MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

IMPLEMENTATION OF MODEL BASED TRANSFERABLE BELIEF MODEL FOR PATTERN RECOGNITION AND CLASSIFICATION

GÜRCAN DURUKAN

525016027

MASTER THESIS

Department of Electrical and Electronics Engineering

Thesis Supervisor

Prof. Dr. Cabir Vural

ISTANBUL, 2020

TABLE OF CONTENTS

ACKNOWLEDGEMENT

First of all, I would like to thank Havelsan for financially supporting this thesis under the project name HVL-SÖZ-19/001. Their financial support is greatly appreciated and acknowledged.

Next, I would like to thank my supervisor Cabir Vural for his never-ending patience and understanding. Without him I could never understand theories used in making of this thesis. I am also thankful Muhammet Altun for bringing forward the subject of Belief Function Theory to our attentions. I am also grateful to my colleague and friend Kaya Arıbaş for working alongside this project with me. Finally I would like to thank my family for their support and help during difficulties I have faced in life.

ÖZET

MODEL TABANLI AKTIRABİLİR İNANÇ MODELİNİN ÖRÜNTÜ TANIMA VE SINIFLANDIRMA PROBLEMLERİNE UYARLANMASI

Bu tezde, Aktarılabilir İnanç Modelinden türetilmiş olan Model Tabanlı Sınıflandırıcı gemilerin otomatik olarak sınıflandırılması amacıyla gerçeklenmiştir. Aktarılabilir İnanç Modeli, Bayesian Teorisi, Bulanık Küme Teorisi, Genel Olasılık Teorisi gibi yöntemlere kıyasla kanıt eksikliğinden doğan belirsizliği daha iyi modelleyebilmektedir. Aktarılabilir İnanç Modeli, farklı kaynaklardan elde edilen bilgilerin kaynaştırılmasına imkan vermektedir. Ayrıca, kaynaştırılacak bilgiler arasında çelişki olup olmadığı Aktarılabilir İnanç Modeli kullanılarak belirlenebilmektedir. Farklı bilgi kaynaklarından gelen çelişmeyen bilgiler kaynaştırılacak yüksek sınıflandırma doğruluğu elde edilmiştir. Model Tabanlı Sınıflandırıcının sınıflandırma doğruluğu oluşturulan yapay veritabanları üzerinde gerçeklenen benzetimlerle belirlenmiştir.

Anahtar kelimeler: Örüntü Tanıma, Sınıflandırma, İnanç Fonksiyonu Teorisi, Aktarılabilir İnanç Fonksiyonu, Çelişki Tespiti

ABSTRACT

IMPLEMENTATION OF MODEL BASED TRANSFERABLE BELIEF MODEL FOR PATTERN RECOGNITION AND CLASSIFICATION

In this thesis, the Model Based Classifier derived from the Transferable Belief Model is implemented for the purpose of automatic ship classification. The Transferable Belief Model models uncertainty caused by lack of evidence better compared to other approaches such as Bayesian Theory, Fuzzy Set Theory and General Probability Theory. The Transferable Belief Model allows one to combine information obtained from different sources or not can be determined by using the Transferable Belief Model. As a result, non-conflicting information coming from different information sources are combined to get high classification accuracy. Artificial learning sets are used in the simulations to obtain classification accuracy of the Model Based Classifier.

Key words: Pattern Recognition, Classification, Belief Function Theory, Transferable Belief Model, Conflict Detection.

LIST OF NOTATIONS

SYMBOL	REFERENCE	DESCRİPTİON
$P(A)$	Equation 2.2	Power Set of A
$m(A)$	Equation 2.3	Mass Function of A
Ω		Frame of Discernment
A		A Hypothesis under a Frame of Discernment
$bel(A)$	Equation 2.7	Belief Function of A
$pl(A)$	Equation 2.10	plausibility function of A
$m_{1\oplus 2}(A)$	Equation 2.18	Dempster's Rule
$m_{1\cap 2}(A)$	Equation 2.22	Dempster Junction Rule
$Betp(w)$	Equation 2.23	Pignistic Transformation
$m[B](A)$	Equation 2.26	Conditional Belief Function
$m^{\Omega\uparrow\Omega x \Theta}(Ax\Theta)$	Equation 2.28	Vacuous Extension
$m^{\Omega x \Theta\downarrow\Omega}(A)$	Equation 2.29	Marginalization
$m[B](A)^{\Omega}$	Equation 2.31	Conditioning on Ω
$m^{\Omega}[B]^{\uparrow\Omega\times\Theta}(C)$	Equation 2.32	Ballooning Extension
$pl[w_k](x)$	Equation 3.4	Conditional Bba of class w_k
$m^{\Omega}_{prior}\{c\}(\{w_k\})$	Equation 3.5	Prior Bba assignment of class w_k
$m^{\Omega}_{combined\ prior}\{c\}(\{w_k\})$	Equation 3.8	Combined Prior Bba Assignment
$\varepsilon_{P(\Omega)}$	Equation 4.1	Generated Vector Space by Focal Elements
$d(m_1, m_2)$	Equation 4.6	Jousselme Distance Between $m_1 and\ m_2$
$D(A_1, A_2)$	Equation 4.7	Jaccard Distance Matrix
$d_{\Omega}(m_1, m_2)$	Equation 4.8	Jousselme Distance Between $m_1 and\ m_2$ on Ω
$Inc(A_1, A_2)$	Equation 4.14	Inclusion Index
$d_{inc}(m_1, m_2)$	Equation 4.15	Degree of Inclusion of m_1 in m_2
$\delta_{inc}(m_1, m_2)$	Equation 4.16	Degree of Inclusion of m_1 and m_2
$conf(m_1, m_2)$	Equation 4.18	Degree of Conflict with AMCA
$q(C_1)$	Equation 5.1	Commonality Function
$q_{1\cap 2}(C)$	Equation 5.3	Computation of Dempster Junction Rule with Commonality Functions

ABBREVIATIONS

BFT	Belief Function Theory
MBC	Model-Based Classifier
TBM	Transferable Belief Model
BBA	Basic Belief Assignment
AMCA	Arnaud Martin’s Conflict Algorithm
PDF	Probability Density Function

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Accurate modeling of the uncertainty has always been a problem in pattern recognition and classification. Until the last two decades this problem had been tried to be avoided as much as possible, since modeling uncertainty is a complex issue. Commonly used frameworks such as Bayesian Theory, Probability theory, Fuzz Clustering Theory Likelihood Theory, Coarse Set Theory are not able to model uncertainty caused by the lack of evidence. In this framework's uncertainty is modeled as a random variable or process [1-3]. Consequently, accurate modeling uncertainty in these frameworks requires infinite amount of data from which statistical outcomes are derived. In order to handle uncertainty in a more robust manner a different framework is needed. For this purpose, Belief Function Theory (BFT) was founded by Arthur P. Dempster.

The discrete representation and modeling of the uncertainty due to lack of evidence in terms of BFT was discussed in detail by Glenn Shafer in his book "A mathematical theory of evidence" in 1976 [4]. Due to its heavy computational loads, BFT did not catch the attention of researchers until 2000s. As personal computers became more powerful, uncertainty modeling via BFT become the most commonly preferred approach.

In this thesis implementation, a classification algorithm called Model-Based Classifier (MBC) is discussed. The MBC is derived from a subjective model referred to a Transferable Belief Model (TBM). Even though TBM is using the same probabilistic infrastructure as BFT, TBM is not a probabilistic model. It evaluates information provided from different sources at credal level then tries to combine them and makes a decision [7]

The MBC is nothing new and a well-defined classification algorithm. Its details are provided in [6].
However, to best of our knowledge there isn't any other implementation of it with the use of data fusion.
Even though there is another classification algorithm using the same framework that is called The Case-Based Classifier. It is not the topic of this thesis. Therefore, this paper solely focuses on the implementation and improvement of the MBC.

1.1 Contribution

The main contribution of this thesis is to create a basic belief assignment (BBA) from an observation obtained by an information source measuring an attribute of an object such as length, weight, shape etc. Also, BBAs formed from different information sources are combined via Dempster -Junction Rule defined under the TBM. Finally, the combined BBA is converted into a probability mass function by means of pignistic transformation. Even though using a single information source for decision making may give satisfactory results for various classification problems, using multiple non-conflicting sources enables one to assign a degree of trust on information provided by each source with the possibility of reducing uncertainty. This issue is discussed in conflict detection and management in order to implement a suitable conflict detection algorithm into the MBC.

1.2 Outline of Thesis

The thesis is divided into of six chapters. In Chapter 2, key concepts of the BFT and fundamental operations used in the MBC are reviewed and several numerical examples are provided. In Chapter 3, structure of the MBC is discussed and the artificial learning set generated for the simulations is explained. In addition, simulation results based on the artificial learning set are carried out and statistical performance of the MBC is investigated. Conflict detection algorithm used in this thesis is discussed in Chapter 4. This chapter also shows how to determine the and provide statistical performance of the conflict detection algorithm implemented in the thesis. In Chapter 5, a speed-oriented optimization algorithm is given for the Dempster Junction Rule. Finally, conclusions are given and possible future research directions are listed in Chapter 6

2 FUNDAMENTAL CONCEPTS OF BELIEF FUNCTION THEORY

This chapter explains the relevant concepts of Belief Function Theory (BFT) that are essential to the Model Based Classifier.

2.1 Types of Belief Functions

BFT is a probability model that allows one to model uncertainty due to lack of evidence. Similar to the sample space in probability theory, in BFT frame of discernment denotes all possible values of an attribute. Any subset of a frame of discernment is called hypothesis. However, differently from the probability theory, Hypothesis in BFT can contain multiple elements from subsets of a frame of discernment. If a hypothesis contains a single element, it is called singleton. If all hypothesis are singleton, then Belief Function becomes Bayesian Belief Function or Bayesian probability

There are two major distinctions between BFT and probability theory. First distinction is that the additivity of mass assignments is not required. For a frame of discernment denoted by Ω and a hypothesis as $A \subseteq \Omega$, a mass assignment given in Equation (2.1) is possible.

$$m(A) + m(\overline{A}) \leq 1 \qquad (2.1)$$

where $\overline{A}$ stands for complement of A

This distinction allows uncertainty caused lack of evidence to be modelled accurately.

The second distinction is that making any interpretation about the subset of a hypothesis without an additional knowledge is not possible. For instance, for a mass function $m(\{x,y\}) = 0.4$ one can not to comment on singleton mass function $m(\{x\})$ and $m(\{y\})$ without any further evidence.

For the purpose of this thesis, there are three equivalent representations under the BFT framework. Those are mass function, belief function and plausibility function

Please note that there is an ambiguity between the expression belief function theory and belief function representation under it. For this reason, belief function will be denoted by bel to prevent any misrepresentation.

2.1.1 Mass Function

The mass function representation is the most fundamental type of belief representation. All other representations can be obtained from the mass function representation. Throughout the thesis mass function and basic belief assignment (bba) are used interchangeably. They represent the same thing.

Two conditions must be satisfied for a valid mass function. First, mass functions of all hypothesis under a frame of discernment should take on values between zero and one. Second, the sum of all mass functions should equal to one. For a frame of discernment Ω and hypothesis $A \subseteq \Omega$, these two conditions are expressed mathematically as:

$$P(A) \rightarrow [0,1] \qquad (2.2)$$

$$\sum_{A \subset \Omega} m(A) = 1 \qquad (2.3)$$

There are two types of assumptions for a given mass function. If the true value of the hypothesis may lie outside the corresponding frame of discernment, then the mass function of the empty set takes on non-zero value. This case is called open world assumption. In the open world assumption one might have

$$m(\emptyset) > 0 \qquad (2.4)$$

On the other hand, true values of all hypothesis are inside the frame of discernment then the so called close world assumption holds. Zero mass is assigned to the empty set under the closed world assumption

$$m(\emptyset) = 0 \qquad (2.5)$$

A mass function defined under the closed world assumption is called a normalized mass function. Equation (2.6) can be used to transform an unnormalized mass function into a normalized one.

$$m(A) = \begin{cases} \dfrac{m(\acute{A})}{1 - m(\emptyset)}, if\ A \neq \emptyset \\ 0, otherwise \end{cases} \qquad (2.6)$$

2.1.2 Belief Function

For a frame of discernment Ω and a hypothesis $A \subseteq \Omega$, belief function representation can be derived from the mass function by using the following equality

$$bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B) \quad , \forall A \subseteq \Omega, B \neq \emptyset \qquad (2.7)$$

The belief function represents the degree of a belief committed to a hypothesis. One natural interpretation of Equation (2.7) is that all belief functions are normalized.

<u>Example 2.1:</u>

For a frame of discernment $\Omega = \{w1, w2, w3\}$ a bba is given as $m^{\Omega}(\{w1, w2\}) = 0.2$, $m^{\Omega}(\{w2\}) = 0.25$, $m^{\Omega}(\{w3, w2\}) = 0.2$, $m^{\Omega}(\{w3\}) = 0.25$, $m^{\Omega}(\phi) = 0.1$

Determine bel({w3, w2})

Solution:

From Equation (2.7)

$$bel(\{w3, w2\}) = \sum_{B \subseteq \{w3, w2\}, B \neq \emptyset} m(B)$$

$$m(\{w3, w2\}) + m(\{w3\}) + m(\{w2\}) = 0.2 + 0.25 + 0.25$$

2.1.3 Plausibility Function

For the same frame of discernment and hypothesis given for a belief function, the plausibility function can be considered as the reverse operation of the belief function. Belief function represents the degree of belief committed to a hypothesis while plausibility function represents the degree of doubt given to the hypothesis. As the two are reverse operations of each other, they can be derived from one another by using Equation (2.8) and (2.9). It is worth mentioning that Equation (2.9) is valid only for normalized belief and plausibility functions.

$$pl(A) = bel(\Omega) - bel(\bar{A}) \qquad (2.8)$$

$$pl(A) = 1 - bel(\bar{A}) \qquad (2.9)$$

In addition, plausibility function can be derived from mass function by means of Equation (2.10)

$$pl(A) = \sum_{B \subseteq \Omega, B \cap A \neq \emptyset} m(B), \forall A \subseteq \Omega \qquad (2.10)$$

Any plausibility function obtained from Equation (2.10) is normalized.

Example 2.2:

For the same frame of discernment and bba given in Example 2.1 determine pl({w3,w2})

Solution:

By using equation (2.10):

$$pl(\{w_3, w_2\}) = \sum_{B \subseteq \{w_1, w_2, w_3\}, B \cap \{w_1, w_2, w_3\} \neq \emptyset} m(B)$$

$$\mathrm{m}(\{w_1, w_2\}) + \mathrm{m}(\{w_3, w_2\}) + \mathrm{m}(\{w_2\}) + \mathrm{m}(\{w_3\}) = 0.9$$

2.1.4 Categories of Belief Functions

If a belief function is normalized with a single focal set it is called categorical belief function. For a categorical belief function there exists only one $A \subseteq \Omega$ such that

$$m(A) = 1, A \subseteq \Omega \qquad (2.11)$$

If a belief function is categorical and its focal set is equal to frame of discernment, it is called vacuous belief function. Having a vacuous belief function means that no information is available at all and there exists total ignorance. For a vacuous belief function, we have

$$m(\Omega) = 1 \tag{2.12}$$

A simple belief function assigns non-zero masses to the frame of discernment and one of its subsets. That is, for a simple belief function, we have

$$m(A) = x, \qquad A \subseteq \Omega \text{ and } 0 \leq x \leq 1 \tag{2.13}$$

$$m(\Omega) = 1 - x \tag{2.14}$$

If frame of discernment is not a focal set the belief function is called dogmatic belief function. For a dogmatic belief function, the following condition must be satisfied.

$$m(\Omega) = 0 \tag{2.15}$$

If a belief function is normalized and all of its focal sets are singletons, it is called Bayesian belief function. For Bayesian belief function, Equation (2.16) and Equation (2.17) always hold.

$$P(A) = bel(A) = pl(A), \forall A \subseteq \Omega \tag{2.16}$$

$$P(a) = m(a), \forall a \subseteq \Omega \tag{2.17}$$

2.2 Combination of Mass Functions

Combination of mass functions (or equivalently fusion) is one of the key concepts in the BFT. The idea behind combining mass functions derived from different information sources is that uncertainty may be reduced if decision making is built upon the combined mass function. However, one must pay attention to conflict that might arise during mass combination. Conflict arises when a pair of focal sets from different mass functions have empty sets as their intersection. If conflict is high, the combination may create a misleading mass function.

To classify the conflict issue further let us investigate a famous example referred to a Zadeh problem [13]. For a frame of discernment Ω={w1,w2,w3}, the following Bayesian belief functions are defined from it:

$m_1(w_1) = 0.2\ , m_1(w_2) = 0.8$

$m_2(w_1 1) = 0.2 \; m_2(w_3) = 0.8$

When these two mass functions are combined, (for the time being, do not bother with the combination rule) the mass function will be $m_{1\oplus 2}(w1) = 1$ even though the belief assigned to w1 on each mass function is 0.2 compared to 0.8 assigned to other hypothesis. This is because the intersection of w2 and w3 is an empty set. On a closed world assumption, the value assigned to the empty set is very useful since it indicates the amount of conflict proportionally.

For the closed world assumption the Dempster's Rule is used for combining two or more mass functions. Dempster's Rule is given by

$$m_{1\oplus 2}(A) = \eta \sum_{B\cap C=A} m_1(B) m_2(C), \forall A \subseteq \Omega, A \neq \emptyset \quad (2.18)$$

$$\eta^{-1} = 1 - \sum_{B\cap C=\emptyset} m_1(B) m_2(C) \quad (2.19)$$

Where η is defined as normalization constant. As the name suggests, η makes sure the that combined mass function is normalized and thus mass of the empty set is equal to zero. In addition, η has one more function in closed world assumption. It is a direct indicator of the amount of conflict between two mass functions and as a result the weight of conflict is calculated from it. The formula for the weight of conflict is given by:

$$Con(m1, m2) = -\log(\eta^{-1}) \quad (2.20)$$

When there are multiple mass functions, weight of conflict is determined additively. That is:

$$Con(m_1, \ldots, m_{n+1}) = Con(m_1, \ldots, m_n) + Con(m_1 \oplus \ldots \oplus m_n, m_{n+1}) \quad (2.21)$$

Note that Equation (2.20) and (2.21) is more of a general mathematical definition of the conflict measure. It is not feasible to use these equations for conflict detection as their computational cost is too high. An alternative approach that can be used in practice for conflict detection is explained in chapter 4.

For the open world assumption, on the other hand, it is better to use Dempster Junction Rule for combination. It is defined by

$$m_{1\cap 2}(A) = \sum_{B\cap C=A} m_1(B)m_2(C), \forall A \subseteq \Omega \qquad (2.22)$$

Just as in the closed world assumption case, the empty set is also direct indicator of the amount of conflict. However, the conflict information is less important compared to the closed world assumption since there is a possibility that true answer may be outside the frame of discernment, though it is possible to determine it. However, if conflict information is not important it is better to normalize the obtained mass function. This is particularly valid when the amount of conflict is high and mass functions are combined recursively but conflict itself is not important.

Example 2.3:

For a frame of discernment of $\Omega = \{w1, w2, w3\}$ two bbas are given as:

$m_1^{\Omega}(\{w1, w3\}) = 0.6$ $m_1^{\Omega}(\Omega) = 0.3$ $m_1^{\Omega}(\phi) = 0.1$

$m_2^{\Omega}(\{w1, w2\}) = 0.2$ $m_2^{\Omega}(\{w2\}) = 0.25$ $m_2^{\Omega}(\{w2, w3\}) = 0.2$ $m_2^{\Omega}(\{w3\}) = 0.25$ $m_2^{\Omega}(\phi) = 0.1$

Determine the combined mass function $m_{1\cap 2}(A)$ for all hypothesis in Ω by using a suitable combination rule.

Solution:

Since both mass functions are normalized Dempster Junction Rule used for combination. From Equation (2.22) mass function is determined as:

$m_{1\cap 2}(\{w1\}) = 0.12$

$m_{1\cap 2}(w3) = 0.12 + 0.15 + 0.075 = 0.345$

$m_{1\cap 2}(\{w2\}) = 0.075$

$m_{1\cap 2}(\{w1, w2\}) = 0.06$

$m_{1\cap 2}(\{w1, w2\}) = 0.06$

$m_{1\cap 2}(\emptyset) = 0.34$

2.3 Pignistic Transformation

It was mentioned earlier that TBM frame-work consists of two levels. First level is the credal level where types of belief functions are assigned and transformed into each other. The purpose of this level is to determine the amount of uncertainty and degree of belief given to each hypothesis. For the most part, decision making is not possible at credal level. For instance, for a frame of discernment $\Omega = \{w_1, w_2, w_3\}$ and mass function assignment whose focal sets are $m(\{w_1, w_2\}) = 0.15,\ m(\{w_1, w_3\}) = 0.15, m(\{w_3\}) = 0.6\ , m(\{\emptyset\}) = 0.1$ it is not possible to make accurate decision even though the highest mass is assigned to singleton focal set. This is because individual masses of the members of the frame of discernment are unknown and the mass of empty set is greater than zero. For these reasons' decisions are made at another level.

The second level is called pignistic level. In this level, basic belief assignment values are transformed into probabilities so that a decision can be made

To make the transition from the credal level to the pignistic level, the so called pignistic transformation is applied to bbas. For an unnormalized bba, the pignistic transformation is defined as

$$BetP(w) = \sum_{w \in A} \frac{m(A)}{|A|(1 - m(\emptyset))}, \qquad \forall A \in \Omega \qquad (2.23)$$

For a normalized bba $1 - m(\emptyset)$ part in Equation (2.23) is omitted. After pignistic transformation, probabilities of singletons are obtained.

Since the pignistic level itself is a probability model, the expected value can be determined from it. In addition, it is also possible to the calculate expected value directly from the pignistic transformation using

$$E(X) = \sum_{w \in \Omega} X(w) BetP(w) \qquad (2.24)$$

Example 2.4:

Apply pignistic transformation to the mass function in Example 2.3 to find probabilities of the singletons

Solution:

The mass function obtained in Example 2.3 is:

$m_{1\cap 2}(\{w1\}) = 0.12$

$m_{1\cap 2}(w3) = 0.12 + 0.15 + 0.075 = 0.345$

$m_{1\cap 2}(\{w2\}) = 0.075$

$m_{1\cap 2}(\{w1, w2\}) = 0.06$

$m_{1\cap 2}(\{w1, w2\}) = 0.06$

$m_{1\cap 2}(\emptyset) = 0.34$

From Equation (2.23), the following results are obtained:

$$Betp(w1) = \frac{m(\{w1\})}{(1-m(\phi)*1} + \frac{m(\{w1,w2\})}{(1-m(\phi)*2} = 0.227$$

$$Betp(w2) = \frac{m(\{w2\})}{(1-m(\phi)*1} + \frac{m(\{w1,w2\})}{(1-m(\phi)*2} + \frac{m(\{w3,w2\})}{(1-m(\phi)*2} = 0.205$$

$$Betp(w3) = \frac{m(\{w3\})}{(1-m(\phi)*1} + \frac{m(\{w3,w2\})}{(1-m(\phi)*2} = 0.568$$

2.4 Conditional Belief Functions

In probability theory, we have conditional probability density functions. Similarly, in BFT conditional belief functions exist. For a hypothesis B and a categorical mass function $m_B(B) = 1$ where $B \subset \Omega$, conditional belief function is denoted by $m[B]$. This means that it is known with absolute certainty that B is true for the given frame of discernment.

Since conditioning a mass function can be regarded as combination of two mass functions under the same frame of discernment, it can be determined with the Dempster's Rule or Dempster Junction Rule with the following expression (2.25):

$$m[B] = m \oplus m_B \qquad (2.25)$$

The explicit form of Equation (2.25) is given as:

$$m[B](A) = \begin{cases} pl(B)^{-1} \sum_{C \subseteq B} m(C \cup A) & \forall A \subseteq B,\ A \neq \emptyset \\ 0, & otherwise \end{cases} \qquad (2.26)$$

The conditional plausibility function is defined as:

$$pl[B](A) = \frac{pl(B \cap A)}{pl(B)}, \forall A \subseteq \Omega \qquad (2.27)$$

2.5 Product Space Operations

When dealing with classification problems under the BFT and TBM frame-works one may encounter with mass functions defined on two or more frames of discernments even though mass functions are mostly defined on a single discrete frame of discernment. Therefore, operations that can extend the single frame of discernments into the joint space or reverse operations that project joint space into one of the frames of discernment would be required. In this section, operations needed for implementing the MBC are defined.

2.5.1 Vacuous Extension and Marginalization

Vacuous Extensions are required when there exist two mass functions defined on different frame of discernments and they need to be processed on the joint frame of discernment. For frames of discernments Ω and Θ, notation $\Omega \uparrow \Omega \times \Theta$ means that a hypothesis $A \subseteq \Omega$ is extended to joint space $\Omega \times \Theta$. Vacuous Extension formula is given in Equation (2.28).

$$m^{\Omega\uparrow\Omega \times \Theta}(Ax\Theta) = \begin{cases} m^{\Omega}(A), & if\ A \subseteq \Omega \\ 0, & otherwise \end{cases} \qquad (2.28)$$

The reverse of Vacuous Extension is the marginalization. It is required when a mass function defined over a joint space is needed to be projected onto one of the frames of discernments it is made up of. The notation is $\Omega \times \Theta \downarrow \Omega$ is used for marginalization for Ω and it is determined with Equations (2.29) and (2.30)

$$m^{\Omega \times \Theta\downarrow\Omega}(A) = \sum_{\{B\subseteq\Omega \times \Theta | \mathrm{Proj}(B\downarrow\Omega)=A\}} m^{\Omega \times \Theta}(B) \qquad (2.29)$$

$$\mathrm{Proj}(B \downarrow \Omega) = \{\omega \in \Omega | \exists\theta \in \Theta,\ \ (\omega,\theta) \in B\} \qquad (2.30)$$

It should be emphasized that when Vacuous Extension and Marginalization operations are performed one after another on a mass function, obtained results are always going to be a Vacuous Belief Function

Example 2.5:

For two discrete frame of discernment defined as Ω={w1,w2} and Θ={q1,q2} the mass function defined over $\Omega \times \Theta$ is given as:

$m^{\Omega x\Theta} = (\{w1, q1\}, \{w2, q2\}) = 0.2,\ m^{\Omega x\Theta} = (\{w1, q1\}, \{w1, q2\}) = 0.3$

$m^{\Omega x\Theta} = (\{w1, q2\}, \{w2, q1\}, \{w2, q2\}) = 0.4,\ m^{\Omega x\Theta} = (\{w2, q1\}) = 0.1$

Determine the mass function defined on Ω by means of marginalization

Solution:

The joint frame of discernment is given by:

$\Omega x\Theta = (\{w1, q1\}, \{w1, q2\}, \{w2, q1\}, \{w2, q2\})$

By using Equations (2.29) and (2.30) marginalization is determined as:

$m^{\Omega x\Theta \downarrow \Omega}(\{w1\}) = 0.3$

$m^{\Omega x\Theta \downarrow \Omega}(\{w2\}) = 0.1$

$m^{\Omega x\Theta \downarrow \Omega}(\Omega) = 0.2 + 0.4 = 0.6$

<u>Example 2.6:</u>

Apply vacuous extension to the obtained mass function in the previous example

Solution:

By using the Equation (2.28) vacuous extension is determined as the following:

$m^{(\Omega x\Theta \downarrow \Omega)\uparrow \Omega x\Theta}(\{w1, q2\}, \{w1, q1\}) = 0.3$

$m^{(\Omega x\Theta \downarrow \Omega)\uparrow \Omega x\Theta}(\{w2, q2\}, \{w2, q1\}) = 0.1$

$m^{(\Omega x\Theta \downarrow \Omega)\uparrow \Omega x\Theta}(\Omega x\Theta) = 0.6$

Please note that compared to the mass function in Example 2.5, some amount of information is lost when marginalization and then vacuous extension are applied one after another. The reason is that marginalization and vacuous extensions are not one to one mapping operations. In addition, if vacuous extension is applied first and marginalization is applied second to any mass function then the obtained results would be a vacuous belief function. Therefore, one can conclude that vacuous extension and marginalization are not commutative with each other

2.5.2 Conditioning and Ballooning Extension

In section 2.4 Conditional Belief Function was defined over a single frame of discernment. However, there are some cases for which conditioning a mass function on frame of discernment other than its original one is required. This is achieved with Conditioning operation.

Assume that there exist two frame of discernments Ω and Θ. Consider two hypotheses denoted by $A \subseteq \Omega$ and $B \subseteq \Theta$. When conditioning hypothesis A with hypothesis B is desired, the first step for achieving this goal is to perform Vacuous Extension operation on hypothesis B with frame of discernment Ω and assume that $m_B^{\Omega x \Theta}(\Omega x B) = 1$. Then, Equation (2.31) is applied to obtain the required conditioning.

$$m^{\Omega}[B](A) = (m_A^{\Omega x \Theta}(Ax\Theta) \cap m_B^{\Omega x \Theta}(\Omega x B))^{\downarrow \Omega} \qquad (2.31)$$

The reverse operation of the conditioning is called Ballooning Extension. It determines the least committed bba for the conditioning operation defined by Equation (2.31). It is computed from:

$$m^{\Omega}[B]^{\uparrow \Omega \times \Theta}(C) = \begin{cases} m^{\Omega}[B](A), & if\ C = (A \times B) \cup (\Omega \times (\Theta \backslash B))\ for\ A \subseteq \Omega\ and\ C \subseteq \Omega\ x\ \Theta \\ 0, & otherwise \end{cases} \qquad (2.32)$$

<u>*Example 2.6:*</u>

Apply conditioning operation on the mass function given in Example 2.4 for the condition of {q2}:

Solution:

This solution is obtained in two stages.

First stage conditional belief function operation is performed for the assumption $m_B^{\Omega\, x\, \Theta}(\Omega x q2) = 1$. Then, we got:

$m^{\Omega\, x\, \Theta} \cap m_Y^{\Omega\, x\, \Theta} \rightarrow$

$m^{\Omega\, x\, \Theta}[q2](\{w2, q2\}) = 0.2$

$m^{\Omega\, x\, \Theta}[q2](\{w1, q2\}) = 0.3$

$m^{\Omega\, x\, \Theta}[q2](\{w2, q2\}, \{w1, q2\}) = 0.4$

$m^{\Omega\, x\, \Theta}[q2](\emptyset) = 0.1$

Second marginalization on Ω is be applied to the mass function calculated at the first stage. The final result is:

$m^{\Omega}[q2](\{w1\}) = 0.3$

$m^{\Omega}[q2](\{w2\}) = 0.2$

$m^{\Omega}[q2](\Omega) = 0.4$

$m^{\Omega}[q2](\emptyset) = 0.1$

Example 2.7:

Apply ballooning extension to the mass function obtained in the previous example.

Solution:

From equation (2.32) the following result is obtained:

$m^{\Omega}\, [q2]^{\Uparrow \Omega\, x\, \Theta}(\{w1, q2\}, \{w2, q1\}, \{w1, q1\}) = 0.3$

$m^{\Omega}\, [q2]^{\Uparrow \Omega\, x\, \Theta}(\{w2, q2\}, \{w2, q1\}, \{w1, q1\}) = 0.2$

$m^{\Omega}\, [q2]^{\Uparrow \Omega\, x\, \Theta}(\Omega\ x\ \Theta) = 0.4$

$m^{\Omega}\, [q2]^{\Uparrow \Omega\, x\, \Theta}(\{w2, q1\}, \{w1, q1\}) = 0.1$

3 MODEL-BASED CLASSIFIER

In this chapter the proposed classifier called model-based classifier (MBC) is developed. In Section 3.1 theoretical background of the MBC is explained and its general notation is given. In Sections 3.2 and 3.3, the implementation of MBC with single sensor and multiple sensors are explained respectively. In Section 3.4, an artificial learning sets are generated for the simulations are discussed. In section 3.5, simulation results based on the learning set generated in section 3.4 is given. In section 3.6, statistical performance of the MBC algorithm is evaluated.

3.1 The MBC and Its General Notations

The MBC is a non-parametric density estimation-based method that uses conditional and prior probabilities to compute posterior probabilities similar to Bayesian Classifiers [8] [9]. To achieve this goal, the MBC requires sets of class data produced from same attributes of objects such as weights, lengths, color etc. Information sources provide attribute data. Please note that the word "agent" and information source are used synonymously in this thesis.

By using this information as training data or a learning set; a sample region is created to assemble conditional bbas for each information sources.

The general notation of MBC is defined as follows:

- $\Omega = \{\omega_k\}_{k=1}^{K}$ is the finite class set or category.
- $\mathcal{X}$ is the *J*- dimensional feature vector space.
- $\{o_i\}_{i=1}^{I}$; o_i is defined as the set of objects
- $x_i \in \mathcal{X}$ the feature vector of object o_i
- $c_i \in \Omega$ is the class to which object o_i belongs to
- $e_i = (x_i, c_i) \in \Omega$ is the current data collection about object o_i
- $\mathcal{L} = \{e_i\}_{i=1}^{I}$ is the learning set

With the help of a learning set MBC is able to find the correct class of an unknown object whose feature vector is known. For this purpose, the first step is to create agent's belief with the help of bbas belief. For $k = 1,2, \dots, K$ let a bba $m^{\mathcal{X}}\{x\}[c = \omega_k]$ be denoted as $m^{\mathcal{X}}[\omega_k]$. This notation shows the agent's belief about object *o* for the hypothesis that "object *o* belongs to class w_k" based on the observed attribute value

Let us assume that there exists a Vacuous Belief Function (VBF) defined on a frame of discernment Ω with feature vector *x* and prior probability *c*. The steps for implementing MBC defined as follows [10], [11]:

- The conditional bba $m^{\mathcal{X}}[\omega_k]$ is extended with ballooning extension to determine $m^{\mathcal{X}}[\omega_k]^{\Uparrow\Omega\times\mathcal{X}}$
- The extended bbas are combined with Dempster Junction Rule:

$$m^{\Omega\times\mathcal{X}} = \cap_{k=1}^{K} m^{\mathcal{X}}[\omega_k]^{\Uparrow\Omega\times\mathcal{X}}$$

- The combined bbas are conditioned with the condition of $\Omega \times \{x\}$
- Marginalization is applied on Ω to obtain the final result.

The formula for computing the bba $m^{\Omega}[x]$ is given by:

$$m^{\Omega}[x] = \bigcap_{k=1}^{K} \overline{\omega_k}^{pl^{\mathcal{X}}[\omega_k](x)} \qquad (3.1)$$

Where $\overline{w_k}$ denotes the conjugate of w_k defined on Ω. The explicit form of Equation (3.1) is given as

$$m^{\Omega}[x](A) = \prod_{\omega_k \in A} pl^{\mathcal{X}}[\omega_k](x) \prod_{\omega_k \in \bar{A}} \left(1 - pl^{\mathcal{X}}[\omega_k](x)\right) \qquad (3.2)$$

$$pl^{\Omega}[x](A) = 1 - \prod_{\omega_k \in A} \left(pl^{\mathcal{X}}[\omega_k](x)\right) \qquad (3.3)$$

If prior bbas assigned based on agent's belief are different from the empty set the respective bbas are combined with one of the Dempster Combination Rules. After the posterior bbas are formed for the MBC the pignistic transformation is applied and the class with the highest probability is selected as the correct class.

3.2 Implementation of Model Based Classifier with a Single Agent

In this chapter the implementation of the MBC for a single agent is discussed. Please note that the word "agent" can refer to various sources. The agent in question could be a single person making a decision or it could be a sensor measuring several attributes of an object such as weight length shape color etc. … However, it is always assumed that a single agent always produces a single feature set or learning set.

Assume that there exists a learning set produced from an agent denoted by L_s for a frame of discernment $\Omega = \{w1, w2, w3, \dots, wv\}$. Further let $N(k) = a_k$ give the number of elements belonging to class w_k inside the a region with radius *r* in the learning set L_s,where $1 \leq k \leq v$. With this definition, the conditional bba for class w_k is calculated by

$$pl[wk](x) = \frac{a_k}{\sum_{p=1}^{v} a_p} \qquad (3.4)$$

In the next step, posterior bbas should be determined from the conditional bbas. Equation (3.2) can be used for that purposed.

In the following step, prior bba is calculated. Let $Y(w_k)$ denote the number of elements that belongs to the class $w_k$in L_s . Then, the prior bba for w_k is obtained from

$$m_{prior}^{\Omega}\{c\}(\{w_k\}) = \frac{Y(w_{k)}}{\sum_{l=1}^{v} Y(w_{l)}} \qquad (3.5)$$

In the final step of the implementation, posterior and prior bbas are required. This is done by using Equations combined (3.6) and (3.7) given below:

$$m^{\Omega}[x](\{w_k\}) = m_{prior}^{\Omega}\{c\}(\{w_k\}) x \left(\sum_{n=1}^{2^v} f(\theta_n) \right) \qquad (3.6)$$

where

$$f(\theta_n) = \begin{cases} m_{posterior\ combined}[x](\theta_n), w_k \in \theta_n \\ 0, \quad otherwise \end{cases} \qquad (3.7)$$

Once the final conditional bba is obtained, the pignistic transformation is applied to make decision.

3.3 Implementation of Model Based Classifier with Multiple Agents

In the previous section the implementation of the MBC with a single agent was discussed. Here its implementation with multiple agents is explained. As mentioned previously, each agent is assumed to produce its own learning set. Therefore, if there exists n agents then there is going to be n amount of learning sets denoted by $LS_1, LS_2, \; LS_3, \dots . LS_n$

For a given learning set $LS_m \; 1 \leq m \leq n$ and a class $w_k \; 1 \leq k \leq v$, the conditional bbas are calculated with Equation (3.4). Then the posterior bba is calculated from conditional bba by using Equation (3.2) similar to what was done for a single agent. Once posterior bbas are obtained for each learning set they are fused or combined. This is achieved with the Dempster Combination Rule defined in Equation (2.22).

In the next step, prior bbas are obtained. Let the notation $Y_m(w_k)$ give the number of elements that belongs to w_k inside the learning set LS_m. Then, the combined prior bba for class w_k is calculated by:

$$m^{\Omega}_{combined\ prior}\{c\}(\{w_k\}) = \frac{\sum_{l1=1}^{n} Y_{l1}(w_k)}{\sum_{l2=1}^{n} \sum_{l3=1}^{v} Y_{l2}(w_{l3})} \qquad (3.8)$$

Once both combined posterior and prior bbas are obtained they are combined by using Equation (3.6) and (3.7). Finally pignistic transformation is applied for decision making.

3.4 Artificial Learning Sets Used in Simulations

Due to the difficulties for obtaining real data for the implementation of the MBC, artificially generated learning sets are formed for the simulations. For this purpose, six ships were chosen from the book entitled “Modern Deniz Sistemleri” by Sami [12]. The selected ship classes and their respected lengths are given in the Table 3.1

Table 3.1 *Ship Classes and their lengths in meters*

Ship Classes	Ship Lengths
w_1	24.5
w_2	32
w_3	61.9
w_4	56.2
w_5	117.5
w_6	130

When generating learning sets each ship length is assumed to be modeled by a Gaussian random variable whose mean value corresponds to its length. Variance value for each ship model is inversely proportional to its length meaning that for a long ship variance is smaller and for a short ship it is big. With the help of this idea a two dimensional hypothetical space is created where one axis shows ship's length and other shows how much deviation is allowed from the mean value in percentage value.

Two learning sets were generated. The number of elements for each class in each learning set is random. Also, if any generated value for a given class deviates more than two times of the respective variance it is considered as outlier and not included int the learning set. The learning set are called S_1andS_2. They are given in Figures 3.1 and 3.2 respectively.

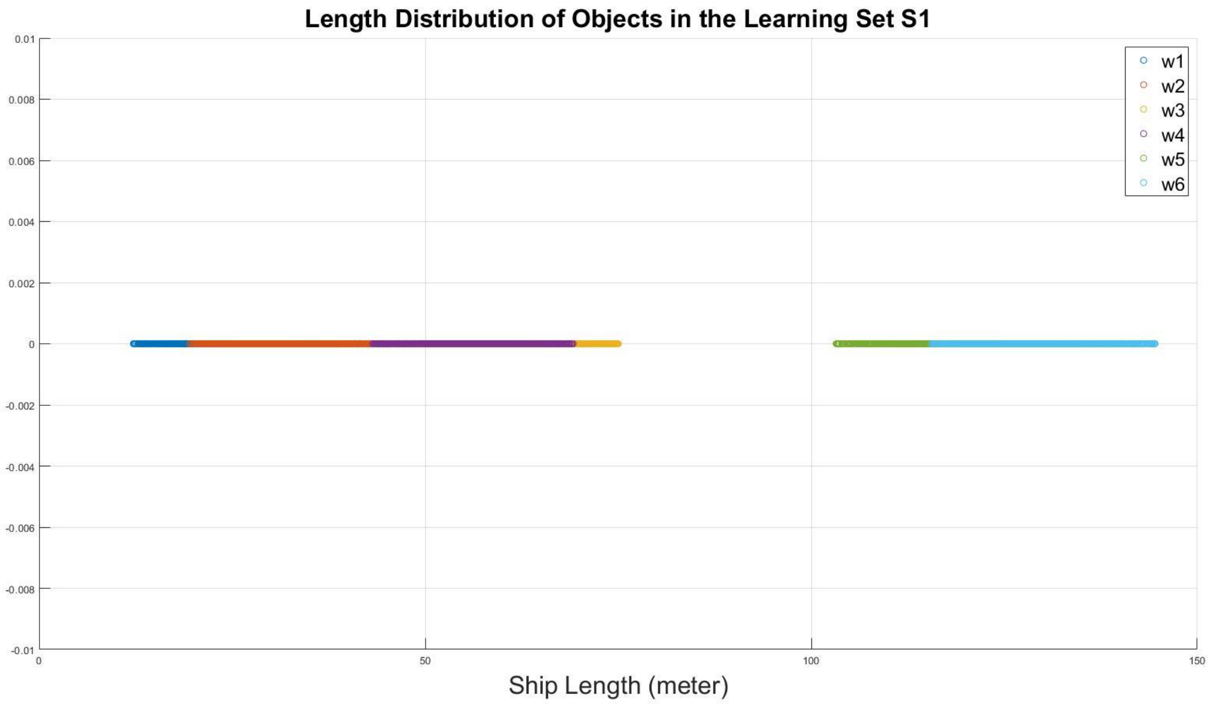

Figure 3.1 Length Distribution of Objects in the learning set S_1

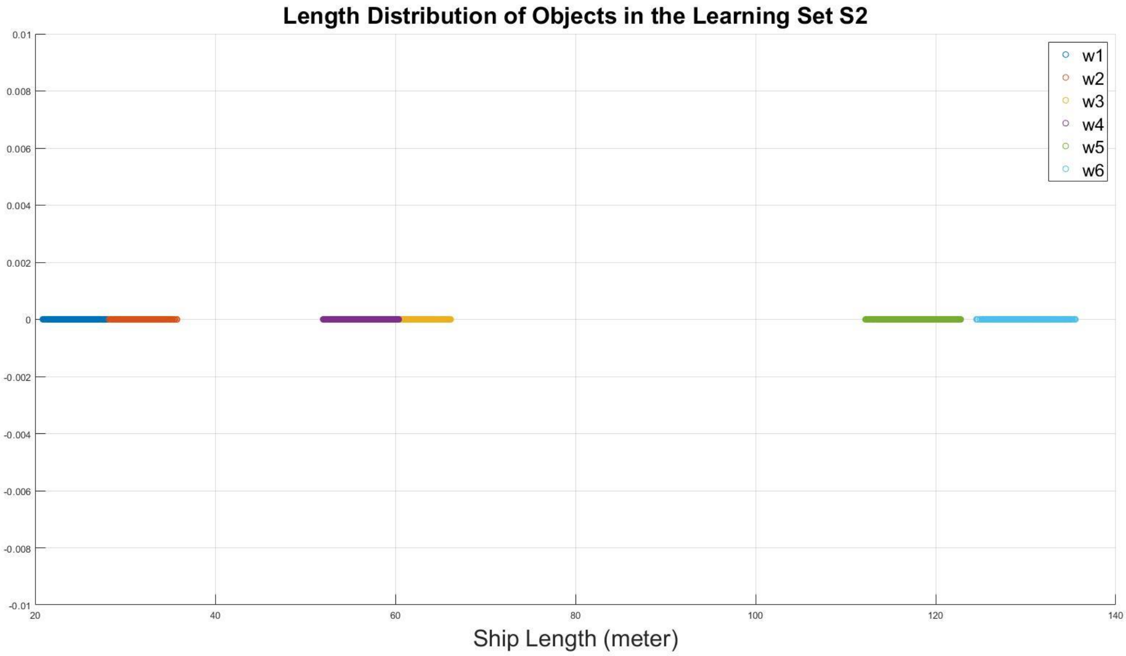

Figure 3.2 Length Distribution of Objects in the learning set S_2

The number of elements for each class in each learning set is given at Tables 3.2 and 3.3 respectively.

Table 3.2 *Number of Elements in the learning Set* S_1

Learning Set: S_1	
Ship Classes	**Number of Elements**
w_1	2209
w_2	2174
w_3	1985
w_4	2174
w_5	1612
w_6	1857

Table 3.3 *Number of Elements in the learning Set* S_2

Learning Set: S_2	
Ship Classes	**Number of Elements**
w_1	1267
w_2	1011
w_3	1855
w_4	2043
w_5	2270
w_6	1090

Learning set elements are generated in Matlab development environment. Pease note that for each learning set a Matlab Array is formed. The number of matrix inside each learning set is equal to class sizes and number of elements inside each matrix shows the data number that belongs to corresponding class. To increase precision, they are stored in “double” format. Figures 3.3 and 3.4 below show a sample Matlab array named as database and one of the class matrixes inside the database, respectively.

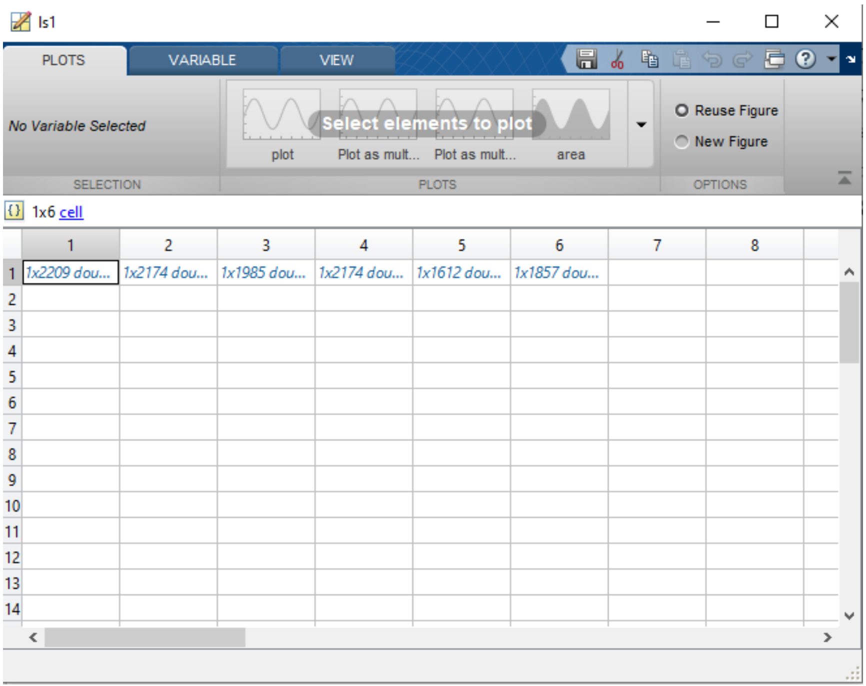

Figure 3.3 Matlab Array called Database

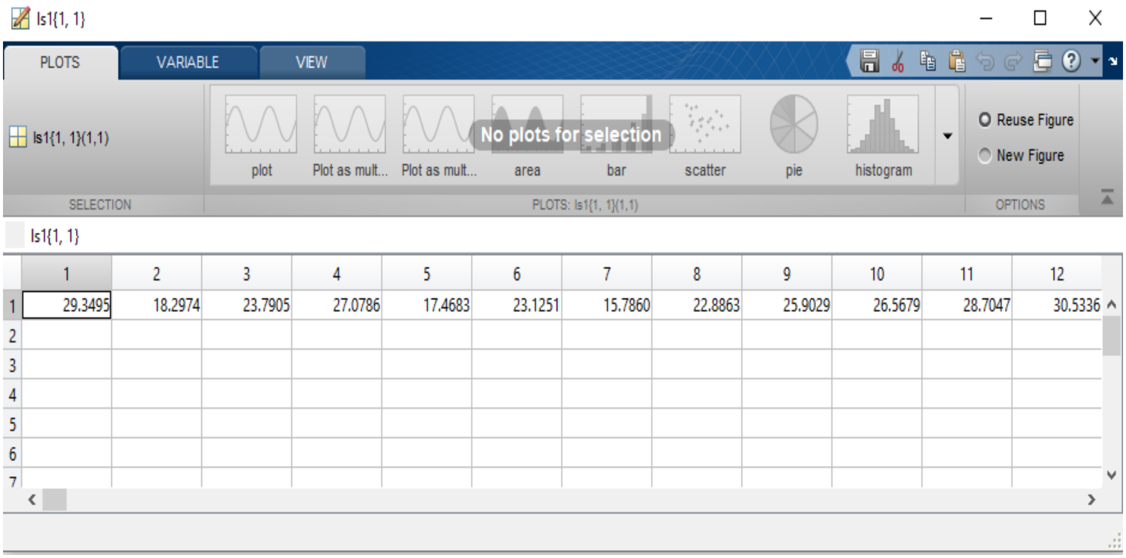

Figure 3.4 One of the Class Inside the Database

3.5 Simulation Results

First, the MBC is implemented with $S_1$and S_2learning sets separately. During the application process each unknown input is generated with the methodology explained in section 3.4. The hypothetical space used during the generation of learning sets are used to determine the variance values that are also taken as sampling interval radius. The results shown in Tables 3.4 and 3.5 are obtained after the algorithm explained in Section 3.2 was run.

Table 3.4 Simulation Results for the Learning Set S_1

Learning Set: S_1		Class Probabilities after Pignistic Transformation					
		w_1	w_2	w_3	w_4	w_5	w_6
Input	w1	0.5349	0.4637	0.0004	0.0004	0.0004	0.0004
	w2	0.2077	0.7899	0.0006	0.0006	0.0006	0.0006
	w3	0.0004	0.0004	0.5942	0.4041	0.0004	0.0004
	w4	0.007	0.0326	0.2279	0.7373	0.0007	0.0007
	w5	0.0006	0.0006	0.0006	0.0006	0.6274	0.3702
	w6	0.0016	0.0016	0.0016	0.0016	0.0016	0.9921

Table 3.5 Simulation Results for the Learning Set S_2

Learning Set: S_2		Class Probabilities after Pignistic Transformation					
		w_1	w_2	w_3	w_4	w_5	w_6
Input	w1	0.9919	0.0016	0.0016	0.0016	0.0016	0.0016
	w2	0.0014	0.9928	0.0014	0.0014	0.0014	0.0014
	w3	0.0007	0.0007	0.7329	0.2643	0.0007	0.0007
	w4	0.0009	0.0009	0.0009	0.9957	0.0009	0.0009
	w5	0.0008	0.0008	0.0008	0.0008	0.9962	0.0008
	w6	0.0037	0.0037	0.0037	0.0037	0.0778	0.9074

Each ship class is shown to be classified correctly according to Table 3.4 and 3.5. However, when class attributes are intersecting with each other, the gap between the probability for the correct class and the probability for the other classes decline since intersection increases uncertainty. Despite this fact, correct class probabilities are separate enough from other probabilities to make correct classification.

Next both learning sets are evaluated together with the algorithm detailed in Section 3.3. The corresponding results are given in Table 3.6

Table 3.6 Simulation results for the Combined Learning Set

S_1 and S_2 Learning Sets Combined		Class Probabilities after Pignistic Transformation					
		w_1	w_2	w_3	w_4	w_5	w_6
Input	w1	0.9983	0.0017	0	0	0	0
	w2	0.0003	0.9997	0	0	0	0
	w3	0	0	0.8174	0.1826	0	0
	w4	0	0.0001	0.0003	0.9996	0	0
	w5	0	0	0	0	0.9994	0.0006
	w6	0	0	0	0	0.0001	0.9999

From Table 3.6 it is obvious that observed probabilities for the correct classes increases significantly when learning sets are evaluated together. This is even more apparent when deciding between two classes that have a large amount of area intersecting with each other. One can deduce that using multiple information sources seems to increase the pignistic probability for the correct class compared to using only one information source. This is because combining multiple information sources reduce uncertainty.

Please note that during section 3.5 only a single classification result for each of the classes are given since multiple input data for each class would be redundantly takes space in the thesis. Instead in Section 3.6 statistical evaluation the algorithm is discussed in order to show each ship classes can correctly be classified using the MBC with multiple input values.

3.6 Statistical Evaluation of the MBC

In this part, of the thesis simulation results of the MBC is evaluated statistically. For the statistical evaluation 50 inputs are generated with the same method explained in Section 3.4. Precision recall and F-beta score are used as performance measures. Also, confusion matrices are formed for all three cases.

With accuracy measure calculates the rate of correct classification to total number of samples. Precision measure evaluates the rate true positive cases to number of samples that classified as positive. Recall measures computes the rate of true positive cases to number of samples that should have been classified as positive. Finally, F-beta scores which is derived from precision and recall, measures any given system's statistically correct classification performance.

Confusion matrices resulting from processing inputs for each class in the learning set S_1 are given in the table 3.7. Please note that confusion matrixes are in the form of:

$$\begin{bmatrix} True\ Positive & False\ Positive \\ False\ Negative & True\ Negative \end{bmatrix}.$$

Table 3.7 Confusion Matrices resulting from after processing inputs for each class in the learning set S_1

Confusion Matrixes for the learning set S_1					
w_1	w_2	w_3	w_4	w_5	w_6
$\begin{bmatrix} 42 & 14 \\ 8 & 182 \end{bmatrix}$	$\begin{bmatrix} 36 & 8 \\ 14 & 188 \end{bmatrix}$	$\begin{bmatrix} 26 & 11 \\ 24 & 198 \end{bmatrix}$	$\begin{bmatrix} 39 & 24 \\ 11 & 185 \end{bmatrix}$	$\begin{bmatrix} 41 & 10 \\ 9 & 183 \end{bmatrix}$	$\begin{bmatrix} 40 & 9 \\ 10 & 184 \end{bmatrix}$

In order to understand the statistical behavior of the system, precision and recall are calculated from the values in Table 3.7. In Table 3.8, measures for each class as well as system average are provided.

Table 3.8 Precision and Recall for each class in the learning set S_1

Precision and Recall Measurement for Agent S_1							
	w_1	w_2	w_3	w_4	w_5	w_6	average
Precision	0.75	0.82	0.70	0.62	0.80	0.82	0.752
Recall	0.84	0.72	0.52	0.78	0.82	0.80	0.746

For the intended classification purpose in this thesis both precision and recall are equally important. That is why the parameter β is taken as 0.5. The corresponding F-beta measurement scores are given in Table 3.9.

Table 3.9 F-beta Measurement Score for the learning set S_1 (β=0.5)

F-beta Measurement Scores (learning set: S_1)						
w_1	w_2	w_3	w_4	w_5	w_6	average
0.79	0.77	0.60	0.69	0.81	0.81	0.749

Table 3.9 shows that while the MBC's general classification performance is relatively high it does not provide enough correct classification for every class. For example, F-beta measurements scores for the classes w3 and w4 are quite low compared to those of the other classes. The main reason for this situation can easily be recognized by examining Figure 3.1 showing the distribution of objects in the learning set learning set S_1. In Figure 3.1 there exists a wide overlapping area for the classes w3 and w4 that causes uncertainty

for both classes. As a consequence, F-beta measurement scores of them are quite less than

Table 3.10 Resulting Confusion Matrices after processing inputs for each class in the learning set S_2

Confusion Matrixes for the learning set learning set S_2					
w_1	w_2	w_3	w_4	w_5	w_6
$\begin{bmatrix} 50 & 0 \\ 0 & 247 \end{bmatrix}$	$\begin{bmatrix} 50 & 0 \\ 0 & 247 \end{bmatrix}$	$\begin{bmatrix} 48 & 1 \\ 2 & 249 \end{bmatrix}$	$\begin{bmatrix} 49 & 2 \\ 1 & 248 \end{bmatrix}$	$\begin{bmatrix} 50 & 0 \\ 0 & 247 \end{bmatrix}$	$\begin{bmatrix} 50 & 0 \\ 0 & 247 \end{bmatrix}$

the system average.

Next, a similar analysis is performed for the learning set learning set S_2 with the same methodology. The respective confusion matrix is given at the Table 3.10

Precision and Recall measurement can be calculated from Table 3.10. They are given in Table 3.11

Table 3.11 Precision and Recall Measures for the learning set S_2

Precision and Recall for S_2							
	w_1	w_2	w_3	w_4	w_5	w_6	average
Precision	1	1	0.98	0.96	1	1	0.990
Recall	1	1	0.96	0.98	1	1	0.990

F-beta measurements for the learning set S_2 computed from Table 3.11 are provided in Table 3.12 for $\beta = 0.5$

The result shown in Table 3.12 indicate classification accuracy is high for all classes. The reason behind this outcome is the same with the previously explained one. When the learning set S_2 is examined, classes are seem to be well separated as there are few and limited overlapping regions between any classes.

Table 3.12 F-beta Measurement Score for the learning set S_2 ((β=0.5)

F-beta Measurement Scores for S_2						
w_1	w_2	w_3	w_4	w_5	w_6	average
1	1	0.97	0.97	1	1	0.990

When the obtained F-beta measure for both S1 and S2 are evaluated together one natural question is the following: Is it possible to improve the results obtained from $S_1$with the help of results obtained for S_2? To answer this question statistical evaluation of the multi agent classification of the MBC for the learning sets $S_1$and S_2 is examined next.

First confusion matrixes obtained from combined evaluation of $S_1$and $S_2$are formed. Table 3.13 shows the resulting confusion matrices.

Table 3.13 Acquired Confusion Matrix after processing inputs for each class for the learning sets $S_1$and S_2

Confusion Matrixes for Combined Evaluation of Agent S1 and S2					
w_1	w_2	w_3	w_4	w_5	w_6
$\begin{bmatrix} 50 & 0 \\ 0 & 249 \end{bmatrix}$	$\begin{bmatrix} 50 & 0 \\ 0 & 249 \end{bmatrix}$	$\begin{bmatrix} 49 & 0 \\ 1 & 250 \end{bmatrix}$	$\begin{bmatrix} 50 & 1 \\ 0 & 249 \end{bmatrix}$	$\begin{bmatrix} 50 & 0 \\ 0 & 249 \end{bmatrix}$	$\begin{bmatrix} 50 & 0 \\ 0 & 249 \end{bmatrix}$

Similar to previous cases precision and recall measures are calculated for multi agent system. It is given in the Table 3.14

Table 3.14 Precision and Recall Measures for the learning sets $S_1$and S_2

Precision and Recall Measurement for Agent S1 and S2							
	w_1	w_2	w_3	w_4	w_5	w_6	average
Precision	1	1	1	0.98	1	1	0.996
Recall	1	1	0.98	1	1	1	0.996

For the last step F-beta measurement score is calculated for the multi-agent system this time. Results are given in the Table 3.15.

Table 3.15 F-beta Measurement Score for the learning sets S1 and S2

F-beta Measurement Scores for Agent $S_1$and S_2						
w_1	w_2	w_3	w_4	w_5	w_6	average
1	1	0.99	0.99	1	1	0.996

When Table 3.15 is observed and results are compared to Table 3.9 and Table 3.12 it is clearly observed that evaluation of multiple learning sets have positive impact on correct classification performance. It is more significant for the agent S1 since it has the lower classification performance. However, classification performance is also increased for the learning set S_2 as well. Based on this observation it can be concluded that when multiple learning sets are evaluated together classification performance increases compared to the case in which only one learning set is used. For instance, F-beta measurement score for the ship class w_3 is 0.6 for the learning set S_1 and 0.97 for the learning set S_2. However, the combined F-beta measurement score for the same class is 0.99 which is higher than the any of the learning sets F-beta measurement score when they are evaluated separately.

4 CONFLICT DETECTION

Conflict arises during information fusion in the framework of Belief Function Theory. Most of the time sources from which data are obtained are imperfect. Even though the cause of imperfections can be in many forms, it can be divided into two major categories [14]. First cause is the lack of trust to data provided by sources. Second cause is that one obtains conflicting data from the same information source under the same conditions. For the latter cause of imperfection, it is not possible to apply any sort of conflict management for the purpose of information. However, for the former cause, conflict detection and management are entirely possible with degree of uncertainty. In the scope of this thesis, conflicts resulted from the former category are going to be evaluated.

Under perfect conditions there would not be need for either information fusion nor conflict detection since data would be perfectly gathered and classification could be achieved without errors. Under real conditions, however, this is not the case. There are many forms of imperfection that make the data obtained from an information source questionable. For instance, the gathered data could be incomplete, measurement could have been taken with an inadequate measurement tool or there could be background affecting the information source's reliability. All of these and even more factors are the cause of uncertainty for the given information source [15].

As a consequence, one must determine the degree of uncertainty of data provided by a source in order to achieve satisfactory classification results. Unfortunately, making a correct judgement about uncertainty is not possible by using the single source on its own. However, uncertainty can be determined with the use of multiple imperfect information sources and the degree of uncertainty among different sources can be obtained. Determining uncertainty requires data fusion algorithm such as Dempster Rule or Dempster Junction rule explained in the Section 2.2.

The idea behind conflict detection and management is that if data obtained from several information sources are consistent with each other, correct classification is still possible. That is why devising an algorithm checking the consistency of information sources relative to each other is essential. This is called conflict detection. With a conflict detection algorithm, checking if data coming from discrete various information sources are conflicting with each other or not can be achieved. Conflict detection results can be used to determine the degree of trust that can be put in to the end result computed from a data fusion algorithm.

In BFT conflict detection and management is still an open-ended discussion and many conflict detection algorithms exist. To the best of our knowledge, a general conflict detection solution does not exist. Moreover, there is not a well-defined solution to fuse highly conflicting masses [16]. For this reason, several conflict detection algorithms were examined in order to find a suitable conflict detection algorithm for the proposed classifier [14], [15], [16], [17].

Three measures were taken into account to assess conflict detection algorithms. The measures are accuracy, convergence for thresholding, and speed.

Main objective of a conflict detection algorithm is to find out whether information obtained from different sources are consistent with each other or not. If an algorithm could not detect the conflict accurately enough for the MBC algorithm; then it would not be possible to designate the degree of certainty to the obtained results and identify the conflict threshold correctly. For this reason, the first measure is the accuracy. It is the most important evaluation criteria for the selection of an algorithm.

In the TBM framework a conflict between any two mass functions is a number between 1 and 0 where 1 means that mass functions are completely inconsistent with each other (degree of conflict is maximum) and 0 means that conflict does not exist. Since MBC algorithm is concerned with thousands of mass functions, detecting conflict requires assigning a threshold. Threshold assignment would be easier if the conflict detection results converged to two different discreet points that are well separated from each other. Hence the second measure is convergence for thresholding is important for determining a conflict threshold easily.

Speed is the last measure. Since MBC algorithm itself has a high computational complexity growing exponentially as the class size increases; it should be using a conflict detection algorithm with as much low computational load as possible that will not impose additional burden on the classification algorithm.

Based on these three measures the conflict detection algorithm described for the MBC is in the study entitled "About conflict in the theory of belief functions" by Arnaud Martin [14] was chosen. In this thesis this algorithm is going to be called Arnaud Martin's Conflict Detection Algorithm AMCA.

This chapter is made up of five sections. In the Sections 4.1 and 4.2, two essential parts of the AMCA algorithm – Jousselme Distance and Degree of Inclusion- are defined. In Section 4.3, the AMCA algorithm itself is going to be explained. How conflict thresholds are determined discussed in section 4.4 Statistical performance of the algorithm is given at Section 4.5.

4.1 Jousselme Distance

Jousselme Distance is thoroughly explained and detailed in the article entitled "A new distance between two bodies of evidence" [17]. The aim of Jousselme Distance is to determine how "far" the implied solution between two different bba is to each other.

For a frame of discernment Ω whose elements are $w_1, w_2, w_3, \ldots, w_k$ where $k > 0$ is integer; values of a mass function $m(w_l), l \leq k$ can be modeled as a discreet random variable with fixed values. With this definition vector space generated by focal elements of the mass function is denoted by $\varepsilon_{P(\Omega)}$ where $P(\Omega)$ represents all of the subsets of Ω. The formal definition of $\varepsilon_{P(\Omega)}$ is given as

$$\varepsilon_{P(\Omega)} = \sum_{i=1}^{k} \alpha_i \beta_i \qquad (4.1)$$

where $\alpha_i \in \Omega$ and β_i is one of symbols in the focal sets of the mass function

In the vector space defined in Equation (4.1) a distance calculation function within metric space is needed to measure how far two mass function values are from each other. Assume that there exist : A_1 *and* $A_2 \in P(\Omega)$ and the notation $d(A_1, A_2)$ stands for the distance between A_1 *and* A_2. Such function should be defined on $\varepsilon \; x \; \varepsilon$ and must have the following properties:

- $d(A_1, A_2) \geq 0$ (Nonnegativity) (4.2)
- *if* $A_1 = A_2$ then $d(A_1, A_2) = 0$ (Nondegeneracy) (4.3)
- $d(A_1, A_2) = d(A_2, A_1)$ (Symmetry) (4.4)
- $d(A_1, A_2) \leq d(A_1, A_3) + d(A_1, A_3)$ *where* $A_3 \in P(\Omega)$ (4.5)

With the conditions defined from Equation (4.2) to (4.5) that if function *d* is in metric space, so is $\varepsilon_{P(\Omega)}$. Therefore, the Jousselme distance formula is proposed as:

$$d(m_1, m_2) = \overrightarrow{(m_1} - \overrightarrow{m_2})^T D(\overrightarrow{m_1} - \overrightarrow{m_2}) \quad (4.6)$$

Where D mentioned is a $2^k x 2^k$ matrix defined in metric space as well. In addition, D should also be sensitive to similarities and differences between two given bbas. That is why Jaccard Distance Matrix most commonly used in the area of Computer Vision is chosen for D. In computer vision, the main purpose of Jaccard Distance Matrix is to find similarities and difference between given images. Consequently, Jaccard Distance Matrix is perfectly suitable for comparison of two bbas. The mathematical definition of Jaccard Distance Matrix is given by:

$$D(A_2, A_1) = \frac{|A_2 \cap A_1|}{|A_2 \cup A_1|} \quad (4.7)$$

where $|A_2 \cap A_1|$and $|A_2 \cup A_1|$ shows cardinality of the given set operations

When Equation (4.7) is substituted in Equation (4.6), Jousselme Distance formula for two different mass functions of $m_1$and m_2 defined on the same frame of discernment Ω is becomes.

$$d_\Omega(m_1, m_2) = \sqrt{0.5 * (m_1 - m_2)^T D(m_1 - m_2)} \quad (4.8)$$

The number resulting from Equation (4.8) is called the total conflict measurement. Measuring the total conflict through a distance-based conflict algorithm such as Jousselme Distance is quite useful. However, it does not always give the outcome of 0 for the value of $m(\Omega) = 1$ and any m value. This represents the state of total ignorance.

That is why an additional definition of conflict measure is needed that can check the inclusiveness of one mass function on another[14].

4.2 Degree of Inclusion

In the previous section, Jousselme Distance – a distance-based conflict algorithm- was explained. While Jousselme Distance is an excellent algorithm for measuring conflict between two mass functions, it is not effective for measuring conflict on empty sets and at the state of total ignorance.

Consider two mass functions denoted by m_1 and m_2. m_1 is said to be included in m_2 if all focal elements of m_1 are included in m_2. This inclusion is denoted as $m_1 \subseteq m_2$. The notation for conflict derived from this definition is $Conf(m_1,m_2)$ and it should have the following properties:

- $Conf(m_1,m_2) \geq 0$ (Non-negativity) (4.9)
- $Conf(m_1,m_2) = 0$ (Identity) (4.10)
- $Conf(m_1,m_2) = Conf(m_2,m_1)$(Symmetry) (4.11)
- $0 \leq Conf(m_1,m_2) \leq 1$(Normalization) (4.12)
- $Conf(m_1,m_2) = 0 \Leftrightarrow m_1 \subseteq m_2 \; or \; m_2 \subseteq m_1$ (Inclusion) (4.13)

Please note that compared the properties of Jousselme Distance, Inclusivity is less restrictive here since Jousselme Distance only allows nondegeneracy or identity ($d(m_1,m_2) = 0$) and not inclusivity. In addition, Pisagor Inequality is not needed. Hence conflict between two masses can be reduced through an intermediate mass.

Let A_1 and A_2 be focal elements for the mass functions m_1 and m_2, respectively. To begin with an inclusion index is defined given as Equation (4.14):

$$Inc(A_1, A_2) = \begin{cases} 1, A_1 \subseteq A_2 \\ 0, otherwise \end{cases} \qquad (4.14)$$

Then a degree of inclusion m_1 in m_2 can be defined as

$$d_{inc}(m_1,m_2) = \frac{1}{|C_1|\,|C_2|} \sum_{A_1 \in C_1} \sum_{A_2 \in C_2} Inc(A_1, A_2) \qquad (4.15)$$

Where $|C_1|$ and $|C_2|$ shows the number of focal sets in mass functions m_1 and m_2 respectively.

From Equation (4.15), a degree of inclusion of m_1 and m_2 is given by

$$\delta_{inc}(m_1,m_2) = \max\big(d_{inc}(m_1,m_2), d_{inc}(m_2,m_1)\big) \qquad (4.16)$$

Equation (4.16) is inversely proportional to the conflict measure. For example, if $m_1 \subseteq m_2$ then $\delta_{inc}(m_1,m_2) = 1$. That's why true notation of degree of inclusion is defined as

$$1 - \delta_{inc}(m_1, m_2) \quad (4.17)$$

4.3 Arnaud Martin's Conflict Detection Algorithm (AMCA)

AMCA is derived from the Jousselme Distance and the Degree of Inclusion. Its mathematical definition is given as by

$$Conf(m_1, m_2) = \left(1 - \delta_{inc}(m_1, m_2)\right) * d(m_1, m_2) \quad (4.18)$$

The first term on the right hand side of the Equation (4.18) was defined in Equation (4.17) and the second term was explained in Equation (4.8). Please note that since both terms are normalized, the Equation (4.18) itself is also normalized.

Example 4.1 illustrates how the conflict between two mass functions is determined using AMCA

<u>Example 4.1:</u>

Consider the following mass functions $m_1$and m_2 defined over a frame of discernment $\Omega = \{w_1, w_2\}$ given as:

$m_1(w_1) = 0.5 \;\; m_1(w_2) = 0.2 \;\; m_1(\Omega) = 0.3$

$m_2(w_2) = 0.3 \;\; m_2(\Omega) = 0.7$

Determine the conflict between the mass functions by using AMCA

Solution:

From Equation (4.7), Jaccard Distance is computed as:

$$D = \begin{bmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0.5 & 0 \\ 0.5 & 0.5 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrix form of m_1 is $M_1 = \begin{bmatrix} 0.5 \\ 0.2 \\ 0.3 \\ 0 \end{bmatrix}$

Similarly matrix form of m_2 is $M_2 = \begin{bmatrix} 0 \\ 0.3 \\ 0.7 \\ 0 \end{bmatrix}$

Hence $M_1 - M_2 = \begin{bmatrix} 0.5 \\ -0.1 \\ -0.4 \\ 0 \end{bmatrix}$

From Equation (4.8) Jousselme Distance is determined as $d_\Omega(m_1, m_2) = 0.361$

From Equation (4.15), $d_{inc}(m_1, m_2)$ and $d_{inc}(m_2, m_1)$ are found as:

$$d_{inc}(m_1, m_2) = \frac{1}{6}(1 + 0 + 1 + 1 + 1 + 0) = 0.67$$

$$d_{inc}(m_2, m_1) = \frac{1}{6}(0 + 0 + 1 + 0 + 1 + 1) = 0.5$$

Consequently, From equation 4.18 conflict is calculated as:

$$\delta_{inc}(m_1, m_2) = 0.67$$

$$Conf(m_1, m_2) = (1 - 0.67) * 0.361 = 0.119$$

4.4 Determination of Conflict Threshold

In previous sections, the idea behind the conflict detection algorithm used in this thesis was explained thoroughly. If the result of AMCA is zero, there is no conflict between any given two mass functions. If the result is ,1 then the given mass functions are in total conflict. However, most of the time conflict result obtained from any conflict detection algorithm including AMCA is a value between 0 and 1. Based on a conflict value that is between 0 and 1, how can one tell if the given mass functions are conflicting with each other? A threshold value is needed to answer this question. For a given threshold, conflict exists between the two mass functions if AMCA produces a number greater than the threshold. Otherwise conflict does not exist

Before explaining the method used to determine the conflict threshold an auxiliary definition that called “conflict identity” is needed. The definition of conflict identity is as follows:

Definition:

For any given mass functions pairs, conflict identity is the information of whether conflict exists between them. If an observer knows the conflict identity of a mass function pair then it knows whether mass functions are in conflict.

Determining the optimum conflict threshold is not a trivial task since there is no well-defined rule. In addition, the conflict threshold could potentially vary greatly from one problem to another even for the same conflict detection algorithm. The best method for determining the threshold is trial and error. For a given conflict detection algorithm, one can determine the threshold by looking at the frequencies of the conflict values for the mass functions that are known to conflict with each. Similarly, conflict values for the mass functions that are known not to conflict with one another can be examined. This can be done by assuming that conflict identity of mass function pairs can be modeled as a probability density functions (PDF). The aim of this modeling is to create a histogram of PDF. Then, a suitable value that can be used for making decision about conflict identity of the mass function pair can be selected from the histogram.

To form the corresponding histograms, two sets of data were taken from the artificial learning set explained in Section 3.4. Each data set contains 5000 points pairs One pair comes from the learning set $S_1$and the other is from the learning set S_2. Mass function’s conflict identity are formed by using these point pairs.

To form mass function pairs that do not conflict with each other, one of the six classes from w_1 to w_6 is randomly selected. Then, the first data point of the pair is chosen from the randomly selected class in the learning set S_1. Then we make sure that the second data point of the pair is chosen from the same class in the learning set S_2. In the final step, two mass functions are formed by using the pairs based on the algorithm explained in Section 3.2 and 3.3 and the respective conflict value is determined via AMCA. This selection algorithm mimics the scenario for which both information sources are indicating the same outcome maximizing the likelihood of correct classification. Therefore, the degree of conflict is minimal.

In order to create two mass functions that are highly conflicting with each other; one the six classes from w_1 to w_6 is again randomly selected and the first data point of the pair is taken as before. Then we make sure that the other data point of the pair is selected from a class different from the randomly selected class in S_2. Then, two mass function pairs are created the previously explained manner and their conflict is calculated through AMCA. Unlike the previous method, this selection approach ensures that correct classification cannot be achieved since the underlined mass functions are in conflict with each other.

Histograms corresponding to the conflict values obtained from the non-conflicting and conflicting mass functions are illustrated in Figures 4.1 and 4.2 respectively.

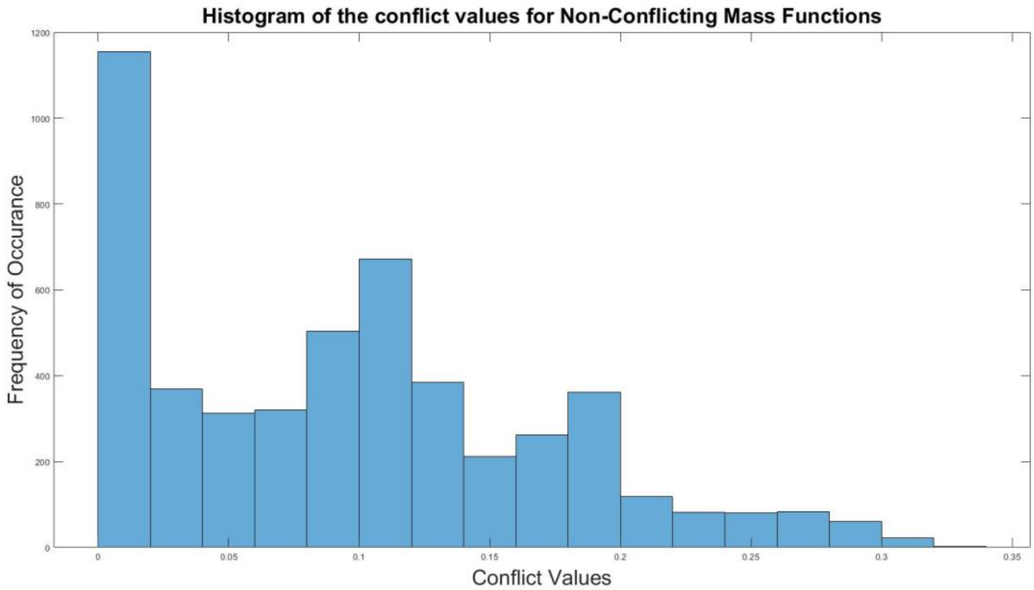

Figure 4.1 Histogram of the conflict values obtained from the Non-Conflicting Mass Function Pairs

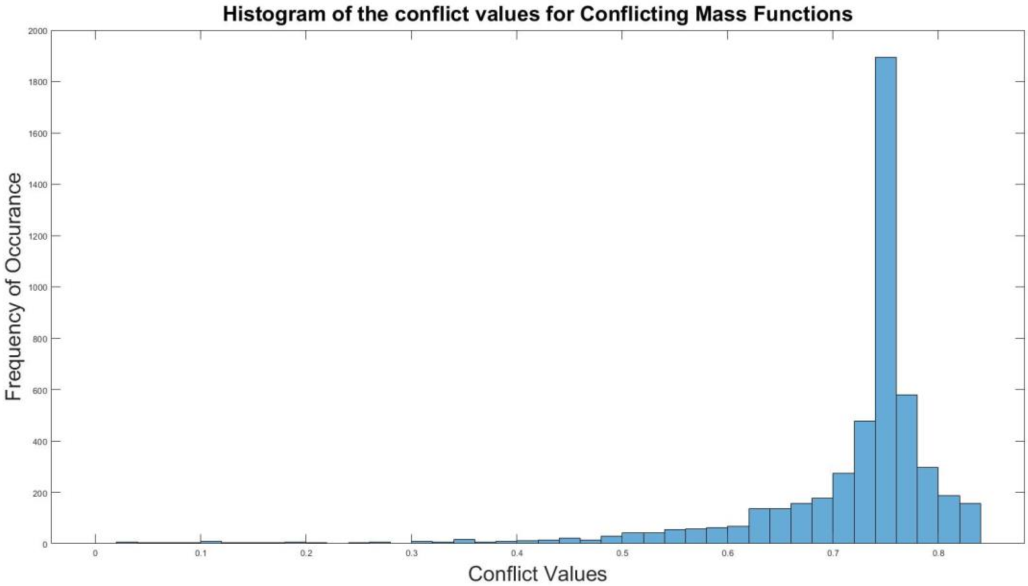

Figure 4.2 Histogram of the conflict values obtained from the Conflicting Mass Functions

Conflict thresholds are determined in the following way from the histograms. For each histogram for a given threshold more than 98 percent correct conflict identity classification should be achieved while overall correct classification should be more than 98 percent as well. Two requirements could not be satisfied when one threshold is used. For that reason, two conflict thresholds were used.

The conflict threshold determined from Figure 4.1 is found to be 0.35. In a similar manner the conflict threshold from Figure 4.2 is found to be 0.5. For a given mass function pairs if a calculated conflict value is lower than 0.35 conflict does not exist, if conflict value is higher than 0.5 conflict exists. Otherwise we cannot make a decision.

In the next section statistical performance of AMCA algorithm based on the conflict threshold values is evaluated.

4.5 Statistical Evaluation of AMCA

Statistical evaluation is going to be performed similar to what is done in Section 3.6. First statistical hypotheses have to be defined. This is required since there are values that are inconclusive between two conflict thresholds. Thus, the corresponding confusion table is different from the one in Section 3.6. After confusion table is created, precision recall and accuracy are calculated based on these values. In the final step, f-beta measurement will be given.

The statistical hypothesis that are used in this section are defined as follows:

- True Positive: Conflict exists between mass functions and the algorithm detects it successfully.
- True Negative: There is no conflict between mass functions and the algorithm decides that there is no conflict.
- False Positive: There is no conflict between mass functions. However, the algorithm decides that there is conflict.
- False Negative: There is conflict between mass functions. However, algorithm decides that there is no conflict:
- Undecided: Conflict for the given mass functions exists between two threshold values. The algorithm cannot decide whether conflict exists or not.

Table 4.1 The Confusion table of AMCA based on the determined threshold values

True Positive	4798
True Negative	5000
False Positive	0
False Negative	89
Undecided	113

For a total of 10000 data pairs, the confusion table created based on the determined threshold values is given in Table 4.1

Table 4.2 Precision Recall and Accuracy values of AMCA

Precision	1
Recall	0.981
Accuracy	0.979

Precision, recall and accuracy are determined from Table 4.1. They are specified in Table 4.2

Using values in the Table 4.2 F-measurement is calculated to be 0.990. This shows that both precision and recall values as well as F-measurement is very close to 1.

Using precision, recall and F-measurement we can conclude that AMCA algorithm works as intended with the determined threshold values it is able to detect conflict between any given mass function pair for our problem.

5 SPEED-ORİENTED OPTİMİZATİON OF MBC

During the implementation of MBC it was observed that MBC algorithm's computational speed reduce drastically as class size increases. Furthermore, majority of the slowdown results from the computation of Dempster Junction Rule. This is somewhat expected due to two main reasons. First Dempster Junction Rule requires subsets of a frame of discernment. The Dempster Junction Rule formula that was given in Section 2.2 in Equation 2.22. From which it can be clearly observed that for each increment in class size increases the computational complexity exponentially. Second, computation step requires set operations. Set operations with symbolic values are known to be computed slowly in most high-level programming languages.

When these two issues combined it causes a serious computational bottleneck for MBC. The objective of this chapter is to come up with solutions in order to speed up the computation and reduce the waiting time for Dempster Junction Rule operation to be complete.

The number of set operations was minimized by computing Dempster Junction Rule in a different way. After some research it is discovered that would be mitigated some of the computational performance loss due to increased class size by using what is called commonality function. Commonality function is one of the key concepts in the Belief Function theory. This function finds committed total mass to one subset and all of its supersets. For a frame of discernment Ω and two subsets $C_1 \subseteq C_2 \subseteq \Omega$ notation $q(C_1)$ expresses amount of total mass in C_1 committed to C_1 and all of the subsets in the superset C_2. The mathematical definition of commonality function is given as Equation (5.1)

$$q(C_1) = \sum_{C_1 \subseteq \Omega, C_1 \subseteq C_2} m(C_2), \forall\, C_1 \subseteq \Omega \qquad (5.1)$$

From a given commonality function the corresponding mass function is calculated from Equation (5.2)

$$m(C_1) = \sum_{C_1 \subseteq \Omega, C_1 \subseteq C_2} (-1)^{|C_2 \setminus C_1|}\, q(C_2), \forall\, C_1 \subseteq \Omega \qquad (5.2)$$

Dempster Junction Rule Computation becomes quite trivial if it is computed through commonality function. The computation rule in terms of commonality function is given as Equation (5.3):

$$q_{1\cap 2}(C) = q_1(C)\ x\ q_2(C), \forall\ C \subseteq \Omega \qquad (5.3)$$

As it is evident from Equation (5.3) that commonality function reduces Dempster Junction Rule to a simple product operation at a slight cost of conversion to and from mass function to commonality function.

<u>Example 5.1:</u>

Calculate Dempster Junction Rule through commonality function and convert the joint commonality back to mass function

For a frame of discernment $\Omega = \{w_1, w_2, w_3\}$ and the following two mass functions:

$m_1(w_1,w_2) = 0.5, m_1(w_2) = 0.2, m_1(w_1,w_3) = 0.2, m_1(w_1,w_2,w_3) = 0.1$

$m_2(w_3) = 0.7, m_2(w_2,w_3) = 0.3$

Solution:

First mass functions are converted commonality functions as via Equation (5.1)

$q_1(w_1) = m_1(w_1,w_2) + m_1(w_2) + m_1(w_1,w_2,w_3) = 0.8$

$q_1(w_2) = m_1(w_2) + m_1(w_1,w_2) + m_1(w_1,w_2,w_3) = 0.8$

$q_1(w_3) = m_1(w_2) + m_1(w_1,w_2,w_3) = 0.3$

$q_1(w_1,w_2) = m_1(w_1,w_2) = 0.5 + m_1(w_1,w_2,w_3) = 0.6$

$q_1(w_1,w_3) = m_1(w_2) + m_1(w_1,w_2,w_3)$=0.3

$q_1(w_2,w_3) = m_1(w_1,w_2,w_3)$=0.1

$q_1(w_1,w_2,w_3) = m_1(w_1,w_2,w_3)$=0.1

$q_2(w_1) = 0$

$q_2(w_2) = m_2(w_2, w_3)$=0.3

$q_2(w_3) = m_2(w_3) + m_2(w_2, w_3) = 1$

$q_2(w_1, w_2) = 0$

$q_2(w_1, w_3) = 0$

$q_2(w_2, w_3) = m_2(w_2, w_3)$=0.3

$q_2(w_1, w_2, w_3) = 0$

From equation 5.3 Dempster Junction Rule is computed as:

$q_{1\cap 2}(w_1) = q_3(w_1) = 0$

$q_3(w_2) = q_1(w_2)\ x\ q_2(w_2) = 0.24$

$q_3(w_3) = q_1(w_3)\ x\ q_2(w_3) = 0.3$

$q_3(w_1, w_2) = 0$

$q_3(w_1, w_3) = 0$

$q_3(w_2, w_3) = q_1(w_2, w_3)\ x\ q_2(w_2, w_3) = 0.03$

$q_3(w_1, w_2, w_3) = 0$

From equation 5.2 commonality functions can be converted back to mass functions:

$m_3(w_1) = 0$

$m_3(w_2) = q_3(w_2) + (-1)^1 q_3(w_2, w_3) = 0.21$

$m_3(w_3) = q_3(w_3) + (-1)^1 q_3(w_2, w_3) = 0.27$

$m_3(w_1, w_2) = 0$

$m_3(w_1, w_3) = 0$

$m_3(w_2, w_3) = q_3(w_2, w_3) = 0.03$

$m_3(w_1, w_2, w_3) = 0$

$m_3(\phi) = 0.49$

The second solution is to represent elements of the mass functions in a way such that set operations can be computed faster in a high-level programming language. To achieve this goal symbolic elements of a mass functions are represented as vectors. For instance, for a mass function such as $m(w_1, w_3)$ in a frame of discernment of $\Omega = \{w_1, w_2, w_3\}$ symbolic part w_1, w_3 can be represented as a vector $[1\ 0\ 1]$. which means that if an element of frame of discernment existing in the mass function represented as a 1 and otherwise as a 0. Length of the vector is equal to the number of elements in the frame of discernment. This representation will be called vector representation.

Even though vector representation makes an algorithm sensitive to input order, it increases computational performance drastically when combined with commonality functions especially in high level programming environment like MATLAB.

Please see Table 5.1 to asses contribution of these two solutions for Dempster Junction Rule computation. The table show the number for the proposed solution and classical way in MATLAB and computational times are in seconds for various class sizes

Table 5.1 Computational time in seconds of Dempster Junction Rule for different solutions

Class Size	Computational time without proposed solutions	Computational time with proposed solutions
2	1.023	2.909
3	3.515	2.468
4	14.466	2.918
5	64.534	3.938
6	297.073	6.041
7	1492.073	10.357
8	8248.474	19.597
9	48600.850	39.225
10	315876.275	87.768

Table 5.1 shows that except for the class size of 2 the proposed solution always has a performance advantage increasing as size increases. For instance, when the class size is 10 it took more than 3 days to complete the calculation with the without proposed solution

while it took less than ninety seconds to complete same calculation with the proposed solution. Similar observations can be made for the other class sizes.

6 CONCLUSION AND FUTURE IMPROVEMENTS

In this thesis Model-Based Classification (MBC) algorithm was implemented to solve automatic ship classification problem. Basic concepts of Belief Function Theory and necessary operations required for the implementation of MBC was given in Chapter 2. In chapter 3 the algorithm and theory behind the MBC is explained and statistical performance of the algorithms was given in order to validate the obtained results show that algorithm is working as intended. In addition, generation of artificial learning set is explained. In chapter 4 a conflict detection algorithm is explained, conflict threshold is determined using histograms and its statistical performance was given. In chapter 5 a speed-oriented optimization method is explained to reduce the computational load of the MBC. This study can be extended in several ways. First extension might be to add time scalability. By indexing the learning set with respect to time when feature vectors are obtained, more accurate classifications can be achieved for limited time frames. For example, assume that there exists an observation post with the intention of making an accurate cloud density and pattern prediction for a location that has wet season and arid seasons. Since cloud density from season to season considering a whole year's data might affect classification-accuracy negatively. However, if measurements are indexed based on the time they are obtained, a suitable portion of learning set can be used so that a better prediction about weather results in.

Second extension might be is to convert the MBC structure into a deep learning algorithm structure. The advantage of such a conversion is that the MBC becomes a self-updating, self-learning framework making a reliable decisions on the fly without needing any supervision after a certain amount of data is used for training the neural network. Third extension might be to use a different conjunction rule other than the Dempster-Junction Rule, so that combining mass functions whose degree of conflict is high could be achieved.

Finally, a new conjunction calculation method whose computation complexity increases linearly with respect to class size instead of exponentially could be envisioned. That would be a breakthrough achievement for since one of the major disadvantages of the MBC is its high computational cost when the class sizes are too big.

KAYNAKLAR/REFERENCES

1. Zadeh, L. A., “Fuzzy sets,” Information and Control, vol. 8, no. 3, pp.338-353, 1965.
2. Zadeh, L. A., “ Fuzzy sets as a basis for a theory of possibility,”.Fuzzy Sets and Systems, vol. 1, pp. 3-28, 1978.
3. Pawlak, Z., “Rough sets,” International Journal of Computer& Information Sciences, vol. 11, no. 5, pp.341-356, 1982.
4. Shafer, Glenn. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
5. Smets, Philippe, and Robert Kennes. “The Transferable Belief Model.” *Artificial Intelligence*, vol. 66, no. 2, 1994, pp. 191–234., doi:10.1016/0004-3702(94)90026-
6. Denuxdenoeux, Thierry, and Philippe Smets. “Classification Using Belief Functions: Relationship Between Case-Based and Model-Based Approaches.” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 6, 2006, pp. 1395–1406., doi:10.1109/tsmcb.2006.877795.
7. Smets, Philippe, and Robert Kennes. “The Transferable Belief Model.” *Artificial Intelligence*, vol. 66, no. 2, 1994, pp. 697–699., doi:10.1016/0004-3702(94)90026-4.
8. R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification. NewYork: Wiley, 2001
9. K. Fukunaga, Introduction to Statistical Pattern Recognition. Boston, MA: Academic, 1990
10. Smets, P., “Belief functions: The disjunctive rule of combination and the generalized Bayesian theorem,” Int. J. Approx. Reasoning, vol. 9, no. 1, pp. 1–35, 1993.
11. Appriou, A., “Uncertain data aggregation in classification and tracking processes,” in Aggregation and Fusion of Imperfect Information, B. Bouchon-Meunier, Ed. Heidelberg, Germany: Physica-Verlag, 1998, pp. 231–260.
12. Atalan, Sami. *Modern Deniz Sistemleri Harp Gemileri*. Türk Deniz Kuvvetleri, 2016
13. Zadeh, L. A. (1986). A Simple View of the Dempster-Shafer Theory of Evidence and Its Implication for the Rule of Combination. *AI Magazine*, *7*(2), 85. *

14. Martin, Arnaud. “About Conflict in the Theory of Belief Functions.” *Advances in Intelligent and Soft Computing Belief Functions: Theory and Applications*, 2012, pp. 161–168., doi:10.1007/978-3-642-29461-7_19.
15. Lefevre, E., et al. “Belief Function Combination and Conflict Management.” *Information Fusion*, vol. 3, no. 2, 2002, pp. 149–162., doi:10.1016/s1566-2535(02)00053-2.
16. Deng, Yong. “Generalized Evidence Theory.” *Applied Intelligence*, vol. 43, no. 3, 2015, pp. 530–543., doi:10.1007/s10489-015-0661-2.
17. Jousselme, Anne-Laure, et al. “A New Distance between Two Bodies of Evidence.” *Information Fusion*, vol. 2, no. 2, 2001, pp. 91–101., doi:10.1016/s1566-2535(01)00026-4.

ÖZGEÇMİŞ

Gürcan Durukan 2015 yılında Bilkent Elektrik Elektronik Bölümünden mezun oldu. Şu anda Marmara Üniversitesinde Elektrik Elektronik Mühendisliği dalında yüksek lisans yapmaktadır. Ana ilgi alanları örüntü tanıma sınıflandırma ve makine öğrenme olmakla beraber sayısal görüntü işleme ve video işleme alanıyla da ilgilenmektedir.