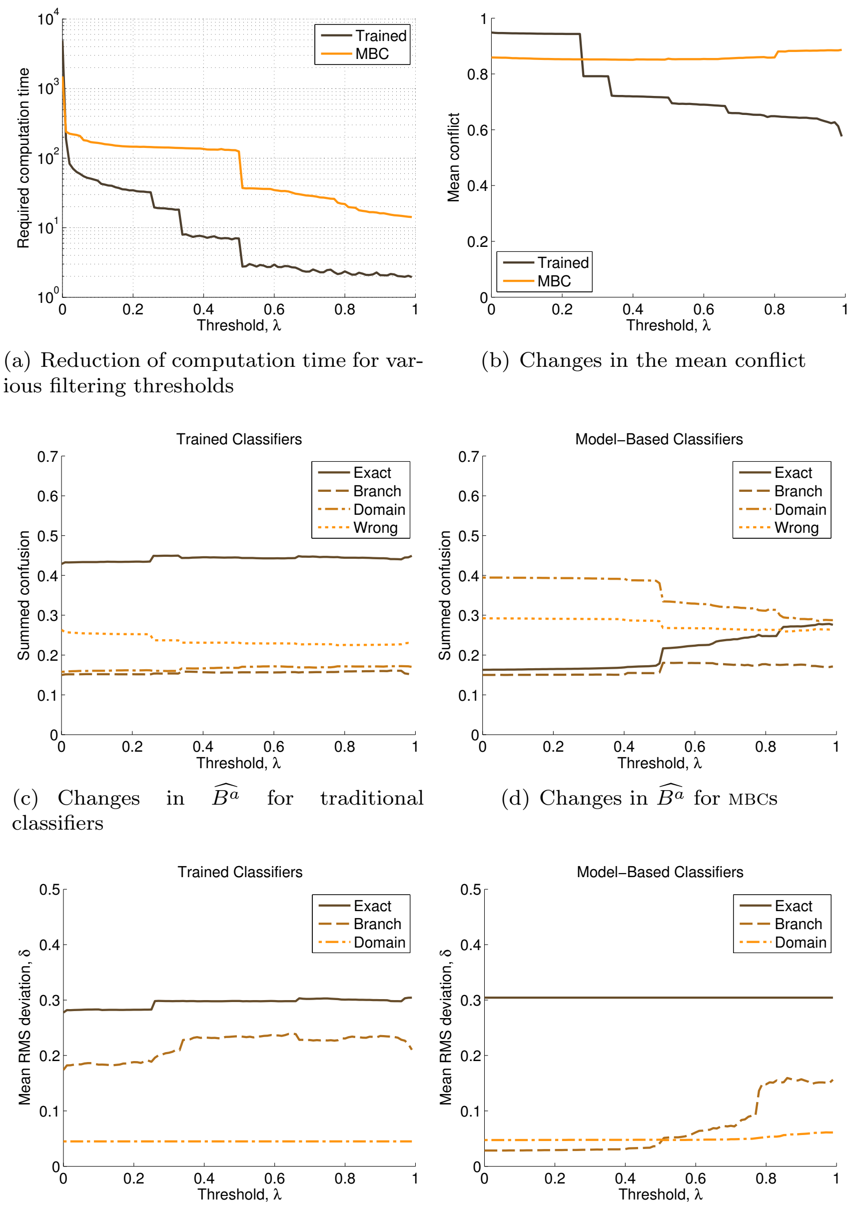
Due to filtering, the results change. The question is, how much filtering may be applied without influencing the results too much. Figure 6.8(a) shows that most gain computation-wise, is achieved for a threshold $0 < \lambda < 0.1$. Another point where computation time is reduced appears at λ=*0.5*. It is interesting to see how classifier performance is changed for the trained classifiers and the MBCs. Their performance is compared at various thresholds on the confusion and the distance metrics as described in Section 5.2.1. These performances are shown in the lower four graphs in figure 6.8.

Figures 6.8(c) and 6.8(e) show that the performance of the trained classifiers does not change much due to filtering while a significant reduction in computation time is achieved. For the MBCs this is also the case for $\lambda \leq 0.5$. When λ is increased further, the performance of the combined MBCs changes for the better.

The change in the mean conflict on the objects is visualised for various λ in figure 6.8(b) to investigate the influence of filtering further. For the conflict there is no significant change at λ=*0.5* for either of the classification set-ups. It is visible that the MBC seems unaffected by filtering, whereas the trained classifiers show a decrease in conflict. Since Section 3.4 discussed the added value of conflict, this might not be as desirable as one should think, agreement on the wrong solution might cause problems.

Based on these results the conclusion is that overall the performance is not changed too much when applying filtering before applying PCR6 to combine classifier output. Especially for low threshold values, $\lambda \leq$*0.2* the changes are small while the processing time is reduced by a factor *100* making it more feasible for implementation in (near) real-time systems.

The filter threshold λ=*0.5* seems important looking at figure 6.8(a). The sudden decrease in required computation time at this point however is due to the models of the classes and not due to the combination rule. The used classes have similar membership fields, causing the classifiers to assign the same amount of belief to multiple classes. For $\lambda \geq 0.5$, multiple beliefs are set to *0* leading to the sudden reduction in computation time since less combinations of labels are considered in PCR6. For problems where classes have less similar membership fields, this sudden reduction around λ=*0.5* will not occur.



(a) Reduction of computation time for various filtering thresholds

(b) Changes in the mean conflict

(c) Changes in $\widehat{B^a}$ for traditional classifiers

(d) Changes in $\widehat{B^a}$ for MBCs

(e) Changes in $\widehat{\delta_{B^a}}$ for trained classifiers

(f) Changes in $\widehat{\delta_{B^a}}$ for MBCs

Figure 6.8: Changes in performance when filtering is applied

6.2.5 Discussion

Different criteria lead to different conclusions on which classifiers perform best. It is important to consider all evaluation criteria and consider the different aspects of performance in relation to the problem domain. Table 6.4 presents the results for the six classification set-ups on all evaluation criteria. The confusion matrix is omitted from this table due to spacing and the fact that the mean diagonal of this matrix is given by $\widehat{B_0}$. The entire confusion matrices in the case of PCR6 are shown in figure 6.5.

Based on the results from Section 6.2.4 the computational considerations are also omitted from table 6.4. All classification set-ups are feasible in (near) real-time and this factor is therefore no longer used to distinguish between system set-ups at this stage.

The error estimation criterion (E) shows that the MB trained classifiers are the best choice if the goal is only to find the correct label. These classifiers however still have a high error-rate, $E \geq 35\%$, which means that at least *35* objects out of a hundred will most likely be misclassified. Furthermore, the confusion metrics for these classifiers show that much belief is assigned to completely wrong labels. This indicates that these types of classifiers are either completely right or completely wrong.

Although the MB trained classifiers show good results for being exactly right, they are not very robust. A high value for $\widehat{\delta_0}$ indicates that these classifiers are very good in classifying some classes but cannot correctly classify other classes. This conclusion is supported by the entire confusion matrix in figure 6.5(b) where vertical lines are visible indicating a bias for certain class labels. This is also supported by the high confusion metric on wrong labels. Results for the MB trained classifiers show that all different criteria need to be investigated before coming to a conclusion on system performance.

The results in table 6.4 also show the added value of using the knowledge of the solution space in the combination rule. For all three types of classifiers, the PCR6 rule of combination outperforms the voting algorithm based on the new

Table 6.4: Different results for all classifier set-ups

Criteria	Trained		MB-trained		MBC	
	PCR6	Voting	PCR6	Voting	PCR6	Voting
E	*42.4%*	*42.4%*	*37.5%*	*36.1%*	*56.7%*	*48.9%*
$\widehat{B_0}$	*0.4289*	*0.2658*	*0.2372*	*0.1956*	*0.1660*	*0.1276*
$\widehat{B_1}$	*0.1499*	*0.1100*	*0.0625*	*0.0651*	*0.1664*	*0.1388*
$\widehat{B_2} + \widehat{B_3}$	*0.1578*	*0.1800*	*0.2870*	*0.3260*	*0.3853*	*0.4203*
wrong	*0.2634*	*0.4442*	*0.4133*	*0.4133*	*0.2824*	*0.3133*
$\widehat{\delta_0}$	*0.2775*	*0.1685*	*0.2101*	*0.1493*	*0.1202*	*0.0871*
$\widehat{\delta_1}$	*0.0737*	*0.0391*	*0.0397*	*0.0472*	*0.0411*	*0.0223*
$\widehat{\delta_2} + \widehat{\delta_3}$	*0.0375*	*0.0375*	*0.0884*	*0.0936*	*0.0352*	*0.0299*

confusion metrics with a loss of stability in the distance metrics. This effect occurs most strongly for the trained classifiers, logical since these do not use model knowledge themselves.

Combining the different evaluation criteria leads to the conclusion that the MBCs combined with PCR6 are suitable for large classification spaces with non-exclusive classes. These classifiers show a robust performance. The fact that the voting combination rule is more stable than PCR6 is compensated by the better performance in confusion metrics. In terms of the confusion metrics, the MBCs with PCR6 perform second best when looking at assigning belief to wrong labels — best performance is achieved by trained classifiers with PCR6. This result is caused by the highest values for $\widehat{\delta_2} + \widehat{\delta_3}$ of all tested classifier set-ups. In this case, the preference is to rather be somewhat right than to be completely wrong thus overall best performance is achieved by MBCs combined with the PCR6 rule of combination. In future work it might be interesting to see how a mix of the different types of classifiers would perform.

6.3 User interaction

Different interfaces are needed to enable the required interaction with the operator. Firstly, an interface where the operator can plan the mission and secondly, an interface to enable the operator to work together with the system to minimise the uncertainty on the classification solution. This section discusses both these interfaces.

6.3.1 Mission planning

Figure 6.9 shows a screenshot of the mission planner that has been implemented for testing. In this mission planner, the tactical area is shown based on a sea chart which can be overlaid with varying information. The operator e.g., can select to see both the altimetry (land height) and bathymetry (water depth) information in the tactical information. On the right side of the interface, numeric information of a selected object is displayed. For expected objects, the operator is enabled to use this numeric display for editing.

Objects can be inserted using mouse clicks which fits user expectations. For each object, the operator can select what information needs to be displayed in the screen, i.e., position uncertainty information and/or (worst-case scenario) weapon range. Furthermore, the expected objects can also be assigned to areas that can be drawn freely or they can be assigned to relative bearings and ranges. For each of the objects the operator can view RADAR and/or SONAR performance using CARPET and RAM-geo (Section 4.3.1, [40], and [31]) respectively that are integrated in the mission planner.

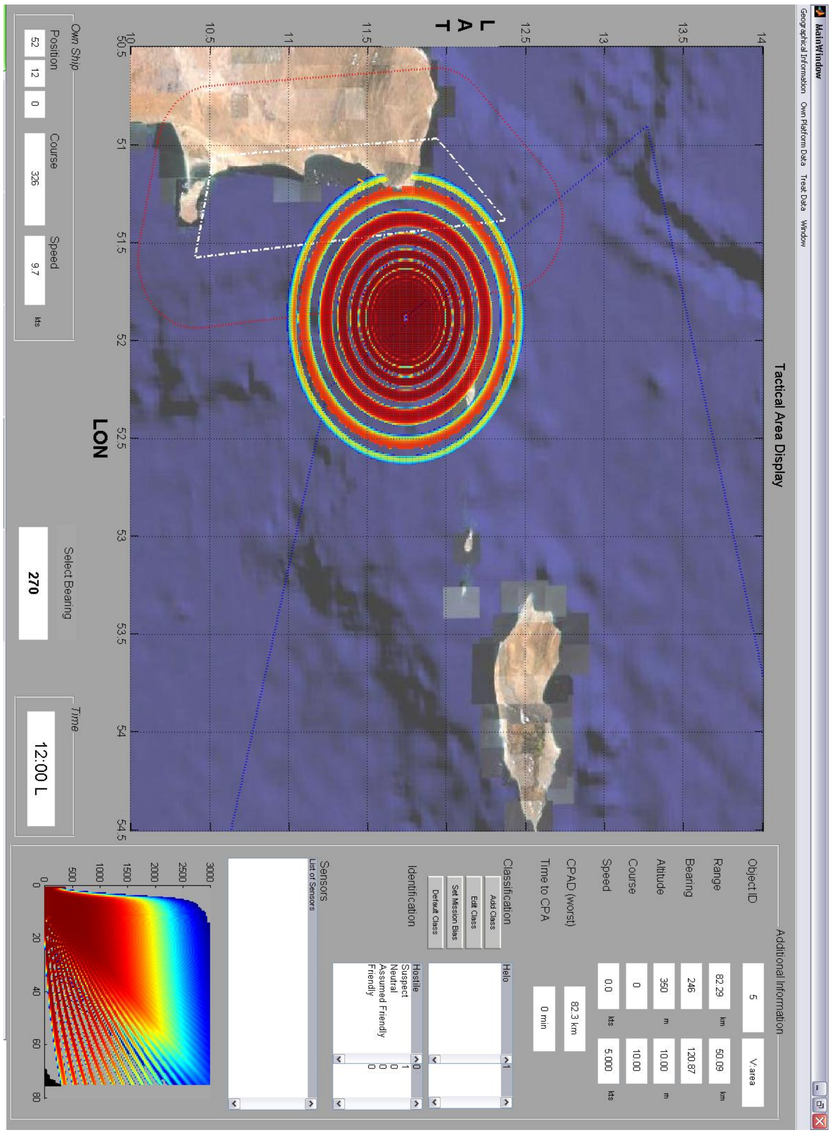


Figure 6.9: Screenshot of the implemented mission planner

Initial interviews with operators indicate that this is the type of information used in mission planning. Furthermore, the operators indicated that the overlay structure of expected sensor performance is desirable. Especially the combination of vertical and horizontal sensor coverage while taking the altimetry and bathymetry into account is seen as a major improvement over the current systems. Although this mission planner does not fulfil all desires yet in terms of visualisation, the operators state that the conceptual design is a good one.

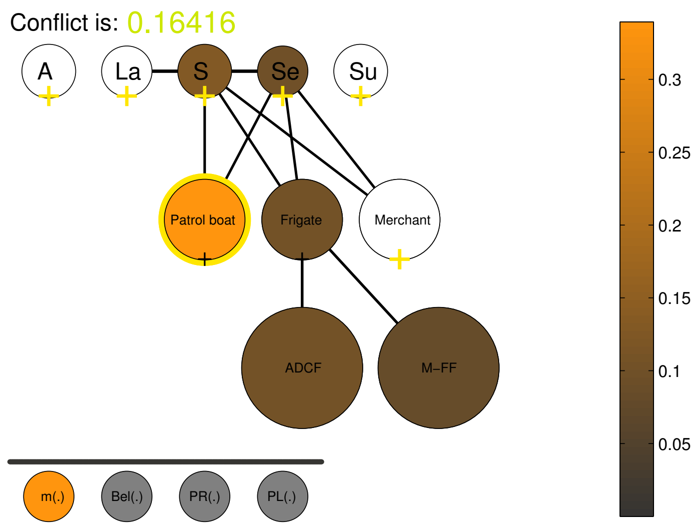


Figure 6.10: Example of the Graphical User Interface for classification, labels that received negative confirmation are not coloured and labels with positive confirmation are have a yellow highlight

6.3.2 Classification interaction

An interface is made that works within the simulation environment to communicate the combined belief to an operator and that can be used for interaction. This interface supports two functions, firstly to communicate current system belief and secondly, to enable the operator to classify manually by assigning a classification solution or by making the solution space smaller (positive and negative confirmation). In case of positive confirmation, the operators opinion is combined with the other classifiers using the combination rule. For negative confirmation UPR is used. The sum of belief that is redistributed due to negative confirmation is displayed, both numerical and in colour (zero to one is mapped on green to red). The operator is visually triggered when the negative confirmation introduces a significant amount of conflict with current system belief. A screenshot of the classification interface is shown in figure 6.10.

Chapter 3 discussed various metrics to assign belief, credibility, probability, or plausibility to class labels. All of these metrics can be visualised in the classification interface from figure 6.10 using the buttons $m(.)$, $Bel(.)$, $PR(.)$, and $PL(.)$ respectively. The question is, how can they be used in an actual system. For the example shown in figure 5.1(a) — where the ground truth was the ADCF — the various belief metrics are shown in figure 6.11 for the MBC combined with PCR6.

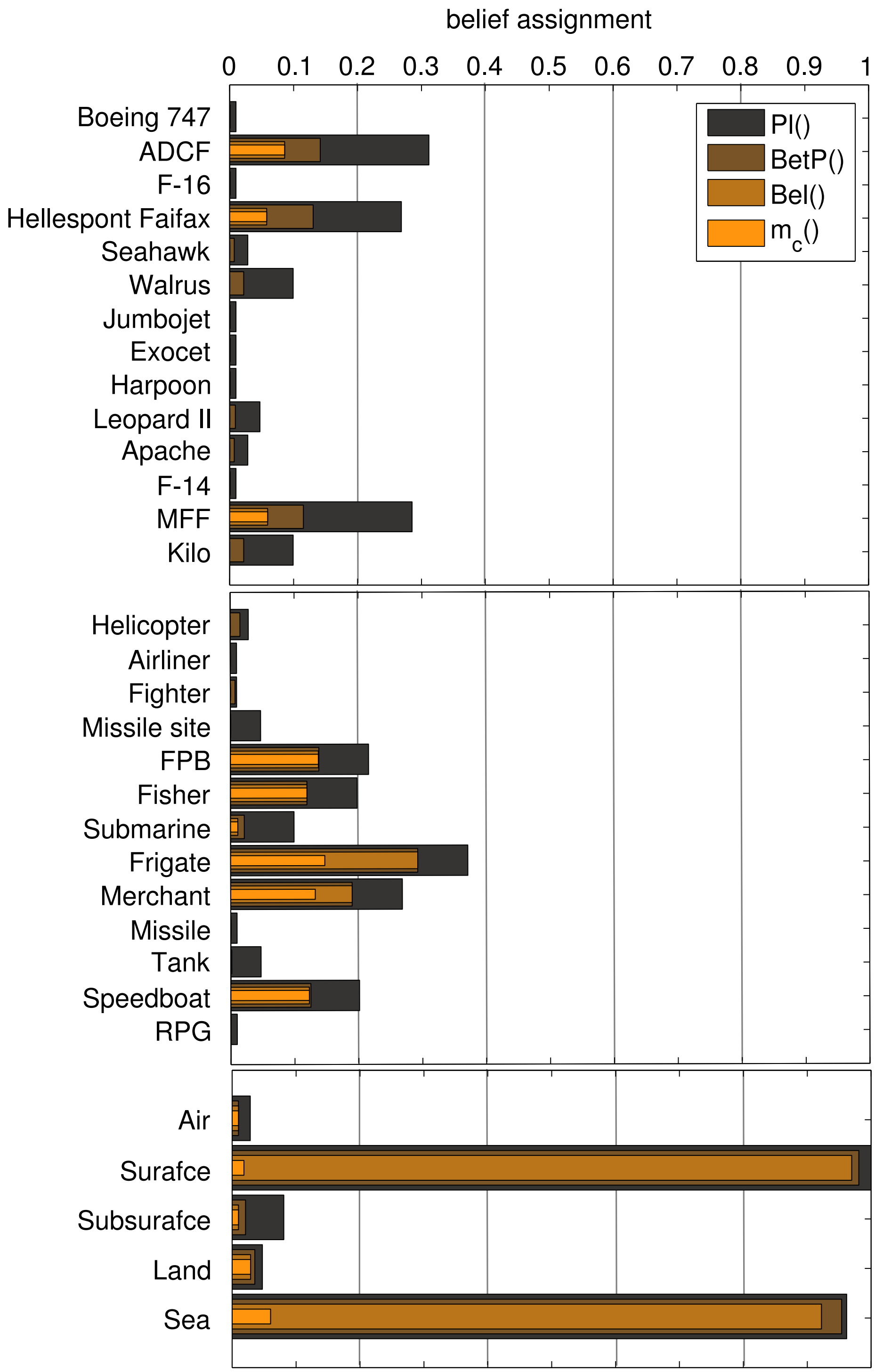


Figure 6.11: Various belief metrics for the classification solution of object *45* of the test dataset with ground truth in ADCF

The credibility that the system holds in a certain hypothesis is calculated by summing up all the evidence that fully supports it, whereas the plausibility sums up all belief that does not fully disagree with it. The difference between these two values gives a measure of uncertainty residing in a particular hypothesis. In figure 6.11 this means that there is not much uncertainty about the object belonging to the surface class.

For the surface domain $\mathrm{Pl}(Surface) = 1$ holds in figure 6.11. This means that none of the classifiers assign zero belief to that domain. There are however classifiers that still assigns belief to a label that is mutually exclusive to the object being sea-based, $\mathrm{Pl}(Sea) \leq 1$.

From the different metrics shown in figure 6.11 can be concluded that the object under consideration is almost certainly a surface object. Furthermore, it is credible that the object is sea-based but not all information sources fully agree on that conclusion.

Using the different belief metrics to calculate the uncertainty of the classification solution therefore means that there is not a single value to quantify this uncertainty. Rather, it is a number of uncertainties on different hypotheses combined with a quantity that sums the total amount of conflicting evidence, $1 - \mathrm{Pl}(X_i)$.

Interviews with operators do not provide enough information yet to draw conclusions on which metric is suitable for visualisation in an actual implementation. All options require a certain amount of explanation about what it shows. Although most indications lean towards using credibility or probability, the amount of conflicting evidence produced by $1-\mathrm{Pl}(X_i)$ might be required for ruling out worst-case scenarios. More extensive tests with operators in different scenarios are required to make more definitive conclusions.

6.4 Generating sensor task requests

The process of task request generation can be split into two processes as mentioned in Section 4.2. The first type of task generation is the construction of surveillance tasks and the second type is the construction of tasks that request additional information for already detected objects.

Table 6.5: Information based on feedback on different time steps

	Alt	*+/-*	*Speed*	*+/-*	*X*	*+/-*	*Y*	*+/-*	Size	Feedback
t_0	*8*	*10*	*8*	*6*	*1500*	*2000*	*2000*	*2000*	Medium	Altitude
t_1	*3*	*5*	*8*	*6*	*1500*	*2000*	*2000*	*2000*	Medium	Altitude
t_2	*0*	*2*	*8*	*6*	*1500*	*2000*	*2000*	*2000*	Medium	X
t_3	*0*	*20*	*8*	*6*	*1500*	*1000*	*2000*	*2000*	Medium	Y
t_4	*0*	*20*	*8*	*6*	*1500*	*1000*	*2000*	*1000*	Medium	Y
t_5	*0*	*2*	*8*	*6*	*1500*	*1000*	*2000*	*300*	Medium	X
t_6	*0*	*2*	*8*	*6*	*1500*	*500*	*2000*	*300*	Medium	*else*

Constructing the surveillance capabilities is based on the expected threats, Section 4.2. The general region of the object is determined based on mission information inserted by the operator and a safety bracket is added to it to determine the search volume. The size of such a safety bracket is also mission based, namely based on the possible speed of the expected object and its weaponry. The required update rate is determined by the risk calculation process as mentioned in [104]. This principle of requesting surveillance capabilities has been shown to work in [7]. Here, the same approach is adopted. This thesis therefore focusses on generating the task requests for additional information on already detected objects.

We consider an object — ground truth ADCF — with very uncertain information in the environment of figure 6.1. This object is given by the following available information on attributes at time t_0:

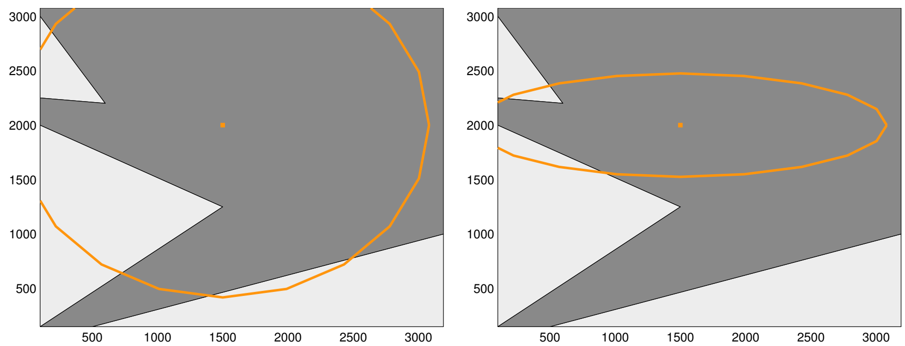
- X-coordinate: *1500 +/- 2000*;
- Y-coordinate: *2000 +/- 2000*;
- Altitude: *8*m +/- *10*m;
- Speed: *8*m/s +/- *6*m/s;
- Size: Medium.

The amount of uncertainty is rather large and the classification solution provided by the MBCs is therefore distributed over all classes with a preference for helicopters based on the altitude and the low speed. The exact solution at t_0 for the three types of classifiers combined with PCR6 is shown in Appendix B. Based on the conflict the requirement is set to reduce uncertainty in the altitude in order to try and exclude all the helicopters. From the knowledge available on classes we know that measuring altitude is indeed the only way of doing this. For different time-steps the information and the resulting feedback is given in table 6.5, the resulting classification results at t_1, t_3 and at t_6 are also provided in Appendix B.

When conflict on the helicopters is sufficiently reduced, the feedback mechanism focusses on the difference between sea- and land-based classes. Figure 6.12 shows the object's position and the uncertainty about the position in the simulated environment. At t_4 the decision is made to first reduce uncertainty in the Y-coordinate since land masses occur most in this direction, figure 6.12(a). After this reduction, the uncertainty in the X-coordinate needs to be reduced according to the feedback mechanism, a decision supported by figure 6.12(b).

After t_5, most differences are resolved resulting in only sea-based classes as a classification result. The mechanism now checks the differences between the remaining classes and decides that the current information is insufficient to decide on a more specific solution. It therefore decides *else*, indicating that another — currently unavailable — attribute is needed to decide on more specific classification solutions.

This last decision is important. Where other classifiers imply to have a good answer, the MBCs admit to still have a lot of confusion and ask for ad-



(a) t_4 (b) t_5

Figure 6.12: Position of the object at times t_4 and t_5

ditional information. After t_3, both the trained classifiers as well as the MB-trained classifiers do not change their classification solution significantly, see Appendix B. Both types have a specific label amongst their top *4* solutions, namely ADCF and *Hellespont Fairfax* respectively. Although the first is the correct solution, the information itself does not justify ignoring other possible labels. From an operational viewpoint, the results from the MBCs are preferred since they cannot exclude other labels based on the available information.

6.5 Other applications

It is interesting to see how the MBCs will perform on the more commonly described problems with mutually exclusive labels. A 2D *8* class problem is considered that is constructed using the PRtools from the Delft University of Technology. This classification problem is constructed using:

- two classes from Highleyman's dataset,
- two classes from the spherical dataset,
- two classes from the banana dataset, and
- two classes from the Lithuanian dataset.

More details about these datasets with mutually exclusive labels can be found in the manual of PRtools, [82]. From this combined set, a dataset containing *3200* objects is generated for testing of which a subset is shown in figure 6.13. This figure shows that most confusion will most likely occur between classes *1* and *2*, *3* and *4*, *5* and *6*, and between classes *7* and *8*.

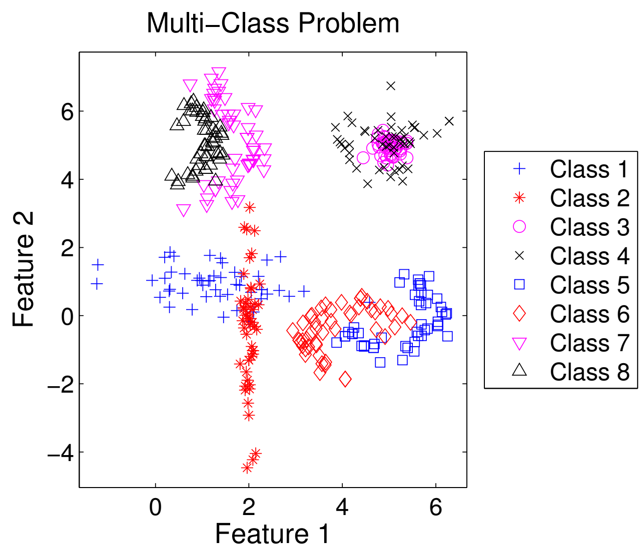


Figure 6.13: Subset of the test data for an *8* class problem

Although the classes are mutually exclusive that does not mean they cannot look similar in the attribute space: a whiskey glass is different from a wine glass but on the attribute *material*, they have the exact same value.

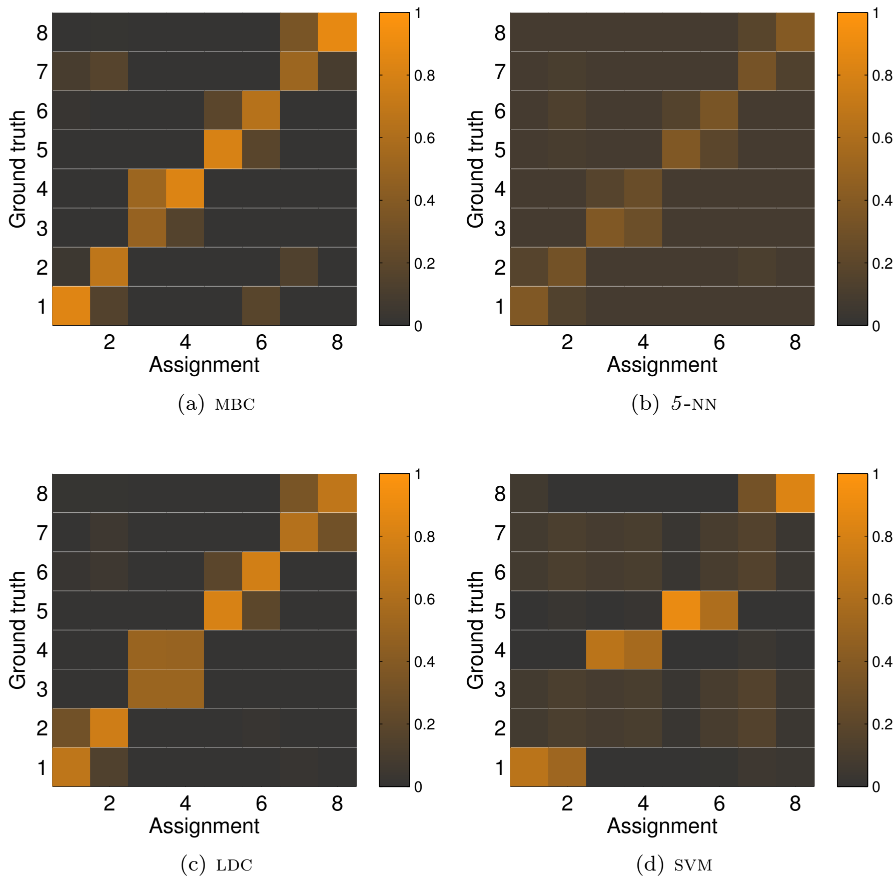
The soft classification versions of

- a *5*-NN classifier,
- a LDC, and
- a SVM

are trained to compare performance with the MBCs. The training of these three types of trained classifiers is done on a dataset with *20* objects of each class that are generated separately from the dataset used for testing.

The results are compared based on an examination of the confusion matrix shown in figure 6.14. The new metrics as introduced in this thesis are not used in this case since the dataset only contains mutually exclusive labels. The new metrics are specifically developed for a close examination for problems with non-exclusive labels and are therefore not relevant in this specific case.

Figure 6.14(c) shows that the LDC has trouble distinguishing between the classes as expected based on figure 6.13. As a matter of fact, all classifiers show similar confusion although the SVM, figure 6.14(d), and the *5*-NN classifier, figure 6.14(b), perform worse than the other two classifiers.



(a) MBC (b) 5-NN (c) LDC (d) SVM

Figure 6.14: Confusion matrices for different classifiers for the *8* class problem

The MBC, figure 6.14(a), has a better performance when looking at the diagonal of M compared to the LDC. The latter however has less trouble with confusion between classes *1* and *2* and class *7*. It is difficult to exactly say which classifier is the better. What can be concluded however, is that the MBCs are suitable for application in a broader set of classification problems than for those for which they have been developed given that they have a proper class model whereas traditional classifiers require a proper training set.

7

Conclusions

The rule of all rules. The rule unwritten. The rule unspoken since the dawn of history... But Barracus wanted you to know that it's the secret to using a war wizard's power. The only way to express it, to make sure that you would grasp what he was intending to tell you, was to give you a book unwritten to signify the rule unwritten.

The Rule Unwritten – Confessor (Ch. 58)

Due to technological and political developments the need for more automation and decision support in Combat Management Systems (CMS) is increasing, especially in the fields of sensor management and classification. Previous work has shown that sensor management and classification depend on each other. The prioritisation mechanism for sensor tasks is highly dependent on the classification solution and the uncertainty therein while classification needs sensor information. How to support the classification process and how to reduce uncertainty efficiently through sensor management is therefore of great interest.

The main research question addressed in this thesis therefore is:

How can operators be supported in their task of interpreting real-time data in complex environments?

Due to maritime military application domain considered in this thesis, more specific research questions are addressed in this thesis. Combined, the answers to those specific questions provide a solution to the main research question for the application domain of this thesis.

How should the class labels be modelled when the operators use different classification trees and require more specific or less specific answers?

A generic set notation for classification is introduced in this thesis to deal with non-exclusive and hierarchical labels. Labels may describe very generic classes like the class of all air object, or more specific classes like the Dutch Air Defence and Command Frigate.

Between the different hierarchical levels in the model ancestral relations indicate is a class at a higher level overlaps with a class on a lower level. This provides a flexible model that may be expended by inserting new hierarchical levels and indicating how they relate to the labels one level higher and one level lower.

How should classifiers cope with uncertain input from sensors and intelligence reports?

This thesis proposes to fit sensor measurements to membership fields that are defined in a multi-attribute space for each class. These membership field are based on available prior information on possible and normal behaviour. Model-Based Classifiers (MBC) are introduced that map confidence intervals of measurements on the membership fields to fit the available information and the expected or possible manifestations. Multiple classifiers are constructed since classes may be described by various membership fields.

An added value of using confidence intervals is the possibility to do the reverse as well: one can determine which uncertainty reduction will lead to a better classification solution. Sensor task requests can then be sent to the sensor manager based on the desired information.

What conditions need to be met to combine classifiers that operate on uncertain input and that assign belief to labels on different hierarchical levels?

Based on the membership fields for the various labels and the available information, different classifiers assign belief to the labels. These classifiers may provide conflicting information. The combination scheme used to fuse the different classifiers should therefore be able to deal with conflicting beliefs and it should to deal with the non-exclusiveness of the labels used in classification.

The various MBCs are combined using Dezert-Smarandache Theory since that theory can deal with conflicting, uncertain and paradoxical information. Using DSmT has the advantage that plausibility, credibility and probability are defined, providing several quantities that can be visualised to interact with the operator. In particular, the PCR6 from DSmT is used because it can deal with multiple, highly conflicting information sources.

An additional rule, the User Preference Redistribution rule (UPR), is introduced in this thesis to enable the operator to exclude parts of the solution space. Using UPR the system redistributes the belief accordingly while keeping track of the amount of conflict the exclusion introduces. When this conflict exceeds a threshold, the operator can be informed regarding this anomaly.

Combining the MBCs also enables anomaly detection by using these classifiers on membership fields based on mission information and normal behaviour for certain classes in the area. This means that anomaly detection can be accomplished without extensive machine learning techniques.

How should classifiers be evaluated taking the hierarchical levels of the class labels into account and that generic but correct answers are preferred over specific answers that may be wrong?

Existing classifier evaluation criteria are based on either classifiers that operate on exclusive classes or on hard classifiers that operate on non-exclusive classes. In this thesis however, soft classifiers operate on non-exclusive hierarchical labels. Furthermore, wrong decisions can have severe consequences in our application domain. New criteria are proposed to evaluate classifier performance in domains with these characteristics.

Two new types of evaluation criteria are proposed, both based on the confusion matrix. The first type is based on summing the confusion values that belong to the different levels of specificity. The second type of criteria examines the Root-Mean-Square (RMS) distances from the mean at the different specificity levels. Based on these criteria, this thesis shows that the MBCs are well suited for classification problems with a large frame of discernment.

Extensions may be made to further increase performance by examining combinations of different classifiers, e.g., MBCs for specific classes and trained classifiers for more generic classes.

How can classification uncertainty be described and how should the classification process determine which information is needed to reduce that uncertainty?

Classification uncertainty is a difficult term to describe in a single quantity. This thesis therefore proposes to look at the plausibility and the different between the plausibility and the credibility. Both of these quantities are defined in the DSmT framework that is used in this thesis. Another quantity that might be used is the pignistic probability. This value calculates the probability based on the belief values. More research and tests with operators are required to determine which of these options is most suited to express the uncertainty in a single class label and the classification process.

Reducing the uncertainty in classification can however be achieved using a different approach. When combining the results from the classifiers the conflict between them is also an indication of uncertainty. Looking at the most generic classes on which classifiers express a different belief and trying to resolve that conflict first will result in a better classification result. Based on the membership field and the model-based classification approach sensor function requests can be generated to reduce the conflict.

How should sensor management get the required information in a complex environment?

This thesis adopts the three-stage sensor management concept from literature in which sensor management is divided in three separate steps: generate the required sensor tasks, allocate the most appropriate sensor for each task, and control the sensor to execute the task as best as possible.

The first stage is achieved based on the mission and the classification uncertainty that lead to sensor task requests for surveillance and additional information respectively. By integrating various propagation and sensor models into a Combat Management System the suitability of a sensor for a specific task can be determined which is needed for sensor allocation. At the last stage of sensor management, all requested tasks are scheduled with a fuzzy Lyapunov scheduling algorithm.

This thesis shows that the information needs for accurate and complete picture compilation (all situation and threat assessment processes combined) can be determined in an automated fashion while allowing operator influence throughout the process. Based on the required information, different sensor functions can be requested from the sensor manager. The emerging management scheme consists of the following steps:

1. insert expected objects and the available prior information (membership fields) about those objects into the system;
2. construct Confidence Intervals based on measurements for each object;
3. map these Confidence Intervals on membership fields that are based on the prior information;
4. combine the results using PCR6;
5. enforce additional operator preferences using UPR;
6. use the conflict from PCR6 to calculate which information is required and generate sensor function requests;
7. allocate the most appropriate sensor for each task using the performance prediction tools available for the different sensors;
8. schedule the sensor systems using the fuzzy Lyapunov scheduler;

9. use the conflict from UPR to communicate with the operator about unexpected occurrences.

In short, incoming data is used in different reasoning processes that are supported by prior information such as e.g., expected behaviour. This thesis provides a methodology with which those reasoning processes can identify which additional information they need to come to less uncertain conclusions. These requests for information can be used to obtain additional data to reduce this uncertainty.

The operator is vital as an information source during the planning and execution phases of missions and several suggestions have been made for the required interfacing functionalities. Integrating available tools for sensor performance prediction into the overall system will enable the system to automate the sensor allocation. It can also be used to communicate operational consequences of command decision in an understandable fashion during the planning and execution phases of missions.

References

[1] Farid Askari, Giampaolo Cimino, Erik Malaret, Hank Pomeranz, and Rich Allard. Web based decision aids for rapid environmental assessment: METOC impact matrix summary (MIMS) and WIPE tactical axon (WTA). Technical Report SR-357, SACLANT Undersea Research Centre, September 2002.

[2] Yaakov Bar-Shalom, Thiagalingam Kirubarajan, and Cenk Gokberk. Tracking with classification-aided multiframe data association. *IEEE Transactions on Aerospace and Electronic Systems*, 41:868–878, July 2005.

[3] Abder Benaskeur. Sensor management in command and control. In *Proceedings of the 7th International Command and Control Research and Technology Symposium*, Quèbec, Canada, 16–20 September 2002.

[4] E. Bertucat and H. Pasquereau. Sensor management in an air defence centre. In *Proceedings of the 1st conference on cognitive systems with interactive sensors*, Paris, France, 15–17 March 2006.

[5] Susanne Biundo, Ruth Aylett, Michael Beetz, Daniel Borrajo, Amedeo Cesta, Tim Grant, Lee McCluskey, Alfredo Milani, and Gèrard Verfaillie. Technological roadmap on planning and scheduling. PLANET European Network of Excellence in AI Planning, 2003. http://www.planet-noe.org.

[6] Erik P. Blash and Susan Plano. JDL level 5 fusion model: user refinement issues and applications in group tracking. In *SPIE Volume 4729: Aerosense*, pages 270–279, Orlando (FL), USA, 1 April 2002.

[7] Fok Bolderheij. *Mission Driven Sensor Management: Analysis, design, implementation and simulation.* PhD thesis, Delft University of Technology, Delft, the Netherlands, November 2007.

[8] Fok Bolderheij and Frans G.J. Absil. Mission oriented sensor management. In *Proceedings of the 1st Conference on Cognitive Systems with Interactive Sensors*, Paris, France, 15–17 March 2006.

[9] Fok Bolderheij, Frans G.J. Absil, and Piet van Genderen. A risk-based object-oriented approach to sensor management. In *Proceedings of the 8th International Conference on Information Fusion*, Philadelphia (PA), USA, 25–29 July 2005.

[10] Fok Bolderheij and Piet van Genderen. Mission driven sensor management. In *Proceedings of the 7th International Conference on Information Fusion*, pages 799–804, Stockholm, Sweden, 28 June – 1 July 2004.

[11] John Boyd. A discourse on winning and losing. AL: Air University Library Document M-U 43947, Maxwell Air Force Base, 1987.

[12] Chenghui Cai and Silvia Ferrari. Information-driven sensor path planning by approximate cell decomposition. *IEEE transactions on systems, man, and cybernetics, Part B: Cybernetics*, 39(3):672–689, June 2009.

[13] Lijuan Cai and Thomas Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the 13th ACM international conference on Information and knowledge management*, pages 78–87, New York (NY), USA, 2004. ACM.

[14] Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, 7:31–54, 2006.

[15] Qu Changwen and He You. A method of threat assessment using multiple attribute decision making. In *Proceedings of the 6th International Conference on Signal Processing*, pages 1091–1095, Beijing, China, 26–30 August 2002.

[16] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, UK, 2nd edition, 2002.

[17] Richard Dedekind. Ueber die von drei moduln erzeugte dualgruppe. *Mathematische Annalen*, 53(3):371–403, September 1900.

[18] Ofer Dekel, Joseph Keshet, and Yoram Singer. Large margin hierarchical classification. In *Proceedings of the 21st International Conference on Machine Learning*, pages 209–216, Alberta, Canada, 4–8 July 2004.

[19] Jan H. van Delft and Herke Schuffel. Human Factors onderzoek voor toekomstige commando centrales KM. Technical Report TNO-TM A-19, Toegepast Natuurkundig Onderzoek, Soesterberg, the Netherlands, 1995.

[20] Arthur P. Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society: Series B*, 30:205–247, 1968.

[21] Jean Dezert and Florentin Smarandache. *Advances and Applications of DSmT for information fusion (collected works)*, volume 1, chapter 3. Partial ordering on hyper-power sets, pages 49–60. American Research Press, Rehoboth, USA, 2004.

[22] Jean Dezert and Florentin Smarandache. *Advances and Applications of DSmT for information fusion (collected works)*, volume 1, chapter 2. The generation of hyper-power sets, pages 37–48. American Research Press, Rehoboth, USA, 2004.

[23] Jean Dezert and Florentin Smarandache. A new probabilistic transformation of belief mass assignment. In *Proceedings of the 11th International Conference on Information Fusion*, pages 1410–1417, Cologne, Germany, 30 June – 3 July 2008.

[24] Jean Dezert and Florentin Smarandache. *Advances and Applications of DSmT for Information Fusion (collected works)*, volume 3, chapter 1. An Introduction to DSmT, pages 1–42. American Research Press, Rehoboth, USA, 2009.

[25] Jean Dezert, Florentin Smarandache, and Milan Daniel. *Advances and Applications of DSmT for Information Fusion (collected works)*, volume 1, chapter 7. A Generalized Pignistic Transformation, pages 143–154. American Research Press, Rehoboth, USA, 2004.

[26] Hans Driessen, J. Zwaga, and Yvo Boers. Radar management for system performance optimization. In *Proceedings of 1st conference on cognitive systems with interactive sensors*, Paris, France, 15–17 March 2006.

[27] Cyril Duron and Jean-Marie Proth. Multifunction radar: Task scheduling. *Journal of Mathematical Modelling and Algorithms*, 1(2):105–116, 2002.

[28] Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 973–978, Seattle (WA), USA, 4–10 August 2001.

[29] Mica R. Endsley, David B. Kaber, and E. Onall. The impact of intermediate levels of automation on situation awareness and performance in dynamic control systems. In *Proceedings of the 6th conference on human factors and power plants: Golbal Perspectives of Human Factors in Power Generation*, pages 7–12, Orlando (FL), USA, 8–13 June 1997. IEEE.

[30] Ozgur Erdinc, Craig Brideau, Peter Willett, and Thiagalingam Kirubarajan. Fast diagnosis with sensors of uncertain quality. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(4):1157–1165, 2008.

[31] Paul C. Etter. *Underwater Acoustic Modeling and Simulation*. Spon press, London, UK, 2003.

[32] Tim P. Gourlay and E.O. Tuck. The maximum sinkage of a ship. *Journal of Ship Research*, 45(1):50–58, March 2001.

[33] Tjerk E. de Greef, H.F.R. Arciszewski, and Jan H. van Delf. Adaptive automation using an object-orientated task model. In *Proceedings of the Conference of the Maritime Systems and Technology for Defence, Security and Safety*, pages 23–30, 2006.

[34] Rolf Haenni and Stephan Hartmann. A general model for partially reliable information sources. In *Proceedings of the 7th International Conference on Information Fusion*, pages 153–160, Stockholm, Sweden, 28 June – 1 July 2004.

[35] David L. Hall. *Mathematical techniques in multi-sensor data fusion.* Artech House, Inc., Norwood (MA), USA, 1992.

[36] David L. Hall and Sonya A.H. McCullen. *Mathematical techniques in multisensor data fusion.* Artech House, Inc., Norwood (MA), USA, 2nd edition, 2004.

[37] Benjamin S. Halpern, Shaun Walbridge, Kimberly A. Selkoe, Carrie V. Kappel, Fiorenza Micheli, Caterina D'Agrosa, John F. Bruno, Kenneth S. Casey, Colin Ebert, Helen E. Fox, Rod Fujita, Dennis Heinemann, Hunter S. Lenihan, Elizabeth M. P. Madin, Matthew T. Perry, Elizabeth R. Selig, Mark Spalding, Robert Steneck, and Reg Watson. A global map of human impact on marine ecosystems. *Science*, 319(5865):948–952, Februari 2008.

[38] Ferdinand van der Heijden, Robert P.W. Duin, Dick de Ridder, and David M.J. Tax. *Classification, parameter estimation and state estimation - an engineering approach using Matlab.* John Wiley & Sons, Chichester, UK, 2004.

[39] Albert G. Huizing and A.A.F. Bloemen. An efficient scheduling algorithm for a multifunction radar. In *IEEE International Symposium on Phased Array Systems and Technology*, pages 359–364, Boston (MA), USA, 10–18 October 1996.

[40] Albert G. Huizing and Arne Theil. *CARPET (Computer-Aided Radar Performance Evaluation Tool): Radar Performance Analysis Software and User's Manual version 1.0.* Artech House, Boston (MA), USA, 1993.

[41] Anthony Hunter and Weiru Liu. A context-dependent algorithm for merging uncertain information in possibility theory. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38(6):1385–1397, 2008.

[42] Fabrice Janez. Optimisation method for sensor planning. In *Proceedings of the 1st conference on cognitive systems with interactive sensors*, Paris, France, 15–17 March 2006.

[43] Jet Propulsion Laboratory — California Institute of Technology — NASA. Internet: http://www2.jpl.nasa.gov/srtm/, cited: 1 July 2009.

[44] D.N. Joanes and C.A. Gill. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189, 1998.

[45] L. Ronnie M. Johansson, Christian Mårtenson, Robert Suzić, and Pontus Svenson. Stochastic dynamic programming for resource allocation. Methodology Report FOI-R–1666–SE, FOI, Command and Control Systems, September 2005.

[46] L. Ronnie M. Johansson and Robert Suzić. Bridging the gap between information need and information acquisition. In *Proceedings of the 7th International conference on Information Fusion*, pages 1202–1209, Stockholm, Sweden, 28 June – 1 July 2004.

[47] Wayne T. Johnson and Ian W. Dall. From kinematics to symbolics for situation and threat assessment. In *Proceedings of Information Decision and Control*, pages 497–502, Adelaide, Australia, 8–10 February 1999.

[48] Jeroen L. de Jong, Gertjan J. Burghouts, Hans Hiemstra, Arjan te Marvelde, Wilbert L. van Norden, Klamer Schutte, and Mink Spaans. Hold your fire!: Preventing fratricide in the dismounted soldier domain. In *Proceedings of the 13th International Command and Control Research and Technology Symposium: C2 for Complex Endeavors*, Bellevue (WA), USA, 17–19 June 2008.

[49] Jeroen L. de Jong and Wilbert L. van Norden. Application of hybrid metaheuristics in sensor management. In *Proceedings of the 18th BeNeLux Artificial Intelligence Conference*, Namur, Belgium, 5–6 October 2006. type B contribution.

[50] Jeroen L. de Jong and Wilbert L. van Norden. Application of metaheuristics in sensor management. In *Proceedings of the 1st international conference on Cognitive Systems with Interactive Sensors*, Paris, France, 15–17 March 2006.

[51] Jeroen L. de Jong and Wilbert L. van Norden. Application of hybrid metaheuristics in sensor management. *Aerospace Science and Technology*, 11(4):295–302, May 2007.

[52] Leon J.H.M. Kester. Model for networked adaptive interactive hybrid systems. In *Proceedings of the 1st Conference on Cognitive Systems with Interactive Sensors*, Paris, France, 15–17 March 2006.

[53] Gary A. Klein and Beth W. Crandall. Recognition-primed decision strategies. Technical Report ARI research note 96–36, Army Research Institute for the Behavioral and Social Sciences, Alexandria (VA), USA, 1996.

[54] Wojciech Komorniczak, Tomasz Kuezerski, and Jerzy F. Pietrasinski. The priority assignment for detected targets in multi-function radar. In *Proceedings of the 13th Int. Conference on Microwaves, Radar and Wireless Communications*, pages 244–247, Wroclaw, Poland, 22–24 May 2000.

[55] Mattias Krysander and Erik Frisk. Sensor placement for fault diagnosis. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38(6):1–1410, 2008.

[56] Gerard J. Kunz, Marianne A.C. Degache, Marcel M. Moerman, Alexander M.J. van Eijk, Filip P. Neele, Stephen M. Doss-Hammel, and Dimitri Tsintikidis. Status and developments in EOSTAR, a model to predict IR sensor performance in the marine environment. In *Proceedings of SPIE Optics in Atmospheric Propagation and Adaptive Systems VII Conference (5572)*, pages 101–111, Maspalomas, Spain, September 2004.

[57] Rikard Laxhammer. Anomaly detection for sea surveillance. In *Proceedings of the 11th international conference on Information Fusion*, pages 55–62, Cologne, Germany, 30 June – 3 July 2008.

[58] A. Vincent van Leijen, Fok Bolderheij, and Wilbert L. van Norden. Unification of radar and sonar coverage modeling. In *Proceedings of the 12th International Conference on Information Fusion*, pages 1673–1678, Seattle (WA), USA, 6–9 July 2009.

[59] Henk A. Lensen, Piet B. W. Schwering, Sebastiaan P. van den Broek, Dirk-Jan J. de Lange, and Rob A. W. Kemp. Application of heterogeneous multiple camera system with panoramic capabilities in a harbor environment. In *Proceedings of SPIE Defense and Security 2009 Symposium — Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VIII (Conference 7305)*, Orlando (FL), USA, 13–17 April 2009.

[60] James Llinas, Christopher Bowman, Galina Rogova, Alan Steinberg, Ed Waltz, and Frank White. Revisiting the JDL data fusion model II. In *Proceedings of the 7th International Conference on Information Fusion*, pages 1218–1230, Stockholm. Sweden, 28 June – 1 July 2004.

[61] M. Margialot and G. Langholz. Design and analysis of fuzzy schedulers using fuzzy lyapunov synthesis. *Engineering Applications of Artificial Intelligence*, 14(2):183 – 188, 2001.

[62] Arnaud Martin and Christophe Oswald. *Advances and Applications of DSmT for Information Fusion (collected works)*, volume 2, chapter 2. A new generalization of the proportional conflict redistribution rule stable in terms of decision, pages 69–88. American Research Press, Rehoboth (MA), 2006.

[63] Arnaud Martin and Christophe Oswald. *Advances and Applications of DSmT for Information Fusion (collected works)*, volume 2, chapter 11. Generalized proportional conflict redistribution rule applied to Sonar imagery and Radar target classification, pages 289–304. American Research Press, Rehoboth, USA, 2006.

[64] Greg A. McIntyre and Ken J. Hintz. A comprehensive approach to sensor management, part I: A survey of modern sensor management systems. *IEEE Transactions on Systems, Man, and Cybernetics*, April, 1999.

[65] Greg A. McIntyre and Ken J. Hintz. A comprehensive approach to sensor management, part II: A new hierarchical model. *IEEE Transactions on Systems, Man, and Cybernetics*, 1999.

[66] Greg A. McIntyre and Ken J. Hintz. A comprehensive approach to sensor management, part III: Goal lattices. *IEEE Transactions on Systems, Man, and Cybernetics*, April, 1999.

[67] Ali Farhang Mehr and Irem Y. Tumer. Risk-based decision-making for managing resources during the design of complex space exploration systems. *Journal of Mechanical Design*, 128(4):1014–1022, 2006.

[68] Bionda Mertens. Reasoning with uncertainty in the situational awareness of air targets. Master's thesis, Delft University of Technology, Delft, the Netherlands, 2004.

[69] Sergio L.C. Miranda, Chris J. Baker, Karl Woodbridge, and Hugh D. Griffiths. Simulation methods for prioritising tasks and sectors of surveillance in phased array radars. *International Journal Simulation*, 5(1–2):18–25, June 2004.

[70] Howard Lee Mohn. Implementation of a tactical mission planner for command and control of computer generated forces in ModSAF (U). Master's thesis, Naval Postgraduate School, Monterey, California, March 1994.

[71] National Oceanic & Atmospheric Administration — National Geophysical Data Center. Bathymetric, topography & relief, Internet: www.ngdc.noaa.gov, cited: 16 January 2009.

[72] Wilbert L. van Norden. Intelligent task scheduling in sensor networks: Introducing three new scheduling methodologies. Master's thesis, Delft University of Technology, Delft, the Netherlands, 22 April 2005.

[73] Wilbert L. van Norden, Fok Bolderheij, and Catholijn M. Jonker. Classification support using confidence intervals. In *Proceedings of the 11th International Conference on Information Fusion*, pages 295–301, Cologne, Germany, 30 June – 3 July 2008.

[74] Wilbert L. van Norden, Fok Bolderheij, and Catholijn M. Jonker. Combining system and user belief on classification using the DSmT. In *Proceedings of the 11th International Conference on Information Fusion*, pages 768–775, Cologne, Germany, 30 June – 3 July 2008.

[75] Wilbert L. van Norden, Jeroen L. de Jong, Fok Bolderheij, and Leon J.M. Rothkrantz. Intelligent task scheduling in sensor networks. In *Proceedings of the 8th International Conference on Information Fusion*, pages 1351–1358, Philadelphia (PA), USA, 25–29 July 2005.

[76] Wilbert L. van Norden and Catholijn M. Jonker. *Advances and Applications of DSmT for Information Fusion (collected works)*, volume 3, chapter 12. Utilizing classifier conflict for sensor management and user interaction, pages 371–386. American Research Press, Rehoboth (MA), May 2009.

[77] Wilbert L. van Norden and Catholijn M. Jonker. Confusion and distance metrics as performance criteria for hierarchical classification spaces. In *Proceedings of the 2009 IEEE/WIC/ACM international joint conference on Web Intelligence and Intelligent Agent Technology*, pages 131–136, Milan, Italy, 15–18 September 2009.

[78] Wilbert L. van Norden and Catholijn M. Jonker. User insisted redistribution of belief in hierarchical classification spaces. In *Proceedings of the 2009 IEEE/WIC/ACM international joint conference on Web Intelligence and Intelligent Agent Technology*, pages 115–122, Milan, Italy, 15–18 September 2009.

[79] Lisa Osadciw and Kalyan Veeramachaneni. Sensor network management through fitness function design in multi-objective optimization. In *Conference Record of the 41st Asilomar Conference on Signals, Systems and Computers*, pages 1648–1651, Pacific Grove (CA), USA, 4–7 November 2007.

[80] Seungyup Paek and Shih-Fu Chang. A knowledge engineering approach for image classification based on probabilistic reasoning systems. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1133–1136, New York (NY), USA, 30 July –2 August 2000.

[81] Dimitri Papageorgiou and Maxim Raykin. A risk-based approach to sensor resource management. In *Advances in Cooperative Control and Optimization*, volume 369-2007 of *Lecture Notes in Control and Information Sciences*, pages 129–144. Springer, Berlin, Heidelberg, October 2007.

[82] Pattern Recoginition Group – Delft University of Technology. PRTools for Matlab. Internet: http://prtools.org/, cited: 13 March 2009.

[83] Wayne L. Patterson. Advanced refractive effects prediction system (AREPS), version 1.0 user's manual. Technical Report A955843, Space and Naval Warfare Systems Centre, San Diego (CA), April 1998.

[84] J. Ross Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco (CA), USA, 1993.

[85] Jens P.Y. Rasmussen. *Information processing and human-machine interaction: an approach to cognitive engineering*. Elsevier Science Publishing Co. Inc., New York (NY), USA, 1986.

[86] Harland Romberg. A game theoretic approach to search. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exibit*, Denver (CO), USA, 14-17 August 2000.

[87] Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7:1601–1626, 2006.

[88] Krispijn A. Scholte. Dynamic bayesian networks for reasoning about noisy target data. Master's thesis, Royal Netherlands Naval College, Den Helder, the Netherlands, 2005.

[89] Krispijn A. Scholte and Wilbert L. van Norden. Applying the PCR6 rule of combination in real time classification systems. In *Proceedings of the 12th International Conference on Information Fusion*, pages 1665–1672, Seattle (WA), USA, 6–9 July 2009.

[90] Odysseas Sekkas, Christos B. Anagnostopoulo, and Stathes Hadjiefthymiades. Context fusion through imprecise reasoning. In *Proceedings of the IEEE International Conference on Pervasive Services*, pages 88–91, Istanbul, Turkey, 15–20 July 2007.

[91] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton (NJ), USA, 1976.

[92] Florentin Smarandache and Jean Dezert. An introduction to the DSm theory for the combination of paradoxical, uncertain and imprecise sources of information. In *Proceedings of the 13th International Congress of Cybernetics and Systems*, Maribor, Slovenia, 6–10 July 2005.

[93] Florentin Smarandache and Jean Dezert. *Advances and Applications of DSmT for Information Fusion (collected works)*, volume 2, chapter 1. Proportional Conflict Redistribution Rules for Information Fusion, pages 3–68. American Research Press, Rehoboth, USA, 2006.

[94] Florentin Smarandache and Jean Dezert. *Advances and Applications of DSmT for Information Fusion (collected works)*, volume 2, chapter 9. Belief Conditioning Rules, pages 237–268. American Research Press, Rehoboth, USA, 2006.

[95] Philippe Smets and Robert Kennes. The transferable belief model. *Journal of Artificial Intelligence*, 66:191–234, 1994.

[96] Phillippe Smets. Data fusion in the transferable belief model. In *Proceedings of the 3rd International Conference on Information Fusion*, pages 21–33, Paris, France, 10–13 July 2000.

[97] Dan Strömberg, Maria Andersson, and Fredrik Lantz. On platform-based sensor management. In *Proceedings of the 5th International Conference on Information Fusion*, pages 1374–1380, Annapolis (ML), USA, 7–11 July 2002.

[98] Matthew E. Taylor, Cynthia Matuszek, Bryan Klimt, and Michael Witbrock. Autonomous classification of knowledge into an ontology. In *Proceedings of the 20th International FLAIRS Conference*, pages 140–145, Key West (FL), USA, 7–9 May 2007.

[99] René Thaens. Sensor scheduling using intelligent agents. In *Proceedings of the 7th International Conference on Information Fusion*, pages 190–197, Stockholm, Sweden, 28 June – 1 July 2004.

[100] Erik Theunissen, Fok Bolderheij, and G.J. Koeners. Integration of threat information into the route (re)planning task. In *Proceedings of the 24th Digital Avionics Systems Conference*, Washington DC, USA, 30 October – 3 November 2005.

[101] Vincent T'kindt and Jean-Charles Billaut. *Multicriteria Scheduling Theory, Models and Algorithms*. Springer-Verlag, Berlin, Germany, 2002.

[102] Mati Tombak, Ain Isotamm, and Tõnu Tamme. On logical method for counting Dedekind numbers. *Lecture notes on Computer Science*, 2138:424–427, 2001.

[103] Lionel Valet and Gilles Mauris. Attribute impact for a seismic image fusion system based on fuzzy rules. In *Proceedings of the 7th International Conference on Information Fusion*, pages 680–685, Stockholm. Sweden, 28 June – 1 July 2004.

[104] Tanja Y.C. van Valkenburg-van Haarst, Wilbert L. van Norden, and Fok Bolderheij. Automatic sensor management: challenges and solutions. In *Proceedings of the SPIE Defense and Security Conference, Optonics and Photonics in Homeland Security (6945)*, pages 694511–1 – 694511–11, Orlando (FL), USA, 16–20 March 2008.

[105] Sebastiaan P. van den Broek, Henri Bouma, Marianne A. C. Degache, and Gertjan Brughouts. Discrimination of classes of ships for aided recognition in a coastal environment. In *Proceedings of SPIE Defense and Security Symposium — Automatic Target Recognition XIX (Conference 7335)*, Orlando (FL), USA, 13–17 April 2009.

[106] T. Wei, Y. Huang, and C.L.P. Chen. Adaptive Sensor Fault Detection and Identification Using Particle Filter Algorithms. *IEEE Transactions*

on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 39(2):201–213, March 2009.

[107] Eric W. Weisstein. Cartesian product. From MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/CartesianProduct.html. Internet, cited: 16 June 2009.

[108] Eric W. Weisstein. Erf. From MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/Erf.html. Internet, cited: May 2008.

[109] Eric W. Weisstein. Newton's method. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/NewtonsMethod.html. Internet, cited: July 2009.

[110] E. Winter and P. Baptiste. On scheduling a multifunction radar. *Aerospace Science and Technology*, 11(4):289–294, 2007.

[111] Guan Xin, Yi Xiao, and He You. An improved Dempster-Shafer algorithm for resolving the conflicting evidences. *International Journal of Information Technology*, 11(12):68–75, 2005.

[112] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90, 1999.

[113] Ted W. Yellman. The three facets of risk. In *Proceedings of the World Aviation Conference*, number 2000-01-5594, pages 10–12, San Diego (CA), USA, 10–12 October 2000.

[114] Lotfi A. Zadeh. On the validity of dempster's rule of combination of evidence. ERL Memo M79/24, University of California, Berkeley (CA), USA, 1979.

[115] Yongmian Zhang and Qiang Ji. Sensor selection for active information fusion. In *Proceedings of the 20th national conference on artificial intelligence*, pages 1229–1234, Pittsburg (PE), USA, 913 July 2005.

List of Acronyms and Abbreviations

A

ADCF	Air Defence and Command Frigate
AI	Artificial Intelligence
AREPS	Advanced Refractive Effects Prediction System

B

BCR	Belief Conditioning Rule

C

C2	Command and Control
CARPET	Computer Aided RADAR Performance and Evaluation Tool
CMS	Combat Management System

D

DBN	Dynamic Bayesian Network
DC	Direction and Control
DisC	Dissimilarity Classifier
DM	Decision Making
DS	Dempster-Shafer

DSmT	Dezert and Smarandache Theory

E

EO	Electro-Optical
EOSTAR	Electro-Optical Signal Transmission And Ranging

F

FPB	Fast Patrol Boat

J

JDL	Joint Director of Laboratories

L

LDC	Linear Distance Classifier

M

MB	Model-Based
MBC	Model-Based Classifier
MFR	Multi Function RADAR

N

NAIHS	Networked Adaptive Interactive Hybrid System
NN	Nearest Neighbour
NOAA	National Oceanic and Atmospheric Administration

O

OODA	Observe-Orient-Decide-Act
OP	Operational Picture

P

PCR	Proportional Conflict Redistribution
PDF	Probability Density Function
PR	Pattern Recognition

R

RADAR	Radio Detection And Ranging
RAM	Range dependent Acoustic Model
RMS	Root-Mean-Square
RNLN	Royal Netherlands Navy
RPDM	Recognition Primed Decision-Making
RPG	Rocket Propelled Grenade

S

SA	Situation Assessment
SONAR	Sound Navigation and Ranging
SVM	Support Vector Machine

T

TA	Threat Assessment
TBM	Transferable Belief Model

U

UAV	Unmanned Aerial Vehicle
UPR	User Preference Redistribution

List of Symbols

Symbol	Description
$\varnothing$	Classical empty-set
$\varnothing_{\mathcal{M}}$	Set containing all labels from D^{Θ} that are constrained, a part of $\mathcal{M}$
$\varnothing_{\mathcal{U}}$	Set containing all labels that are constrained by the user
a	The ancestral order
A_j	The j-th attribute with $j \in \{1, 2, \ldots, J_\ell\}$
b_j	Parameter of uncertainty for Laplace distributed measurement on attribute A_j
$B_a(X)$	The a-th order ancestor confusion on label X
$\widehat{B_a}$	Mean value of the a-th order ancestor confusion
$\mathrm{Bel}(X)$	Credibility of label X
$\mathrm{BetP}(X)$	Pignistic probability of label X
$\mathcal{C}$	Total amount of conflict between sources taking only $\varnothing_{\mathcal{M}}$ into account
CI	Confidence Interval

Symbol	**Description**
D^{Θ}	The hyper-power set constructed using the elements from Θ and the operators $\cap$ and $\cup$
D_1	Subset of the hyper-power set D^{Θ} used in Belief Conditioning Rules
D_2	Subset of the hyper-power set D^{Θ} used in Belief Conditioning Rules
D_3	Subset of the hyper-power set D^{Θ} used in Belief Conditioning Rules
$\mathrm{DSmP}_{\epsilon}(X)$	Generalised probability transform of label X from DSmT
E	Error estimation
f	Type (or family) of sensor tasks
F_1	Performance measure for multi-label learning based on recall and precision
$\mathcal{G}_{\ell}(X)$	Used in PCR6 as factor that ensures a proportional redistribution of conflicting belief from for source ℓ to label X
$\mathcal{H}_a(X)$	Set that contains the unique a-th order ancestral labels of label X
i	Index for labels in Θ with $i \in \{1, 2, \ldots, I\}$
I	The number of labels in Θ
j	Index for attributes with $j \in \{1, 2, \ldots, J_{\ell}$
J_{ℓ}	Number of attributes that the ℓ-th MBC uses
k	Index for specificity levels with $k \in \{1, 2, \ldots, K\}$
K	Number of specificity levels in model $\mathcal{M}$
$\mathcal{K}(X_i)$	Conflict that resides in label X_i due to $\varnothing_{\mathcal{U}}$
$\mathcal{K}_{\text{total}}$	Total amount of conflict introduced by $\varnothing_{\mathcal{U}}$
ℓ	Index for classifiers with $\ell \in \{1, 2, \ldots, L\}$
l	Index for labels

Symbol	**Description**
L	Number of MBCs
$m_\ell(X)$	Generalised basic belief assignment of the ℓ-th MBC to label X
$m_c(X)$	Combined generalised basic belief assignment on label X
$\mathcal{M}$	Fusion model containing Θ and the model contraints $\varnothing_{\mathcal{M}}$
M	Confusion matrix
$M_{i,l}$	Mean belief assigned to X_l when the ground truth is in X_i
n	Index for an element at specificity level k with $n \in \{1, 2, \ldots, N_k\}$
N_k	Number of elements at the k-th specificity level in model $\mathcal{M}$
o	Index for objects
$O_{\Psi,f}$	Operator preference to use sensor Ψ for tasks of family f
$\wp$	Index for labels
p_j	Probability density function belonging to the measurement on A_j
pr_o	Precision of a classifier on object o
P_D	Probability of detection
$\mathrm{Pl}(X)$	Plausibility of label X
q	Index for an element at specificity level k with $q \in \{1, 2, \ldots, N_k\}$
Q_f	Buffer (or queue) containing requested tasks of task type f
r_o	Recall of a classifier on object o
$R(t)$	Risk that is posed to the mission by the object at which task t is directed against

Symbol	Description
$s(X)$	Function that finds all elements from Θ involved in X
$\mathcal{S}_o$	Set of labels for the o-th object
$S(\Psi, t)$	Suitability of sensor Ψ to execute sensor task t
t	Sensor task
u	Index for specificity levels in $\mathcal{M}$ with $u \in \{1, 2, \ldots, K\}$
v	Index for an element in Θ at specificity level k with $v \in \{1, 2, \ldots, N_k\}$
$w_{\Psi,f}$	Calculated weight of buffer Q_f for sensor Ψ
$W_\ell(\alpha)$	Weight function for the ℓ-th MBC, a function of boundary value α
X	A label from Θ
$\overline{X}$	Not X
$X^{\uparrow}$	Set of parent labels of X
$X^{\sqcap}$	Set of bridge labels of X
$X^{\rightarrow}$	Set of labels that intersect with $X^{\sqcap}$ but not with X itself
X^{*}	Set containing labels to which belief assigned to X may be redistributed to by UPR
y_j	All possible values for A_j defined using the mean and a fraction of the square root of the variance
Y	A label from $\varnothing_{\mathcal{U}}$
α	Boundary value
β	Angle with $\beta \in [0, 2\pi]$
$\delta_a(X)$	The RMS distance between the a-th order ancestor labels and the mean value of these elements of label X

Symbol	**Description**
$\widehat{\delta_a}$	Mean value of the RMS distance between a-th order ancestor labels and the mean value at that specificity level
$\Delta_{\mathcal{C}}$	The amount of added value to the total conflict $\mathcal{C}$
$\varphi_\ell(l)$	Function to skip ℓ in a loop that uses l as a counter
$\Phi_{\ell,X}(\alpha)$	Summed value of the membership field Γ_ℓ of label X for the ℓ-th MBC as a function of the boundary value
γ_j	Angles obtained when y_j is re-written in polar coordinates
$\Gamma_{\ell,X}(\vec{y_j})$	Membership field of label X for the ℓ-th MBC with $\ell \in \{1, 2, \ldots, L\}$
λ	Filter threshold
μ_j	Mean value of the measurement on A_j
$\theta_{k,n}$	The n-th element at the k-th specificity level in Θ with $n \in \{1, 2, \ldots, N_k\}$ and $k \in \{1, 2, \ldots, K\}$
$\theta_{k,n}^{\uparrow a}$	The a-th order ancestor element of $\theta_{k,n}$
Θ	Frame of discernment
σ_j	Standard deviation of Gaussian distributed measurement on attribute A_j
$\Omega(k,n) = i$	Mapping function to map element $\theta_{k,n}$ on label X_i
$\Omega^{-1}(i) = (k,n)$	The inverse mapping function
ξ_j	Fraction used to describe y_j
Ψ	Sensor

Curriculum Vitae

Personal Data

Full Name	Willem Leendert van Norden
Title	LTZE2OC ir.
Date of Birth	May 29, 1981
Place of Birth	Katwijk
Nationality	Dutch
✉	CAMS – Force Vision MPC 10A P.O. Box 10000 1780 CA Den Helder the Netherlands
☎	work: +31 223 653894
@	work: W.L.van.Norden@ForceVision.nl home: wilbertvannorden@gmail.com

Qualifications

- Master of Science in Media and Knowledge Engineering from Delft University of Technology (*2005*)
- Graduate from the Royal Netherlands Naval College (*2005*)
- Voorbereidend Wetenschappenlijk Onderwijs (VWO) (*1999*)

Professional experience

2006–present	Commissioned as Business Analyst at Force Vision, Domain Development, Planning and Decision Support
2005–2006	Secondary commission at the Defence Material Organisation for the Shock Trials of the HNLMS Evertsen
2005–2006	Commissioned a/b HNLMS Tromp (Air Defence and Command Frigate)
2004–2005	Masters thesis research at the Delft Cooperation on Intelligent Systems (D-CIS) laboratory in Delft. Title for this thesis is *Intelligent task scheduling in sensor networks; Introducing three new scheduling methodologies*, concluded April 22, 2005
2003	Internship a/b HNLMS Tjerk Hiddes (Multi-Purpose Frigate)
2002–2003	Internship a/b HNLMS Van Amstel (Multi-Purpose Frigate)
1999–2002	Midshipman at the Royal Netherlands Naval College

Publications

2009

- Wilbert L. van Norden and Catholijn M. Jonker. User insisted redistribution of belief in hierarchical classification spaces. In *Proceedings of the 2009 IEEE/WIC/ACM international joint conference on Web Intelligence and Intelligent Agent Technology*, pages 115–122, Milan, Italy, 15–18 September 2009

- Wilbert L. van Norden and Catholijn M. Jonker. Confusion and distance metrics as performance criteria for hierarchical classification spaces. In *Proceedings of the 2009 IEEE/WIC/ACM international joint conference on Web Intelligence and Intelligent Agent Technology*, pages 131–136, Milan, Italy, 15–18 September 2009

- A. Vincent van Leijen, Fok Bolderheij, and Wilbert L. van Norden. Unification of radar and sonar coverage modeling. In *Proceedings of the 12th International Conference on Information Fusion*, pages 1673–1678, Seattle (WA), USA, 6–9 July 2009

- Krispijn A. Scholte and Wilbert L. van Norden. Applying the PCR6 rule of combination in real time classification systems. In *Proceedings of the 12th International Conference on Information Fusion*, pages 1665–1672, Seattle (WA), USA, 6–9 July 2009

- Wilbert L. van Norden and Catholijn M. Jonker. *Advances and Applications of DSmT for Information Fusion (collected works)*, volume 3, chapter 12. Utilizing classifier conflict for sensor management and user interaction, pages 371–386. American Research Press, Rehoboth (MA), May 2009

2008

- Wilbert L. van Norden, Fok Bolderheij, and Catholijn M. Jonker. Combining system and user belief on classification using the DSmT. In *Proceedings of the 11th International Conference on Information Fusion*, pages 768–775, Cologne, Germany, 30 June – 3 July 2008

- Wilbert L. van Norden, Fok Bolderheij, and Catholijn M. Jonker. Classification support using confidence intervals. In *Proceedings of the 11th International Conference on Information Fusion*, pages 295–301, Cologne, Germany, 30 June – 3 July 2008

- Jeroen L. de Jong, Gertjan J. Burghouts, Hans Hiemstra, Arjan te Marvelde, Wilbert L. van Norden, Klamer Schutte, and Mink Spaans. Hold your fire!: Preventing fratricide in the dismounted soldier domain. In *Proceedings of the 13th International Command and Control Research and Technology Symposium: C2 for Complex Endeavors*, Bellevue (WA), USA, 17–19 June 2008

- Tanja Y.C. van Valkenburg-van Haarst, Wilbert L. van Norden, and Fok Bolderheij. Automatic sensor management: challenges and solutions. In *Proceedings of the SPIE Defense and Security Conference, Optonics and Photonics in Homeland Security (6945)*, pages 694511–1 – 694511–11, Orlando (FL), USA, 16–20 March 2008

2007

- Jeroen L. de Jong and Wilbert L. van Norden. Application of hybrid metaheuristics in sensor management. *Aerospace Science and Technology*, 11(4):295–302, May 2007

2006

- Jeroen L. de Jong and Wilbert L. van Norden. Application of hybrid metaheuristics in sensor management. In *Proceedings of the 18th BeNeLux Artificial Intelligence Conference*, Namur, Belgium, 5–6 October 2006. type B contribution

- Jeroen L. de Jong and Wilbert L. van Norden. Application of metaheuristics in sensor management. In *Proceedings of the 1st international conference on Cognitive Systems with Interactive Sensors*, Paris, France, 15–17 March 2006

2005

- Wilbert L. van Norden, Jeroen L. de Jong, Fok Bolderheij, and Leon J.M. Rothkrantz. Intelligent task scheduling in sensor networks. In *Proceedings of the 8th International Conference on Information Fusion*, pages 1351–1358, Philadelphia (PA), USA, 25–29 July 2005

A

Dezert-Smarandache Theory

THE Dezert-Smarandache Theory (DSmT) is a theory for combining evidence that may be paradoxical, incomplete, uncertain and conflicting. Furthermore, this theory deals with frames of discernment that need not be exclusive, i.e., hypotheses may overlap. E.g., consider a police investigation with two suspects: person A and person B. There is evidence to prove the guilt for each of them. Using DSmT we can reason on the evidence that supports the theory that these two persons did it together. Dempster-Shafer theory cannot reason on this possibility since this theory assumes mutually exclusive classes. Both Dempster-Shafer and DSmT however cannot reason on the hypothesis that neither did it. In the theory of evidence, this is referred to as the closed world model. The transferable belief model from Smets, [96] assumes the open-world model. This model may assign to belief to a non-defined solution which in TBM is assigned to the $\emptyset$. The drawback of this approach is the ambiguity between the classical emptyset and the undefined solution, which for this example should be someone rather than nobody.

A.1 The fusion model

Let the frame of discernment be $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$, where each θ_i with $i \in \{1, 2, \ldots, n\}$ is a hypothesis, also called a label. In the general DSmT combination rule, no assumptions are made on the hypotheses other than that all possible labels are represented in Θ, which is referred to as exhaustive labels. A frame with exhaustive labels is used in all theories with the closed world assumption. When no other assumptions are made on the frame of discernment the fusion model is called *free* in DSmT and this model is denoted $\mathcal{M}^f$.

A well known theory for combining evidence is Dempster-Shafer theory. In this theory the frame of discernment is assumed to be exhaustive (closed-world) and it is exclusive (hypotheses cannot occur simultaneously). This fusion model is called *Shafer's* model and is denoted $\mathcal{M}^0$ in DSmT. Within the DSmT framework, the fusion model can take any form between Shafer's model and the free model. All these other models are referred to as hybrid models.

In order to reason on different combinations of hypotheses, the hyper-power set, D^Θ is constructed based on the frame of discernment Θ. This set is constructed using the following three rules:

1. $\varnothing, \theta_1, \theta_2, \ldots, \theta_n \in D^\Theta$;
2. if $X_1, X_2 \in D^\Theta$ then $X_1 \cap X_2 \in D^\Theta$ and $X_1 \cup X_2 \in D^\Theta$;
3. no other elements belong to D^Θ except those obtained using rules 1 and 2.

Assume e.g., a frame of discernment with two elements, $\Theta = \{\theta_1, \theta_2\}$. The hyper-power set then becomes, $D^\Theta = \{\varnothing, \theta_1, \theta_2, \theta_1 \cap \theta_2, \theta_1 \cup \theta_2\}$. These different combinations of the elements from the frame of discernment increase to the extreme when the frame of discernment grows, more specifically, they follow the Dedekind numbers, [102].

Each source that needs to be combined provides its evidence using a belief mapping. I.e., they assign a belief mass, denoted m, to a certain hypothesis from the frame of discernment. For each mapping $m(\varnothing) = 0$ and

$$\sum_{A \in D^\Theta} m(A) = 1$$

holds. This quantity is called the generalised basic belief assignment or the mass.

A.2 Classic DSmT combination rule

Combining information from L sources when assuming the free model can be done with the classic DSmT rule of combination:

$$m_c^f(X) = \sum_{\substack{X_1, X_2, \ldots, X_L\} \in D^\Theta \\ X_1 \cap X_2 \cap \ldots \cap X_L = X}} \prod_{\ell=1}^{L} m_\ell\left(X_\ell\right)$$

where we say that each X_ℓ denotes the element in the hyper-power set on which source ℓ assigns mass to. Since there are L sources we know that $\ell \in \{1, 2, \ldots, L\}$ holds.

Consider two sources e.g., that assign belief to a frame of discernment containing two hypotheses, $\Theta = \{A, B\}$.

	A	B	$A \cup B$
$m_1(.)$	*0.4*	*0.1*	*0.5*
$m_2(.)$	*0.3*	*0.7*	*0*

The classic rule of combination then finds the following values after combining this information.

$$\begin{array}{lclll}
m_c^f(A) & = & 0.4 \cdot 0.3 + 0.4 \cdot 0 + 0.5 \cdot 0.3 & = 0.27 & \\
m_c^f(B) & = & 0.1 \cdot 0.7 + 0.1 \cdot 0 + 0.5 \cdot 0.7 & = 0.42 & \\
m_c^f(A \cup B) & = & 0.5 \cdot 0 & = 0 & \\
m_c^f(A \cap B) & = & 0.4 \cdot 0.7 + 0.1 \cdot 0.3 & \underline{= 0.31} & + \\
 & & & 1.00 &
\end{array}$$

A.3 Conflict redistribution

The classic combination rule may assign masses to all elements in the hyper-power set. In some applications however, some of the combinations might not be physically possible. In the previous example e.g., when A is the hypothesis *it rains* and B is *it does not rain*, these two hypotheses cannot be true at the same time for the same location. Based on the classic combination rule, there now is a conflict between source one and two of $\mathcal{K}_{12}$ =*0.31*. The question is: how should this be dealt with in such a way that all masses are assigned to valid elements in the hyper-power set and that all these valid masses sum up to one while maintaining the closed-world assumption.

The first solution to this problem could be to redistribute conflicting masses to relevant ignorance, which is done in the hybrid DSmT combination rules. E.g., say that $A \cap B$ is physically impossible. The mass assigned to this element by the classic combination rule, is then assigned to $A \cup B$, since the mass belongs to either A or B when they cannot occur simultaneously and there is not enough evidence to assign it to one or the other. This approach leads to:

$$\begin{array}{lclll}
m_c^{hyb}(A) & = & & = 0.27 & \\
m_c^{hyb}(B) & = & & = 0.42 & \\
m_c^{hyb}(A \cup B) & = & \mathit{0} + \mathit{0.31} & \underline{= 0.31} & + \\
 & & & 1.00 &
\end{array}$$

Another approach is to redistribute the conflicting mass to the elements involved in the conflict proportionally to their contribution to the conflict. For two sources this rule is called the Proportional Conflict Redistribution rule 5, PCR5. For multiple sources this rule is generalised in PCR6 and given by

$$m_c^{\mathrm{pcr6}}(X) = m_c^f(X) + \sum_{\ell=1}^{L} \mathcal{G}_\ell(X) \cdot m_\ell(X)^2$$

with

$$\mathcal{G}_\ell(X) = \sum_{\substack{\bigcup_{u=1}^{L-1} X_{\varphi_\ell(u)} \cap X \in \varnothing \\ X_{\varphi_\ell(1)},\ldots,X_{\varphi_\ell(L-1)} \in \left(D^\Theta\right)^{L-1}}} \frac{\prod_{w=1}^{L-1} m_{\varphi_\ell(w)}\left(X_{\varphi_\ell(w)}\right)}{m_\ell(X) + \sum_{w=1}^{L-1} m_{\varphi_\ell(w)}\left(X_{\varphi_\ell(w)}\right)}.$$

The term $\varphi_\ell(w)$ ensures that all elements from the hyper-power set are used except element ℓ — the element under consideration — and it is given by

$$\varphi_\ell \to \begin{cases} \varphi_\ell(w) = w & w < \ell \\ \varphi_\ell(w) = w + 1 & w \geq \ell \end{cases}.$$

Although the equation of PCR6 looks complex, the principle behind it is quite simple. Consider again the example with two information sources in which case the PCR6 rule is the same as the PCR5 rule:

	A	B	$A \cup B$
$m_1(.)$	*0.4*	*0.1*	*0.5*
$m_2(.)$	*0.3*	*0.7*	*0*

The conflict that needs to be distributed when assuming Shafer's model ($\mathcal{M}^0$) is *0.28 + 0.03 = 0.31*. The elements involved in the conflict are A and B, thus the conflict should be proportionally redistributed to them:

$$\begin{array}{lcccccccc} m_c^{pcr5}(A) & = & 0.27 & + & \frac{0.4}{0.4+0.7} \cdot 0.28 & + & \frac{0.3}{0.1+0.3} \cdot 0.03 & = & 0.3943 \\ m_c^{pcr5}(B) & = & 0.42 & + & \frac{0.7}{0.4+0.7} \cdot 0.28 & + & \frac{0.1}{0.1+0.3} \cdot 0.03 & = & 0.6057 \\ m_c^{pcr5}(A \cup B) & = & 0 & & & & & = & \frac{0}{1} \quad + \end{array}$$

We can say that the PCR6 rule is more specific than the hybrid rule of combination since it redistributes conflicting mass to elements that are more specific.

B Classification results

THIS appendix contains the classification results in figures B.1, B.2, B.3 and B.4 for t_0, t_1, t_3,and t_6 respectively obtained by the

- trained;
- Model-Based-trained; and
- Model-Based Classifiers.

Different classifiers of each type are combined with PCR6 for the example from Section 6.4. The example object in this section is represented in table B.1 for the various time steps.

Table B.1: Information based on feedback at different time steps

	Alt	+/-	Speed	+/-	X	+/-	Y	+/-	Size	Feedback
t_0	8	10	8	6	1500	2000	2000	2000	Medium	Altitude
t_1	3	5	8	6	1500	2000	2000	2000	Medium	Altitude
t_2	0	2	8	60	1500	2000	2000	2000	Medium	X
t_3	0	20	8	6	1500	1000	2000	2000	Medium	Y
t_4	0	20	8	6	1500	1000	2000	1000	Medium	Y
t_5	0	2	8	6	1500	1000	2000	300	Medium	X
t_6	0	2	8	6	1500	500	2000	300	Medium	*else*

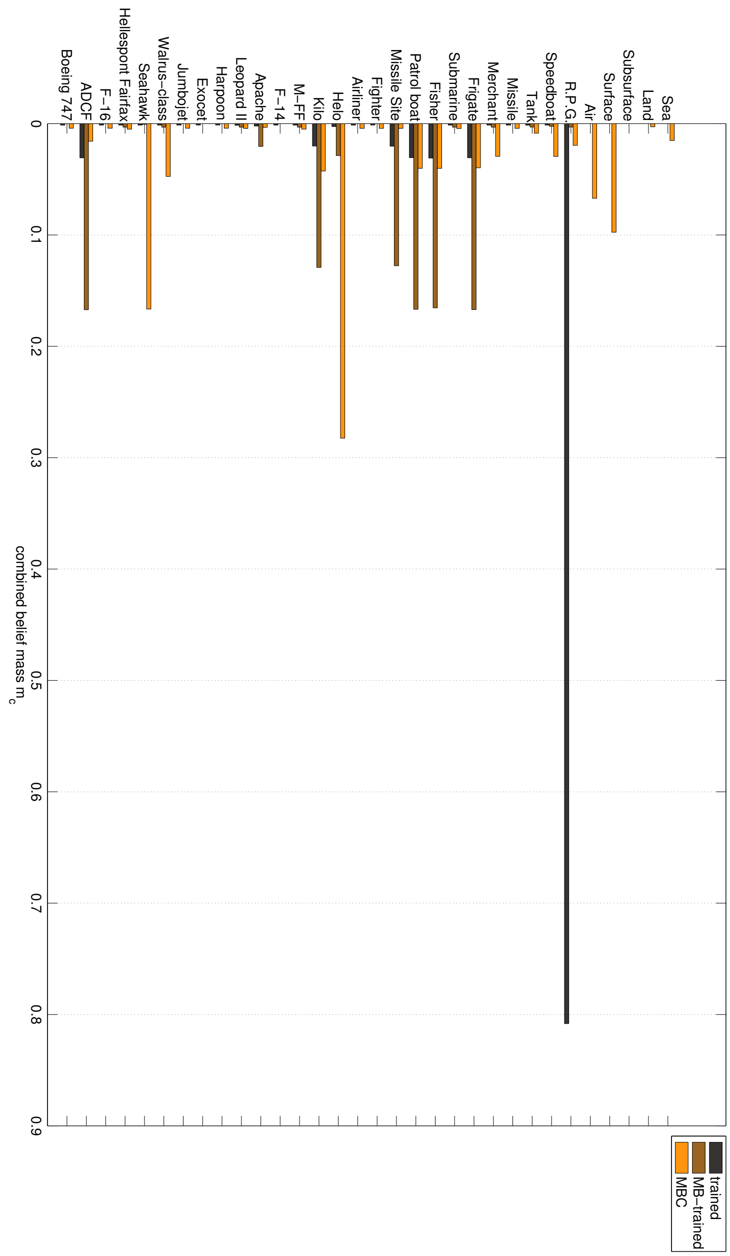


Figure B.1: Classification result for t_0, Section 6.4

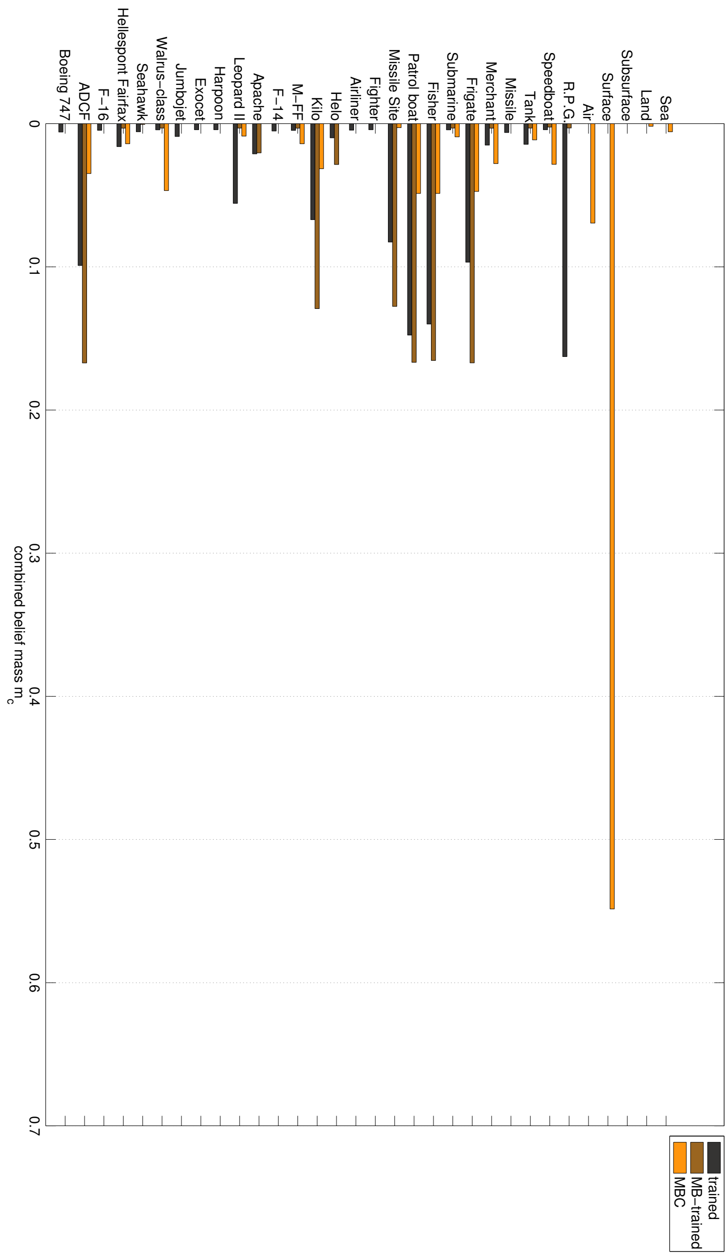


Figure B.2: Classification result for t_1, Section 6.4

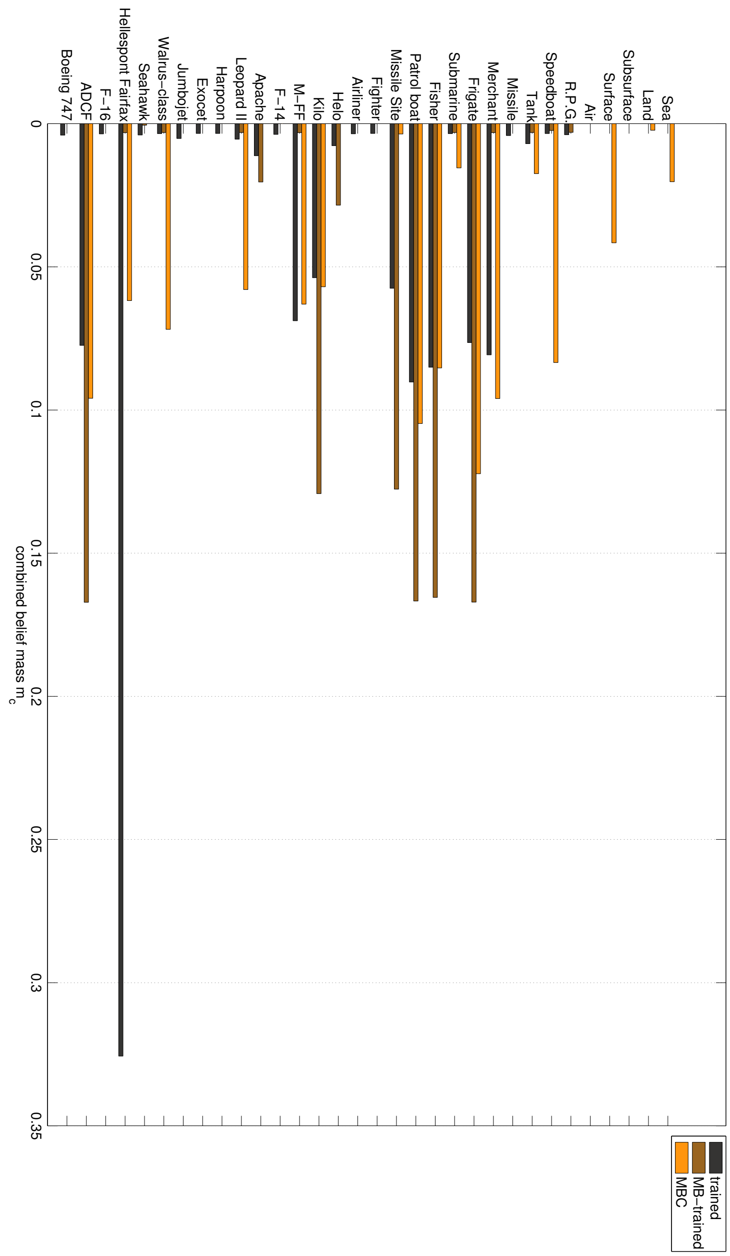


Figure B.3: Classification result for t_3, Section 6.4

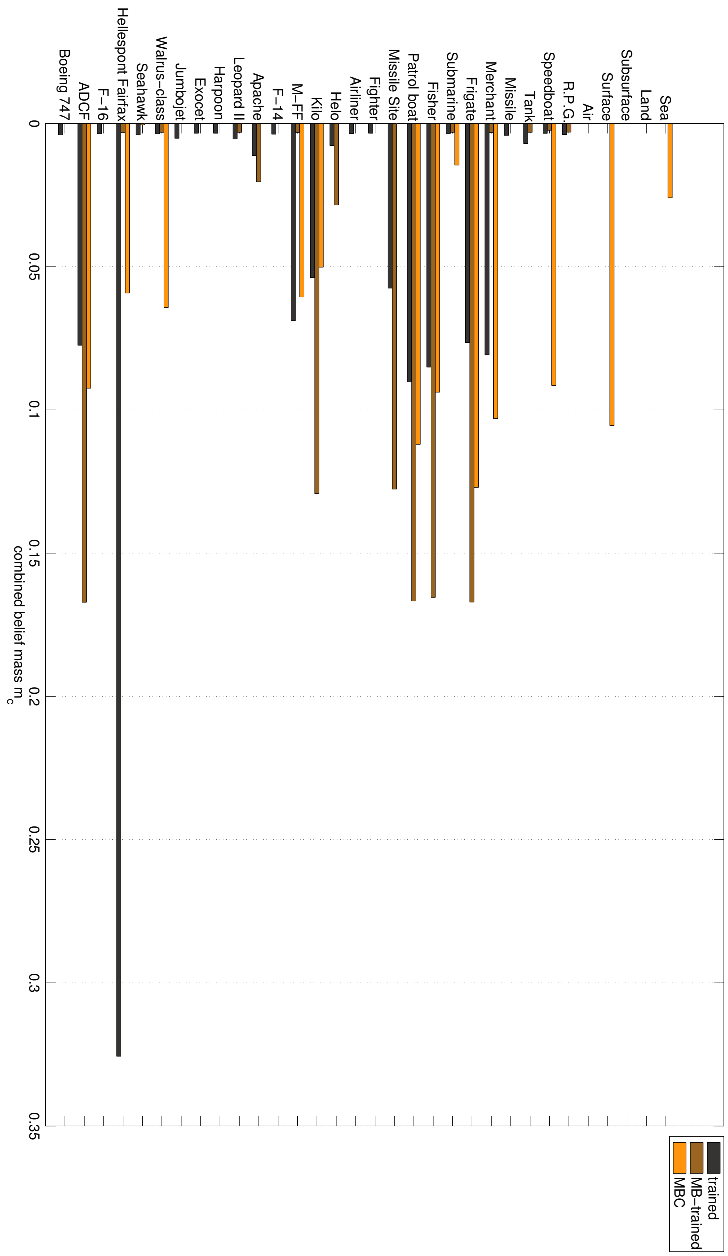


Figure B.4: Classification result for t_6, Section 6.4