UNIVERSITE PIERRE & MARIE CURIE
LA SCIENCE A PARIS

THESIS

presented by

Marcin DETYNIECKI

for the degree

DOCTEUR de l'UNIVERSITE PARIS VI

Specialty:

Artificial Intelligence

Defended November 24, 2000
with the jury:

Director	Bernadette BOUCHON-MEUNIER
Director	Ronald YAGER
Referee	Radko MESIAR
Referee	Henri PRADE
Examiner	Nozha BOUJEMAA
Examiner	Michel GRABISCH
Examiner	Philippe JOLY

Mathematical Aggregation Operators and their Application to Video Querying

Abstract

In this manuscript we present the mathematical aggregation operators and their application to the video querying. This work is divided in three parts. The first one offers the definition of mathematical aggregation operators and some properties, followed by an extensive overview of the existing operators.

The second part is dedicated to the study of the aggregation under uncertainty. We present a deep study on t-norms and t-conorms, pursued by a study on aggregation of truth and falsity values in non-phrase calculus way. We also introduce a non-axiomatic way, based on the metaphor of a balance, which in the one hand allows the visualization of the global behavior and of the sensitivity of an operator and in the other hand offers a guide for the construction of additive generated operators.

The third part is devoted to the illustration of the theoretical results in the framework of video querying. We expound two complementary approaches. The first one based on "computing with words" explains how to browse a video with temporal queries. The second one makes obvious how to aggregate criteria pointing to the same conclusion. We prove the feasibility of the approach with real search engine and we expound the used technology (Java, XML, etc.)

Keywords :

Aggregation	Multimedia	Fuzzy logic
Data Fusion	Video	Truth and Falsity
Mathematical Operators	Query	Balance Metaphor

Résumé

Dans ce manuscrit nous présentons les opérateurs mathématiques dédiés à l'agrégation et leur application à la recherche d'information dans la vidéo. Cet ouvrage est divisée en trois parties. La première présente d'abord la définition d'un opérateur mathématique d'agrégation accompagnée de quelques propriétés souhaitables. Ensuite suit une vue d'ensemble des opérateurs existants.

La deuxième partie est dédiée à l'investigation de l'agrégation dans l'incertain. Nous présentons une étude approfondie des t-normes et t-conormes, suivie d'une étude sur l'agrégation des valeurs de vérité et de fausseté différente du calcul de type logique. Nous introduisons aussi une approche non-axiomatique, qui s'appuie sur la métaphore d'une balance. Avec ceci nous pouvons d'une part visualiser le comportement global et la sensibilité d'un opérateur et d'autre part construire des opérateurs générés additivement.

La troisième partie est dévolue à la description de la façon dont ces résultats théoriques s'intègrent dans le cadre de la recherche d'information dans la vidéo. Nous explicitons deux approches complémentaires. La première, inspirée du "computing with words" de L.A. Zadeh, présente la façon de naviguer dans une vidéo en utilisant des requêtes temporelles. La seconde montre comment agréger différents critères indiquant la même conclusion. Nous démontrons la faisabilité de notre approche en offrant un prototype de moteur de recherche et en exposant la technologie utilisée (Java, XML, etc.)

Mots clefs :

Agrégation	Multimédia	Logique Floue
Fusion de données	Vidéo	Vrai et Faux
Opérateurs Mathématiques	Requêtes	Métaphore d'une Balance

This work is dedicated
to my family

Acknowledgements

This work is the fruit of a close collaboration between the LOFTI team of the Computer science laboratory LIP6 of the University of Paris 6, the Machine Intelligence Institute in New York and the Multimedia Indexing team of the LIP6.

I would like to particularly thank Mme Bernadette Bouchon-Meunier for directing this thesis within the LOFTI team. She was undoubtedly the architect of this work. I would like to thank her for always inducing me to push further my ideas. And also, for all the opportunities she offered me.

I would like to express my gratitude to M. Ronald Yager for accepting to direct my work and for receiving me at the Machine Intelligence Institute in New York. I would like to particularly thank him for all his advises, not only the scientific ones.

I would like to express my thankfulness to M. Pascal Faudemay and M. Philippe Joly for accepting me at the Multimedia Indexing team, where I found the perfect environment for the development of an application, in particular multimedia hardware, and a friendly team.

I would like to express my gratitude to M. Radko Mesiar, with whom I discussed for hours on the mathematical aspects of the aggregation operators, while he was an invited professor in the LOFTI team.

I'm also grateful to M. Henri Prade for accepting to review closely this work and to Mme Nozha Boujemaa and M. Michel Grabisch for being part of the jury.

I think that recognition should be given to all the members of the LOFTI team, who questioned my work during my entire thesis and presented their ideas in the group meetings. So, thanks to Christophe, to Herman, to Maria, to Nedra, to Andreas and to Louis.

A particular credit should be given to all the members of the Multimedia Indexing team, who during the application stage of my work really helped me with all the technical aspects and who enlightened me about the problems relative to the computer management of the video. So, thanks to Nicolas, Gwenael, Cedric, Claude, Rosa, Jean and Phung.

I would like to thank the Direction des Affaires Internationales de l'Université de Paris 6, for their help on the legal, administrative and logistics aspects of my thesis. A particular thanks to Sandrine Besson and to Christelle Colnée. Here, I would also like to thank the team of the LIP6 Ghislaine, Anne, Jacqueline and Andre.

I would like to thank the Maison de Norvège of the Cité Universitaire Internationale de Paris, the International House in New York and the CROUS of rue des Carmes, which offered me student housing during my thesis.

I also specially thank Joëlle and François Rostas, who assisted me during all my studies and who always believed in me.

I would like to finish by thanking my family which morally supported me not only during this thesis, but also during the nine years I have been away from home doing my studies.

This work was funded by the Ministère de l'Enseignement Supérieur et de la Recherche under the form of an "allocation de recherche" attributed in the framework of the international mobility.

Table of contents

Introduction

Aggregation and fusion of information are major problems for all kinds of knowledge based systems, from image processing to decision making, from pattern recognition to machine learning. But now it is becoming urgent with the explosion of the multimedia data. If we look at the information richest multimedia channel, the video, we will witness an explosion in the amount of information, not only because of its inherent richness, but also because to its popularity due to this richness.

Video as format of computer related material is becoming more and more common. Every day new multimedia information systems appear on the market containing more and more video. Also the format of information on Internet is clearly evolving to a video form. Initially we saw the embedding of images on text pages, now we see simple animation on almost every web page. We also know that the amount of information stored in computers is growing. So, the question that naturally arises is : "How to get the information you want?" And a natural answer to this question is : "Tell us what you want!". In other words, in order to provide the user the information he is looking for, we need to be able to answer his queries. But, now the problem of aggregating all the information available in order to answer the questions emerges.

In a more general framework aggregation has for purpose the use simultaneously of pieces of information provided by several sources in order to come to a conclusion or a decision. This problem being particularly significant for the information stored numerically in the computers (and here we are not only talking about multimedia). Aggregation methods are fundamental to resume information dispersed on different descriptors.

In many cases, the available information is imperfect which means that some elements are uncertain and/or imprecise, and some of them are missing (information is then incomplete). Several methodologies are useful to manage such imperfect information. Among the most important ones are probability theory, evidence theory, possibility theory and fuzzy set theory. We focus our work on the last one, because it is directly connected to the logical aggregation, usually used in a boolean way in the querying systems. Fuzzy logic is actually the most adapted method for the numerical representation of logical uncertainties and imperfections.

In this thesis we propose to study the mathematical aggregation operators that are on the basis of every aggregation or fusion in a computer. We illustrate our theoretical results by presenting solutions for the video querying. We opt to offer a coherent and homogeneous discourse by only presenting the results directly related to the mathematical aggregation. We drop off aspects as for instance the architecture of the video search engine [76], or the investigations on the ranking methods [18], [26] and on the defuzzification techniques [27], [25].

This manuscript is divided in three parts. The first one defines and presents the wide range of mathematical aggregation operators. The second one puts forward the investigation done around the aggregation of uncertain information. And the third one, illustrates in the video query framework the results of the theoretical investigation.

The first part : The Mathematical Aggregation Operators

The first section presenting the mathematical aggregation operator is divided into two chapters.

Chapter 1 propounds a definition for mathematical aggregation operators, by fixing a minimal set of mathematical conditions. This minimal set guarantees some coherence to this family of operators. It also presents other mathematical properties that can be expected from any aggregation operator.

Chapter 2 reveals a catalogue of the existing operators. We portray their characteristics and advantages, but we try to be objective by giving also their disadvantages. In this catalogue we find operators going from the quasi-arithmetic means to the t-norms and t-conorms, going through the fuzzy integral or even the OWA operators.

Second part : Studying the Aggregation of Uncertain Information

In this second part we present our investigations and results around the aggregation of the uncertain. This section is divided into 4 chapters.

The first one, **chapter 3**, presents a study on how the different traditionally used logical aggregation operators (t-norms and t-conorms) aggregate the logical values. The idea is to focus our analysis on just one point, the "most fuzzy" one.

The results of the previous study suggested that the different t-norms and t-conorms had different attitudes with respect to the uncertainty. In particular a gap in their attitude was found. Based on this, we study in **chapter 4** a way of creating new operators by reducing t-norms and augmenting t-conorms. Unfortunately we notice that the resulting operators are in general not associative. However we discover an interesting operator, for which we exhibit its worth in the temporal querying calculus presented in chapter 7.

The prior studies on the t-norms and t-conorms pointed out that the purpose of this aggregation is to compute the truth value of a logical phrase. In **chapter 5** we are interested in the aggregation of different truth values observed for the same logical phrase. This study leads to the characterization of two families of aggregation operators, the prudent and the enthusiastic aggregation operator. Also a deep study on the contradiction of information is presented

A main result of chapter 5 is that the most suitable form of aggregation of truth and falsity values is an additive generated operator. So, in **chapter 6** we propose a new framework (philosophy) for additive generated operators. We do not base our work on an axiomatic set but on the metaphor of a balance. We used a metaphor in order to allow intuitive representation of the global behavior of the operator, the visualization of its sensitivity and the analysis of a particular aggregation. But the visualization is not the only use of the metaphor, we also present a guide of how to construct an aggregation operator, using the balance. The metaphor allowing this time to make the right choices while conceiving an aggregation operator.

Third part : The Video Querying

In this **third part** we present how the theoretical results (shown in the second part) can be implemented in order to solve the problem of aggregation of information in the video querying problematic. This part is divided into two chapters.

In **chapter 7**, we present a model that enables us to browse a video using imperfect temporal queries. We propose a fuzzy query system based on fuzzy continuous annotations. We introduced a dictionary with the basic concepts and the way to construct new ones using t-norms and t-conorms. However the choice of the operator is not always an easy task, we suggest some solution based on the previous results. Also we notice that in the case of the video the number of aggregation can easily explode, so we propose an approximation of t-norms founded in the second part.

Although, as pointed out in chapter 5, when aggregating different sources indicating the same result, we should use another kind of aggregation than t-norms and t-conorms. In **Chapter 8**, we present a Java based in-video search engine that illustrates this particular problem.

We bring to a close this work by offering some conclusions that summarize the principal results and provide some perspectives.

First Part

The mathematical Aggregation Operators

The aggregation operators are mathematical objects that have the function of reducing a set of numbers into a unique representative (or meaningful) number. We insist in the mathematical aspect of this aggregation since we are dealing with aggregation of real numbers and not fusion of information at a higher level as for instance the aggregation of rules. But it is important to notice that any aggregation or fusion process done with a computer underlies numerical aggregation. In other words, the mathematical aggregation operators are the key in this kind of processes.

Here we present a catalogue of the existing operators. We portray their characteristics and advantages, but we try to be objective by giving also their disadvantages, which is rarely done in the papers presenting the operators.

We start this first part (**chapter 1**) by adopting a minimal set of mathematical conditions that define an aggregation operator. This is essential, because any mathematical operator that transforms a set of numbers into a unique one does not necessarily give a representative or meaningful final value. More precisely this mathematical axiom set guarantees that we are not obtaining injudicious results. We proceed by presenting other mathematical properties that we can be expected from any operator. Here again we try to give an interpretation to the mathematical characteristics.

The second part (**chapter 2**) presents a catalogue of the existing operators. We portray their characteristics and advantages, but we try to be objective by giving also their disadvantages, which is rarely done in the papers presenting the operators.

We try in this first part to summarize the different points of view in order to obtain a global vision on the domain. This part will be used as the starting point for the next part.

Chapter 1

Aggregation Operators

1.1 Definition

In a rather informal way, the aggregation problem consist in aggregating *n*-tuples of objects all belonging to a given set, into a single object of the same set. In the case of mathematical aggregation operator this set is all the real numbers. In this setting, an aggregation operator is simply a function, which assigns a real number *y* to any *n*-tuple $(x_1, x_2, \dots, x_n)$ of real numbers :

$$y = Aggreg\ (x_1, x_2, \dots, x_n) \tag{1}$$

Naturally, we should impose certain conditions on *Aggreg* to justify the name of "aggregation operator". Several authors [65],[71] have proposed a set of fundamental conditions defining the aggregation operators. It is to notice that these basic definitions are not compatible.

Recently, Mesiar and Komorníková [67] proposed a set of fundamental properties that group all the precedent definitions under weaker conditions. So, we define an aggregation operator as a function $Agreg : \bigcup_{n \in N} [0,1]^n \rightarrow [0,1]$ that satisfies :

- $Aggreg\ (x) = x$ **Identity when unary**
- $Aggreg\ (0,\dots,0) = 0$ and $Aggreg\ (1,\dots,1) = 1$ **Boundary conditions**
- $Aggreg\ (x_1,\dots, x_n) \le Aggreg\ (y_1,\dots, y_n)$ **Non decreasing**
  if $(x_1,\dots, x_n) \le (y_1,\dots, y_n)$

These conditions seem to be recurrent in all other proposed definitions of an aggregation operator. All other properties may come in addition to this fundamental group. In the next section we present an overview of the properties we may expect from an aggregation operator.

1.2 Properties of an aggregation operator

We divided the set of properties into two families : the mathematical properties and the behavioral properties. For more details, see Fodor [42] and Grabisch [47].

Mathematical properties

1.2.1 Boundary Conditions

Here we turn our attention to the behavior of the aggregator in the best and in the worst cases. We expect that an aggregation operator satisfies :

$$Aggreg\ (0,0,\ldots,0) = 0 \qquad (2)$$

$$Aggreg\ (1,1,\ldots,1) = 1 \qquad (3)$$

Condition (2) means that if we observe only completely bad, false or not satisfactory criteria the total aggregation has to be also completely bad, false or not satisfactory. We have that (3) translates that if we observe only true or completely satisfactory criteria then the total aggregation has to be also completely true or satisfactory.

As Mesiar and Komorníková pointed out in [67], this property seems to be fundamental in the definition of aggregation operators.

Extension of this basic condition has been proposed. For example Mayor and Trillas [65] propose as a fundamental condition for an aggregation operator the following :

$$\forall x \in [0,1] \quad Aggreg\ (x,0) = Aggreg\ (1,0) \cdot x \qquad (4)$$

$$\forall x \in [0,1] \quad Aggreg\ (x,1) = (1 - Aggreg\ (1,0)) \cdot x + Aggreg\ (1,0) \qquad (5)$$

We notice that (4) requires that the value $Aggreg\,(x,0)$ is the weighted arithmetic mean of x and 0 ; in the same way $Aggreg\,(x,1)$ is the weighted arithmetic mean of x and 1 (see (5)). These two conditions constrain a little bit more the group of aggregation operators. In fact (2) and (3) are particular cases for x=0 and x=1 respectively for (4) and (5).

1.2.2 Monotonicity (non decreasing)

We deal more precisely with a non-decreasingness with respect to each variable. We expect that if an argument increases then the final aggregation increases (or at least not decreases, remaining equal) :

$$y_i \geq x_i \Rightarrow Aggreg(x_1,\ldots,y_i,\ldots,x_n) \geq Aggreg(x_1,\ldots,x_i,\ldots,x_n) \qquad (6)$$

Strict monotonicity = cancellativity

1.2.3 Continuity

The function *Aggreg* is continuous with respect to each of its variables. This property is a guaranty for certain robustness, for a certain consistency and for a non chaotic behavior.

1.2.4 Associativity

An interesting property is to be able to aggregate by packages. We expect that the choice of the packages has no influence on the result. For three arguments the property can be written :

$$\begin{aligned} Aggreg(x_1, x_2, x_3) &= Aggreg(Aggreg(x_1, x_2), x_3) \\ &= Aggreg(x_1, Aggreg(x_2, x_3)) \end{aligned} \quad (7)$$

This property can also be useful if the operator is defined only for two elements, then the Associativity allows extending the definition to *n* arguments without ambiguity.

1.2.5 Symmetry

Also known as commutativity or anonymity : The order of the arguments has no influence on the result. This property is compulsory when the aggregation is made of arguments having the same importance or arises from anonymous experts or sources.

For every permutation σ of {1,2, … , n } the operator satisfies :

$$Aggreg\left(x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n)}\right) = Aggreg\left(x_1, x_2, \dots, x_n\right) \quad (8)$$

1.2.6 Bisymmetry

Bisymmetry is a property associated to the aggregation of n^2 inputs for a n-ary operators. If we write these inputs in a square matrix, then the bisymmetry translates the fact that it does not matter whether you first aggregate the column vectors and then the outputs of thereof, or vice versa, first you aggregate row vectors and then the relevant outputs. For a binary operator *A* this means for all x_{11}, x_{12}, x_{21}, x_{22}, that :

$$A(A(x_{11}, x_{12}), A(x_{21}, x_{22})) = A(A(x_{11}, x_{21}), A(x_{12}, x_{22})) \quad (9)$$

Note : If an operator is commutative and associative then it is necessarily bisymmetric, however, neither commutativity nor associativity is implied by bisymmetry.

1.2.7 Absorbent Element

If the aggregation operator has an absorbent element *a*, then it can be used like an eliminating score or like a veto (it can be also considered as a qualifying score) :

$$Aggreg\left(x_1, \dots, a, \dots, x_n\right) = a \quad (10)$$

This element is also called annihilator.

1.2.8 Neutral Element

If the operator of aggregation has a neutral element *e*, then it can be used to be associated to an argument that should not have any influence on the aggregation :

$$Aggreg^{[n]}(x_1,\ldots,e,\ldots,x_{n-1}) = Aggreg^{[n-1]}(x_1,\ldots,x_{n-1}) \qquad (11)$$

1.2.9 Idempotence

Also known as unanimity or agreement : If we aggregate *n* times the same value, we expect to find the initial value :

$$Aggreg(x,x,\ldots,x) = x \qquad (12)$$

This property and the reinforcement property are incompatible.

1.2.10 Compensation

Also known as Pareto property. Here we expect that the result of the aggregation is lower than the highest element aggregated (the maximum) and higher than the lowest one (the minimum) :

$$\min_{i=1}^{n}(x_i) \leq Aggreg(x_1,x_2,\ldots,x_n) \leq \max_{i=1}^{n}(x_i) \qquad (13)$$

This property is not to be confused with the counterbalance property.

1.2.11 Counterbalancement

Also called by some authors compensation. This means that some confusion may appear with the previous property.

We call the counterbalance property, the behavior of an operator that decreases the final result if there are arguments that go into an opposite direction.

$$\forall t \in \,]0,1[,\ \forall (x_1,\ldots x_n)\ \exists (y_1,\ldots y_m) \text{ so that } Aggreg(x_1,\ldots x_n, y_1,\ldots y_m) = t \qquad (14)$$

1.2.12 Reinforcement

One characteristic of many types of human information processing, which was strikingly pointed out by Elkan [41], is what Yager and Rybalov [87] **full reinforcement**. By this property we mean to indicate the tendency, on the one hand, of a collection of high scores to reinforce each other to give a resulting score more affirmative then any of the individual scores alone and on the other hand the tendency of a collection of low scores to reinforce each other to give a resulting score more "disfirmative" than any of the individual scores. The first concept is called upward reinforcement and the second concept is called downward reinforcement. Yager shows [87] that the t-norms have only a downward reinforcement behavior, while the t-conorms have only an upward reinforcement behavior. He also shows that the uninorms have a full reinforcement behavior.

This property can be very interesting. For example, in medical diagnosis the appearance of a number of symptoms indicative of a disease will make us more confident in diagnosing a patient as having the disease than any symptoms alone while the lack of appearance of this symptoms will make us more confident diagnosing a patient as not having the disease.

1.2.13 Stability for a linear function

This property translates a stability of the operator for a change of measurement scale :

$$Aggreg\left(r \cdot x_1 + t, r \cdot x_2 + t, \ldots, r \cdot x_n + t\right) = r \cdot \left(Aggreg\left(x_1, x_2, \ldots, x_n\right)\right) + t \qquad (15)$$

A widely study particular case is the self-duality (see [78] and [29]). It corresponds to the stability for the linear function where $r = -1$ and $t = 1$.

1.2.14 Invariance

When aggregating numbers $(x_1, x_2, \ldots, x_n)$ represent measurement of certain criteria, we should specify a scale in which these measurements were performed. Moreover, we may want the aggregation function *Aggreg* to respect a meaningful relation with respect to the given scale. The notion of meaningfulness is formalized in the representational theory of measurement [62] as the invariance property :

For any admissible transformation f, we have :

$$Aggreg\left(f(x_1), f(x_2), \ldots, f(x_n)\right) = f\left(Aggreg\left(x_1, x_2, \ldots, x_n\right)\right) \qquad (16)$$

Note : the only aggregation operator totally invariant (invariant for any bijection f) is the projection.

Behavioral properties

1.2.15 Decisional behavior

It is useful to have the possibility to express the behavior of the decision-maker. For example : tolerant, optimistic, pessimistic or strict. These behaviors are in multi-criteria usually named **disjunctive** and **conjunctive** behaviors.

1.2.16 Interpretability of the parameters

It is to be hoped that the parameters have almost evident semantic interpretation. This property forbids the use of a black box methodology.

1.2.17 Weights on the arguments

It is crucial to have the possibility to express weights on the arguments. This can be understood as privileging some of them.

Chapter 2

The Mathematical Operators

In this chapter, we present an overview of the existing mathematical operators. We explain their main properties and particularities. We also present some notable particular cases.

We start by presenting some of most often used aggregation operators. We call them the basic ones. In this group we find the prototype of an aggregation operator, *the average*, but also we find the median, the minimum and the maximum, as well as some classical generalizations like the weighted mean and the k-order statistics.

We proceed the chapter by presenting the quasi-arithmetic means, a large useful family built on a transformation of average operator. Then we continue by presenting a generalization of the weighted mean, the ordered weighted average (OWA), which has also as particular case the minimum and the maximum. This leads us to the discrete fuzzy integrals : Choquet and Sugeno. The Choquet integral generalizes the OWA operators, while the Sugeno generalizes the weighted maximum and the weighted minimum operators . All these operators give a representative value "in the middle" of the aggregated set.

After the precedent, we present two families specialized on the aggregation under uncertainty : the t-norms and the t-conorms. These operators do not look for giving a "middle value", but instead they compute the intersection and union (respectively) of fuzzy sets. These operators are often used, since they can also be seen as a generalization of the logical aggregation operators : AND (t-norms) and OR (t-conorms).

Some research works reveal that human do not aggregate "logically" as the t-norms and t-conorms do. These works stressed on the fact that the operators classically used do not compensate "low" with "high" values. We present some of the proposed solutions that were based on t-norms and t-conorms : the compensatory operators.

Other kind of operators appeared when relaxing the axiom that differentiates the t-norm and t-conorm : the uninorms. These operators solve another problem of the t-norms and t-conorms, which is the lack of full (downwards and upwards) reinforcement.

We try here to present an objective overview of the domain, by presenting the characteristics, the advantages and disadvantages of each operator. A very good overview is also available in the form of a book [7] edited by Bouchon-Meunier.

2.1 Basic Operators

2.1.1 The arithmetic mean

The simplest and most common way to aggregate is to use a simple arithmetic mean (also know as the average). Mathematically we have :

$$\begin{aligned} M(x_1, x_2, ..., x_n) &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ &= \sum_{i=1}^{n}\left(\frac{1}{n}\cdot x_i\right) \end{aligned} \tag{17}$$

This operator is interesting because it gives an aggregated value that is smaller than the greatest argument and bigger than the smallest one. So, the resulting aggregation is "a middle value". This property is known as the compensation property (see section 1.2.10), and is described mathematically by :

$$\min_{i=1}^{n}(x_i) \le M(x_1, x_2, \ldots, x_n) \le \max_{i=1}^{n}(x_i) \tag{18}$$

It is often used since it is simple and satisfies the properties of monotonicity, continuity, symmetry, associativity, idempotence and stability for linear transformations. But it has neither absorbent nor neutral element and has no behavioral properties.

There exists a classical extension : **the weighted mean**, which allows placing weights on the arguments. But we loose the property of symmetry. It is expressed mathematically by :

$$M_{w_1,\ldots,w_n}(x_1, x_2, ..., x_n) = \sum_{i=1}^{n}(w_i \cdot x_i) \tag{19}$$

where the weights are non negative and $\sum_{i=1}^{n} w_i = 1$.

2.1.2 The median

Another operator that follows the idea obtaining "a middle value" is the median. It consists in ordering the arguments from the smallest one to the biggest one. And then taking the element in the middle. If the cardinality of the set of arguments is not odd then there is not a middle argument but a pair. We take then the mean of the middle pair.

This aggregation operator satisfies the boundary conditions, the monotonicity, the symmetry, the idempotence and evidently the compensation behavior.

There exists a generalization of this operator : **the k-order statistic**, with which we can choose the element on the k^{th} position on the ordered list (from the smallest to the biggest element). Recent works present even more general median-based operators, see Calvo and Mesiar [9].

2.1.3 The minimum and the maximum

Two remarkable particular cases of the k-order statistic are the minimum and the maximum. The minimum gives the smallest value of a set, while the maximum gives the greatest one. They are aggregation operators since they satisfy the axioms of the definition.

The main properties of these operators are : monotonicity, symmetry, associativity, idempotence. Mathematically speaking they have a compensation behavior, but these are the limit cases. Using these operators we will never obtain an aggregated value "in the middle". For this reason, we do not consider that we can talk about compensation behavior in this case.

If we work in a restricted interval [*a*,*b*] the minimum has for absorbent element *a* and for neutral element *b*, while for the maximum it will be the opposite : *a* will be the neutral element and *b* the absorbent one.

It is important to notice that the minimum has a conjunctive behavior and it is a particular t-norm (see below). The maximum is a t-conorm (see below) and has a disjunctive behavior.

Since the possibility of giving weights is important, Yager introduced in [92] the weighted minimum :

$$\min_{w_1,\dots,w_n}(x_1, x_2, \dots, x_n) = \min_{i=1}^{n}\left[\max\left(1 - w_i, x_i\right)\right] \qquad (20)$$

And the weighted maximum :

$$\max_{w_1,\dots,w_n}(x_1, x_2, \dots, x_n) = \max_{i=1}^{n}\left[\min\left(w_i, x_i\right)\right] \qquad (21)$$

2.2 Quasi-arithmetic means

Many extensions of the simple arithmetic mean have been introduced such as the geometric mean :

$$M_{geometric}(x_1, x_2, ..., x_n) = \left(\prod_{i=1}^{n} x_i \right)^{\frac{1}{n}} \tag{22}$$

and the harmonic mean :

$$M_{harmonic}(x_1, x_2, ..., x_n) = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}} \tag{23}$$

In fact all these common means belong to the family of the quasi-arithmetic means. This family has been studied in detail by Kolmogorov [59] and by Aczel [1], [47] and is defined as follows :

$$\begin{aligned} M_f(x_1, x_2, ..., x_n) &= f^{-1}\left[\frac{1}{n} \sum_{i=1}^{n} f(x_i) \right] \\ &= f^{-1}\left[\sum_{i=1}^{n} \left(\frac{1}{n} \cdot f(x_i) \right) \right] \end{aligned} \tag{24}$$

where f is a strictly monotone continuous function in the extended real line. We should notice that a generator f is not unique. Think for instance on the linear transformation of a generator : $f'(x) = a \cdot f(x) + b$, where $a \neq 0$.

We notice that the geometric mean (22) is the particular case of (24) with $f(x) = \log x$ and the harmonic mean (23) is the particular case of (24) with $f(x) = \frac{1}{x}$.

A particular attention should be taken in the case, when there exist arguments x_i and x_j that have for images $f(x_i) = -\infty$ and $f(x_j) = +\infty$. In this case the convention for the addition of minus infinity and plus infinity should be specified.

A very notable particular case, studied in detail by Dujmovic [39] and by Dyckhoff [40] corresponds to the function f is defined by $f : x \to x^{\alpha}$. We obtain then a quasi arithmetic mean of the form :

$$\begin{aligned} M(x_1, x_2, ..., x_n) &= \left[\frac{1}{n} \sum_{i=1}^{n} x_i^{\alpha} \right]^{\frac{1}{\alpha}} \\ &= \left[\sum_{i=1}^{n} \left(\frac{1}{n} \cdot x_i^{\alpha} \right) \right]^{\frac{1}{\alpha}} \end{aligned} \tag{25}$$

This family is particularly interesting, because it generalizes a group of common means, only by changing the value of α:

- For $\alpha = 1$ we obtain the arithmetic mean
- For $\alpha = 2$ we obtain the quadratic mean (also called the Euclidean mean)
- For $\alpha = -1$ we obtain the harmonic mean
- When α tends to $-\infty$, formula (25) tends to the maximum operator.
- When α tends to $+\infty$, formula (25) tends to the minimum operator.
- When α tends to 0, formula (25) tends to the geometric mean.

2.3 Symmetric Sum

We call a symmetric sum a continuous self-dual aggregation operator *S*. We recall the self-duality is defined by :

$$S(x_1, x_2, \ldots, x_n) = 1 - S(1 - x_1, 1 - x_2, \ldots, 1 - x_n) \quad (26)$$

This operator was studied in detail by Silvert [78]. In particular he showed that the symmetric sum of two arguments can be written under the form :

$$S(x, y) = \frac{G(x, y)}{G(x, y) + G(1 - x, 1 - y)} \quad (27)$$

where *G* is a continuous, increasing, positive function satisfying $G(0,0)=0$. It is to notice there is not a unique function *G* characterizing each symmetric sum. It is also important to notice that we use the convention $\frac{0}{0} = \frac{1}{2}$.

We remark that symmetric sums are in general not symmetric or commutative. A good example of symmetric sums is the weighted mean (19).

An interesting particular case is the additive generated aggregation :

$$S_f(x_1, x_2, \ldots, x_n) = f^{-1}\left[\sum_{i=1}^{n} f(x_i)\right] \quad (28)$$

where the generator *f* is a strictly monotone continuous function in the extended real line and satisfying :

$$f(x) + f(1 - x) = 0 \quad (29)$$

In this case, we obtain an associative symmetric sum. If the range of *f* is $[-\infty,+\infty]$, then we obtain the associativity on $[0,1]^2 \setminus \{(0,1),(1,0)\}$. In other words it is associative as long as we do not aggregate the values 0 and 1.

2.4 Ordered Weighted Averaging Operators

The Ordered Weighted Averaging Operators (OWA) were originally introduced by Yager in [95] to provide a means for aggregating scores associated with the satisfaction of multiple criteria, which unifies in one operator the conjunctive and disjunctive behavior :

$$OWA(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} w_j x_{\sigma(j)} \qquad (30)$$

where σ is a permutation that orders the elements : $x_{\sigma(1)} \le x_{\sigma(2)} \le \cdots \le x_{\sigma(n)}$. The weights are all non negative ($w_i \ge 0$) and their sum equals one ($\sum_{i=1}^{n} w_i = 1$).

This operator has been proved to be very useful, because of his versatility, and it is the object of a book edited in 1997 by Yager and Kacprzyk [86].

The OWA operators provide a parameterized family of aggregation operators, which include many of the well-known operators such as the maximum, the minimum, the k-order statistics, the median and the arithmetic mean. In order to obtain these particular operators we should simply choose particular weights (see Table 1).

	OWA
Minimum	$\begin{cases} w_1 = 1 \\ w_i = 0 \quad \text{if } i \neq 1 \end{cases}$
Maximum	$\begin{cases} w_n = 1 \\ w_i = 0 \quad \text{if } i \neq n \end{cases}$
Median	$\begin{cases} w_{\frac{n+1}{2}} = 1 & \text{if } n \text{ is odd} \\ w_{\frac{n}{2}} = \frac{1}{2} \text{ and } w_{\frac{n}{2}+1} = \frac{1}{2} & \text{if } n \text{ is even} \\ w_i = 0 & \text{else.} \end{cases}$
k-order statistics	$\begin{cases} w_k = 1 \\ w_i = 0 \quad \text{if } i \neq k \end{cases}$
Arithmetic mean	$w_i = \frac{1}{n} \quad \text{for } \forall i$

Table 1. Particular cases of OWA

The Ordered Weighted Averaging operators are commutative, monotone, idempotent, they are stable for positive linear transformations and they have a compensatory behavior. This last property translates the fact that the aggregation done by an OWA operator always is between the maximum and the minimum. Since this operator generalizes the minimum and the maximum, it can be seen as a parameterized way to go from the *min* to the *max*. In this context Yager introduced in [95] a degree of maxness (initially called orness), defined by :

$$maxness(w_1, w_2, \ldots, w_n) = \sum_{j=1}^{n} w_{n-j+1} \cdot \frac{n-j}{n-1} = \sum_{j=1}^{n} w_j \cdot \frac{j-1}{n-1} \qquad (31)$$

We see that for the minimum, we have that *maxness*(1,0,…,0)=0 and for the maximum *maxness*(0, …,0,1)=1.

Another operator was introduced by Yager [95] and used by O'Hagan in [70] to characterize a particular OWA. This degree describes the dispersion of the weights and it is based on the idea of entropy :

$$dispersion(w_1, w_2, \ldots, w_n) = -\sum_{j=1}^{n} w_j \cdot \ln(w_j) \qquad (32)$$

One issue of considerable interest related to the use of these operators is the development of an appropriate methodology for the derivation of the weights used in the OWA aggregation. Two of the main approaches used are the following :

The first one introduced by O'Hagan [70] makes use of the measure of maxness and the measure of dispersion. In this approach we only required that the user provides a value $\alpha \in [0,1]$ corresponding to the degree of maxness suited. The idea is to maximize the dispersion of the weights under the constraint of a fixed maxness. The following mathematical programming problem computes the weights, for a given α :

Maximize $-\sum_{j=1}^{n} w_j \cdot \ln(w_j)$ (dispersion)
Under the constraints : - $maxness(w_1, w_2, \ldots, w_n) = \sum_{j=1}^{n} w_j \cdot \frac{j-1}{n-1} = \alpha$ - $\sum_{j=1}^{n} w_j = 1$ - $0 \le w_j \le 1$

Table 2. Mathematical program that computes OWA weights

The second approach makes use of the knowledge of a linguistic quantifier to guide the aggregation [99]. We are interested in regular increasing monotone quantifiers defined by :

- $Q(0) = 0$ and $Q(1) = 1$.
- if $x \le y$ then $Q(x) \le Q(y)$.

These quantifiers translate notions like *most, almost all, many, at least half* and *some*.

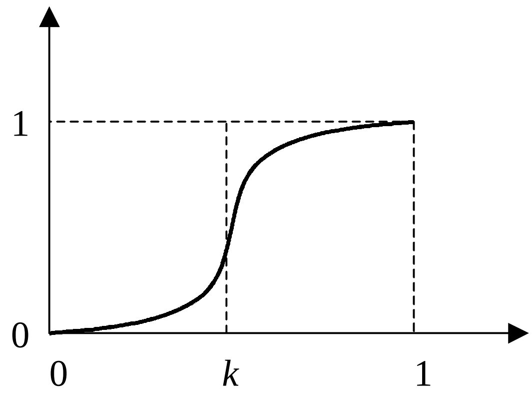


Figure 1. Regular increasing monotone quantifier "at least *k* %"

On the basis of this kind of quantifiers Yager in [95] proposed to compute the weights using the formula :

$$w_j = Q\left(\frac{n-j+1}{n}\right) - Q\left(\frac{n-j}{n}\right) \qquad (33)$$

Using this approach we can define goal functions under the constraint :

Q criteria should be satisfied

To illustrate this approach, let us consider one limit case. For instance if we want that "at least 100%" of the criteria to be satisfied, then we observe that the OWA operators is the *minimum.* And when the minimum is satisfied all other criteria are satisfied.

2.5 Choquet & Sugeno discrete Fuzzy Integrals

2.5.1 Definitions

The fuzzy integral with respect to a fuzzy measure has mainly been studied in a multicriteria decision making framework (see [47] and [45]). It is based on the notion of a fuzzy measure, which can be viewed as the weight of importance of a set. Mathematically we define the **fuzzy measure** as follows :

Let us denote by $C = \{c_1,\ldots, c_n\}$ the set of criteria, and $P(C)$ the power set of C, i.e. the set of all subsets of C. A **fuzzy measure** on C is a set function $\mu : P(C) \rightarrow [0,1]$, satisfying the following axioms.

- $\mu(\emptyset) = 0$ and $\mu(C) = 1$. Boundary conditions
- for $A,B \in P(C)$, if $A \subset B$ then $\mu(A) \leq \mu(B)$. Monotonicity

This kind of measure is more flexible than a probability, which is constrained by its additivity property. In fact, the importance of two criteria in the probability framework cannot be anything else than the sum of the individual importances, while the fuzzy measures can provide a greater (superadditive measure) or a lower (subadditive measure) value. This allows us to model interaction between criteria.

Now, using a fuzzy measure we introduce fuzzy integrals :

The discrete Sugeno integral [80] of scores $x_1,\ldots, x_n$ for criteria $c_1,\ldots, c_n$ with respect to a fuzzy measure μ, is defined by :

$$Sugeno_\mu(x_1, x_2, \ldots, x_n) = \max_{i=1}^{n}\left(\min\left(x_{\sigma(i)}, \mu(C_{\sigma(i)})\right)\right) \qquad (34)$$

where σ is a permutation that orders the elements : $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \cdots \leq x_{\sigma(n)}$, where and $C_{\sigma(i)}=\{c_{\sigma(i)},\ldots, c_{\sigma(n)}\}$.

The discrete Choquet integral [12] of scores $x_1,\ldots, x_n$ for criteria $c_1,\ldots, c_n$ with respect to a fuzzy measure μ, is defined by :

$$Choquet_\mu(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n}(x_{\sigma(i)} - x_{\sigma(i-1)}) \cdot \mu(C_{\sigma(i)}) \qquad (35)$$

with the same notation as above, and $x_{\sigma(0)} = 0$.

An equivalent expression of (35) is

$$Choquet_\mu(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_{\sigma(i)} \cdot (\mu(C_{\sigma(i)}) - \mu(C_{\sigma(i+1)})) \qquad (36)$$

with $C_{\sigma(n+1)} = \emptyset$.

2.5.2 Properties

Sugeno and Choquet integrals are interesting since they are monotone, continuous, idempotent operators, with a compensation behavior. The Choquet integral is stable under positive linear transformation, while the Sugeno integral is stable under a similar transformation with minimum and maximum replacing the product and the sum respectively. This last property points out that the Sugeno integral is more suitable for ordinal aggregation (where only the order of the elements is important) while the Choquet integral is suitable for cardinal aggregation (where the distance between the numbers has a meaning).

The commutativity is only obtained when the fuzzy measure just depends on the cardinality of the sets, i.e. $\mu(A) = \mu(B)$ if card(A) = card(B). The associativity is usually not satisfied.

The generalization capability of the Choquet and Sugeno integrals is remarkable. Both contain the order statistics and in particular the minimum and the maximum. The Choquet integral generalizes the weighted means and the OWA operator, while the Sugeno integral generalizes the weighted minimum and the weighted maximum. In the following table we present the corresponding measures in order to get a particular operator (for details see [45],[47],[80]).

	Sugeno integral
Minimum	$\begin{cases} \mu(A) = 1 & \text{if } A = C \\ \mu(A) = 0 & \text{otherwise} \end{cases}$
Maximum	$\begin{cases} \mu(A) = 0 & \text{if } A = \varnothing \\ \mu(A) = 1 & \text{otherwise} \end{cases}$
k-order statistics	$\begin{cases} \mu(A) = 0 & \text{if } card(A) \le n - k \\ \mu(A) = 1 & \text{otherwise} \end{cases}$
Weighted minimum	$\mu(A) = 1 - \max_{x_i \notin A}\left[\mu(\{x_i\})\right]$ and $\mu(\{x_i\}) = w_i$ for all i
Weighted maximum	$\mu(A) = \max_{x_i \in A}\left[\mu(\{x_i\})\right]$ and $\mu(\{x_i\}) = w_i$ for all i

Table 3. Particular cases of the Sugeno Integral

	Choquet integral
Minimum	$\begin{cases} \mu(A) = 1 & \text{if } A = C \\ \mu(A) = 0 & \text{otherwise} \end{cases}$
Maximum	$\begin{cases} \mu(A) = 0 & \text{if } A = \emptyset \\ \mu(A) = 1 & \text{otherwise} \end{cases}$
k-order statistics	$\begin{cases} \mu(A) = 0 & \text{if } card(A) \leq n - k \\ \mu(A) = 1 & \text{otherwise} \end{cases}$
Arithmetic mean	$\mu(A) = \dfrac{card(A)}{card(C)}$
Weighted mean	$\mu(A) = \sum_{x_i \in A} \mu(\{x_i\})$ and $\mu(\{x_i\}) = w_i$ for all i
OWA	$\mu(A) = \sum_{j=0}^{card(A)-1} w_{n-j}$

Table 4. Particular cases of the Choquet Integral

The main problem in the use of Choquet or Sugeno integral (besides the fact that they are not associative and commutative) is the number 2^n of weights to be provided, for a simple n criteria aggregation. These weights being nothing else than the characterization of the fuzzy measure. A main aspect of the actual research is based on the methods to determine or reduce the number of these weights. Some solutions have been proposed :

A first approach is to work on the measure, by defining (for instance) decomposable measures. An interesting approach was proposed by Grabisch in [46], where he suggests to use k-additive fuzzy measures. The idea is to define measures that are multilinear of degree k, i.e. if $card(A) > k$ then $m(A) = 0$. This approach allows to model the strength of small coalitions and reduces the number of weights to $\sum_{i=1}^{k} C_n^i$ instead of 2^n. The case of 2-additive measure has been pointed out as particularly interesting. The number of weights to establish is $\frac{n^2 + n}{2}$, the measure being defined by :

$$\mu(A) = \sum_{\{x_i, x_j\} \subset A} \mu(x_i, x_j) - (card(A) - 2) \cdot \sum_{\{x_i\} \in A} \mu(x_i) \qquad (37)$$

Another approach is to determine the weights by learning on examples. The data being a set of *n*-dimensional vectors : $(x_1^k, x_2^k, \ldots, x_n^k)$ and their corresponding aggregated values : y_k. Then we identify the fuzzy measure by minimizing the following error :

$$E^2 = \sum_{k=1}^{l} \left(Choquet_\mu (x_1^k, x_2^k, \ldots, x_n^k) - y_k \right)^2 \qquad (38)$$

It can be shown [47] that (38) can be put under a quadratic program form, that is

minimize $\frac{1}{2} \cdot u^t Du + c^t u$
under the constraint $Au + b \geq 0$

Table 5. Quadratic program computing the weights for a Choquet Integral

where *u* is the vector containing all the weights of the fuzzy measure.
It appears as an interesting solution, but if there is not enough data the matrices may be ill conditioned. In fact there must be at least $\frac{n!}{[(n/2)!]^2}$ training vectors.

2.6 Fusion Operators

This section gives an overview of the group of specific operators dealing with fusion. The authors of this operators were particularly aware of the problem of the reliability of the sources.

2.6.1 The Bayesian Approach

The idea in the Bayesian approach is to estimate the most probable "x" knowing that we observed "x_1" from source 1 and "x_2" from source 2. Mathematically we are looking for the maximum of the a posteriori probability $P(x|x_1, x_2)$.

The a posteriori probability can be derived using the Bayes theorem and knowing the value of the a priori probability $P(x_1, x_2|x)$:

$$P(x|x_1, x_2) = \frac{P(x_1, x_2|x) \cdot P(x)}{P(x_1, x_2)} \qquad (39)$$

If we assume that the sources are independent, we obtain the most usual Bayesian fusion formula :

$$P(x|x_1, x_2) = \frac{P(x_1|x) \cdot P(x_2|x) \cdot P(x)}{P(x_1, x_2)} \qquad (40)$$

Note that the a priori probabilities $P(x_1|x)$ and $P(x_2|x)$ can be seen as the credibility of the source (expert). They actually translate the probability of source 1 (and source 2) to observe x_1 (and x_2), given the "real value" x.

The main default of this method is that the a priori probabilities are not easy to obtain. Also some critics come from the independence assumption. Some solutions are reviewed in [14].

2.6.2 Possibilistic approaches

The first idea is to modify directly the information provided by the source given its reliability. Let α be the degree of certainty that a given source is reliable, then Yager [92] (also Prade [72]) proposes to modify the possibility distribution π provided by the source, using the operator :

$$\pi' = \max(\pi, 1 - \alpha) \qquad (41)$$

When α=1 (fully reliable source), π'=π, and when α=0 (absolutely unreliable source), then $\forall x \;\; \pi'(x) = 1$ (total ignorance). Note that 0 does not mean that the source lies, but that it is impossible to known whether its advice is good or not.

There exist other proposals for certainty qualification that behave similarly in the limit cases (α=1 and α=0). Yager suggested an expression of the form :

$$\pi' = \pi * \alpha + 1 - \alpha \tag{42}$$

where $*$ stands for minimum, product or linear product (max(0,a+b-1).

Another point of view is to consider that the reliability of the sources imply priorities in the aggregation. The idea of a prioritized fusion is to accept the conjunctive merging of information from a reliable source 1 and source 2 as long as the data coming from the second is consistent with the former. In case of inconsistency, the information given by the less reliable is simply discarded. If π_1 is obtained from source 1 and π_2 from source 2, the degree of consistency of π_1 and π_2 is defined by :

$$h(\pi_1, \pi_2) = \sup_x \left(\min(\pi_1(x), \pi_2(\text{x}))\right) \tag{43}$$

and the following **prioritized conjunction** has been proposed by Dubois and Prade [37] (see also Yager [94]) :

$$\pi_{1\wedge 2} = \min(\pi_1, \max(\pi_2, 1 - h(\pi_1, \pi_2))) \tag{44}$$

Note that when $h(\pi_1, \pi_2)$ =0, source 1 contradicts source 2 and the only opinion of source 1 is retained (i.e. π_1), while if $h(\pi_1, \pi_2)$ =1 we have the minimum, which is a conjunction.

The **disjunctive** counterpart of this **prioritized** operator has been proposed by Dubois and Prade [38] :

$$\pi_{1\vee 2} = \max(\pi_1, \min(\pi_2, h(\pi_1, \pi_2))) \tag{45}$$

The effect of this operator is to truncate the information supplied by the less prioritary source, while disjunctively combined with source 1. Again if the two sources disagree ($h(\pi_1, \pi_2)$ =0) then we have π_1; if $h(\pi_1, \pi_2)$ =1 then we have the maximum of π_1 and π_2 .

A very complete overview of the possibilistic fusion is offered in [36].

2.7 T-norms and t-conorms

The concept of a triangular norm was introduced by Menger [66] in order to generalize the triangular inequality of a metric. The current notion of a t-norm and its dual operator (t-conorm) is due to Schweizer and Sklar [75] [74]. Both of these operations can also be used as a generalization of the Boolean logic connectives to multi-valued logic. The t-norms generalize the conjunctive 'AND' operator and the t-conorms generalize the disjunctive 'OR' operator. This situation allows them to be used to define the intersection and union operations in fuzzy logic. This possibility was first noted by Hohle [51]. Klement [56], Dubois and Prade [30] and Alsina, Trillas, and Valverde [4] very early appreciated the possibilities of this generalization. Bonissone [6] investigated the properties of these operators with the goal of using them in the development of intelligent systems. t-norm and t-conorms have been well-studied and very good overviews and classifications of these operators can be found in the literature, see [57],[31]. A particular complete work is presented in a book [54] explicitly dedicated to these operators.

2.7.1 Definitions

t-norm : A t-norm is a function $T : [0,1]x[0,1] \to [0,1]$, having the following properties

- $T(x,y) = T(y,x)$ (T1) **Commutativity**
- $T(x,y) \le T(u,v)$, if $x \le u$ and $y \le v$ (T2) **Monotonicity (increasing)**
- $T(x,T(y,z)) = T(T(x,y),z)$ (T3) **Associativity**
- $T(x,1) = x$ (T4) **One as a neutral element**

A well known property of t-norms is :

- $T(x,y) \le \min(x,y)$ (46)

Proof : Using the monotonicity (T2) and axiom T4, we have $T(x,y) \le T(x,1) = x$ and using the commutativity we have $T(x,y) \le T(1,y) = y$. So, $T(x,y) \le \min(x,y)$.

t-conorm : Formally, a t-conorm is a function $S : [0,1]x[0,1] \to [0,1]$, having the following properties :

- $S(x,y) = S(y,x)$ (S1) **Commutativity**
- $S(x,y) \le S(u,v)$, if $x \le u$ and $y \le v$ (S2) **Monotonicity (increasing)**
- $S(x,S(y,z)) = S(S(x,y),z)$ (S3) **Associativity**
- $S(x,0) = x$ (S4) **Zero as a neutral element**

A well known property of t-conorms is :

- $S(x,y) \ge \text{Max}(x,y)$ (47)

It is actually a consequence of axioms (S1, S2, S4).

2.7.2 Dual t-norms and t-conorms

We say that a t-norm and a t-conorm are dual (or associated) if they satisfy the DeMorgan law.

- $\overline{T(x,y)} = S(\bar{x}, \bar{y})$ (the DeMorgan law)

where the line over the expression means the negation of the expression. We will use the most typical negation defined by :

- $\bar{x} = 1 - x$ (negation)

2.7.3 Examples

The definitions of t-norms and t-conorms are always given for only two elements, since these operators are by definition associative and in this case the generalization to *n* elements is trivial. The more common t-norms and their dual t-conorms are :

	t-norm	**t-conorm**
Min-Max	$\min(x,y)$	$\max(x,y)$
Probabilistic	$x \cdot y$	$x + y - x \cdot y$
Lukasiewicz	$\max(x + y - 1, 0)$	$\min(x + y, 1)$
Drastic	$\begin{cases} x & \text{if } y = 1 \\ y & \text{if } x = 1 \\ 0 & \text{anywhere else} \end{cases}$	$\begin{cases} x & \text{if } y = 0 \\ y & \text{if } x = 0 \\ 1 & \text{anywhere else} \end{cases}$

Table 6. Common t-norms and their dual t-conorms

We would like to insist here in some important particularities of these t-norms and t-conorms :

The minimum is the biggest t-norm (i.e. when using the *min*, we obtain a higher value than when using any other t-norm). It is also the only idempotent t-norm. Its dual is also idempotent and it is the smallest t-conorm.

The probabilistic case has the nice property to be "smooth". This can be translated mathematically through a continuous derivative.

The Lukasiewicz t-norm satisfies the classical logical law of non-contradiction (i.e. $T(x, \bar{x}) = 0$). And its dual the Lukasiewicz t-conorm satisfies the classical logical law of the excluded middle (i.e. $S(x, \bar{x}) = 1$).

The drastic case is interesting only from the mathematical point of view. These operators give the values 0 or 1 except when they are forced by the axioms. They are not continuous. The main interesting aspect is that the drastic t-norm is the smallest t-norm and the drastic t-conorm is the biggest t-conorm.

A main result is that we can construct any continuous t-norm by using the precedent t-norms and the equivalent result exists for the continuous t-conorms.

2.7.4 Parameterized t-norms and t-conorms

We have also parameterized t-norms and t-conorms. As special cases we obtain some of the precedent t-norms and t-conorms :

	t-norm	**t-conorm**
Hamacher ($\gamma \geq 0$)	$\dfrac{x \cdot y}{\gamma + (1-\gamma) \cdot (x + y - x \cdot y)}$	$\dfrac{x + y - x \cdot y - (1-\gamma) \cdot x \cdot y}{1 - (1-\gamma) \cdot x \cdot y}$
Yager (p > 0)	$\max\left(1 - \left[(1-x)^p + (1-y)^p\right]^{1/p} \ ,\ 0\right)$	$\min\left(\left[x^p + y^p\right]^{1/p} \ ,\ 1\right)$
Weber-Sugeno (λ_T, λ_S >-1)	$\max\left(\dfrac{x + y - 1 + \lambda_T \cdot x \cdot y}{1 + \lambda_T} \ ,\ 0\right)$	$\min\left(x + y + \lambda_S \cdot x \cdot y \ ,\ 1\right)$
Schweizer & Sklar (q>0)	$1 - \left[(1-x)^q + (1-y)^q - (1-x)^q(1-y)^q\right]^{\frac{1}{q}}$	$\left[x^q + y^q - x^q y^q\right]^{\frac{1}{q}}$
Frank (s > 0, $s \neq 1$)	$\log_s\left[1 + \dfrac{(s^x - 1) \cdot (s^y - 1)}{s - 1}\right]$	$1 - \log_s\left[1 + \dfrac{(s^{1-x} - 1) \cdot (s^{1-y} - 1)}{s - 1}\right]$

Table 7. Parameterized t-norms and t-conorms.

It is to notice that the precedent t-norms and t-conorms are dual, besides in the Weber-Sugeno case. In this last case, the duality is satisfied if the parameters satisfy $\lambda_S = \dfrac{\lambda_T}{1 + \lambda_T}$.

2.7.5 The Archimedean t-norms and t-conorms

A t-norm T is called Archimedean if for each $(x,y) \in]0,1[^2$ there is an number n so that :

$$T(\underbrace{x, \ldots, x}_{n-\text{times}}) < y \tag{48}$$

The subset of continuous Archimedean t-norms (and t-conorms) is particularly interesting because they can be represented by means of a single function that we will call the additive generator. It can be shown that for every continuous Archimedean t-norm T, there exists a continuous decreasing function f such that :

$$T(x_1,\ldots,x_n) = f^{(-1)}\left(\sum_{i=1}^{n} f(x_i)\right) \qquad (49)$$

with $f:[0,1] \to [0,+\infty]$ satisfying $f(1)=0$ and $f^{(-1)}$ is the pseudo inverse of f, defined by :

$$f^{(-1)}(z) = \begin{cases} f^{-1}(z) & \text{if } z \in [0, f(0)] \\ 0 & \text{if } z \in \left]f(0),+\infty\right] \end{cases} \qquad (50)$$

An equivalent theorem exists for the t-conorms.

In Table 7 we present the most common continuous Archimedean t-norms and t-conorms and the corresponding additive generators. We present two simple t-norms and t-conorms and two parameterized families. For a complete overview see [57].

		Usual representation	**Additive generator:** $f(u)$
Probabilistic	t-norm	$x \cdot y$	$-\ln(u)$
	t-conorm	$x+y-x\cdot y$	$-\ln(1-u)$
Lukasiewicz	t-norm	$\max(x+y-1,0)$	$1-u$
	t-conorm	$\min(x+y,1)$	u
Hamacher	t-norm	$\dfrac{x\cdot y}{\gamma+(1-\gamma)\cdot(x+y-x\cdot y)}$	for $\gamma>0$: $-\dfrac{1}{\gamma}\cdot\ln\left(\dfrac{u}{\gamma+(1-\gamma)\cdot u}\right)$
			for $\gamma=0$: $\dfrac{1-u}{u}$
	t-conorm	$\dfrac{x+y-x\cdot y-(1-\gamma)\cdot x\cdot y}{1-(1-\gamma)\cdot x\cdot y}$	for $\gamma>0$: $-\dfrac{1}{\gamma}\cdot\ln\left(\dfrac{1-u}{\gamma+(1-\gamma)\cdot(1-u)}\right)$
			for $\gamma=0$: $\dfrac{u}{1-u}$
Yager	t-norm	$\max\left(1-\left[(1-x)^p+(1-y)^p\right]^{1/p}\ ,\ 0\right)$	$(1-u)^p$
	t-conorm	$\min\left(\left[x^p+y^p\right]^{1/p}\ ,\ 1\right)$	u^p

Table 8. T-norms and t-conorms and their additive generators.

Note : the minimum and the maximum are not Archimedean, but they can be limit cases of Archimedean parameterized cases.

2.8 Compensatory Operators

Several authors noticed that t-norms and t-conorms lack of compensation behavior and that this particular property seems crucial in the aggregation process. One of the first authors to notice this were Zimmermann and Zysno [101]. They discover that in a decision making context humans do not follow exactly the behavior of a t-norm (nor of a t-conorm) when aggregating. In order to get closer to the human aggregation process, they proposed an operator on the unit interval based on t-norms and t-conorms :

$$Z_\gamma(x_1,\ldots,x_n)=\left(\prod_{i=1}^{n} x_i\right)^{1-\gamma}\cdot\left(1-\prod_{i=1}^{n}(1-x_i)\right)^{\gamma} \qquad (51)$$

Here the parameter γ indicates the degree of compensation. This operator is a particular case of the **exponential compensatory operators** [83] :

$$E_\gamma^{T,S}(x_1,\ldots,x_n)=\left(T(x_1,\ldots,x_n)\right)^{1-\gamma}\cdot\left(S(x_1,\ldots,x_n)\right)^{\gamma} \qquad (52)$$

where T is a t-norm and S a t-conorm.

It is important to notice that the exponential compensatory operators are not associative for γ different from 0 or 1.

Another class of non-associative t-norm and t-conorm-based compensatory operator is the **convex-linear compensatory operator** [83], [63] :

$$L_\gamma^{T,S}(x_1,\ldots,x_n)=(1-\gamma)\cdot T(x_1,\ldots,x_n)+\gamma\cdot S(x_1,\ldots,x_n) \qquad (53)$$

Setting the value of the parameter γ is a delicate issue. Zimmerman and Zysno calculated the best γ to match the human behavior. Yager and Rybalov proposed in [87] and [93] a method based on fuzzy modeling techniques to compute the parameter γ :

$$\gamma=\frac{T(x_1,\ldots,x_n)}{T(x_1,\ldots,x_n)+T(1-x_1,\ldots,1-x_n)} \qquad (54)$$

where $T(x_1,\ldots,x_n)$ is called the highness and $T(1-x_1,\ldots,1-x_n)$ the lowness.

Another approach to the construction of compensatory operators based on t-norms and t-conorms was taken by Klement, Mesiar and Pap [55]. They based their construction on the additive generators of continuous Archimedean t-norms and t-conorms. Their **associative compensatory operator** is defined by :

$$C(x,y)=f^{-1}(f(x)+f(y)) \qquad (55)$$

where the function f is defined by :

$$f(x) = \begin{cases} -g\left(\dfrac{x}{e}\right) & if \quad x \le e \\ h\left(\dfrac{x-e}{1-e}\right) & if \quad x \ge e \end{cases} \tag{56}$$

where g is an additive generator of a t-norm, h is an additive generator of a t-conorm and e is a neutral element. It is to notice that this operator is a particular case of uninorms (see next section 2.9).

2.9 Uninorms

T-norms and t-conorms play a notable role in fuzzy logic theory, unfortunately these operators do not admit a compensating behavior. In fact t-norms do not allow low values to be compensated by high values and t-conorms do not allow high values to be compensated by low values (see [87]). For this reason Fodor, Yager and Rybalov introduced in [43] (see also [91]) the family of uniform aggregation operators (uninorm), as a generalization of both t-norm and t-conorm. This operator has a neutral element laying anywhere in the unit interval rather than at one or zero as for the t-norms and t-conorms respectively.

2.9.1 Definition

Formally, a uninorm is a function U: [0,1]x[0,1] $\rightarrow$ [0,1], having the following properties :

- $U(x,y) = U(y,x)$ (U1) **Commutativity**
- $U(x,y) \leq U(u,v)$, if $x \leq u$ and $y \leq v$ (U2) **Monotonicity (increasing)**
- $U(x,U(y,z)) = U(U(x,y),z)$ (U3) **Associativity**
- $\exists e \in [0,1] \quad \forall x \in [0,1] \; U(x,e) = x$ (U4) ***e* is the neutral element**

We see that the first three properties (U1, U2, U3) are common to uninorms, t-norms and t-conorms, but the fourth condition U4 is more general in the case of uninorms, in that it allows any value for the identity. These properties seem to be interesting for aggregation purposes. In fact, the commutativity translates the property of an operator to give the same result independently of the order of the values to be aggregated. The monotonicity translates the fact that if one of the aggregated element augments its value, then the aggregated value at least does not decrease. The associativity imposes to the operator the property of accepting the aggregation by groups. With an associative operator we can aggregate by groups and then aggregate all the groups and obtain the same result as when aggregating all the elements directly. The associativity is also interesting when aggregating new information, in that case we would not need to re-compute the aggregation with all the arguments, but simply aggregate the old calculated value with the new one. Finally, the neutral element is interesting, because it can be considered as the score that we would give to an argument, which should not have any influence in the aggregation. It is somehow a null vote.

2.9.2 Other Properties

One characteristic of many types of human information processing, which was pointed out by Elkan [41], is what we shall call **full reinforcement**. Yager shows in [87] that the uninorms have a full reinforcement behavior, if the neutral element e is different to zero or one. In fact, a uninorm having e=1 as a neutral element is a t-norm, and a t-conorm for e=0.

If we now take a look more precisely at the uninorms we discover that a uninorm behaves as a t-norm in the square $[0,e]^2$ and as a t-conorm on the square $[e,1]^2$. In fact, De Baets and Fodor showed in [16] that to any uninorm with neutral element $e \in]0,1[$, there corresponds a t-norm T and a t-conorm S such that :

$$\forall\ (x,y) \in [0,e]^2 \qquad U(x,y) = e \cdot T\left(\frac{x}{e},\frac{y}{e}\right) \tag{57}$$

$$\forall\ (x,y) \in [e,1]^2 \qquad U(x,y) = e + (1-e) \cdot S\left(\frac{x-e}{1-e},\frac{y-e}{1-e}\right) \tag{58}$$

Another interesting property is the **compensation behavior**. Neither the t-norms nor the t-conorms present a compensation behavior. De Baets and Fodor showed in [16] that on $[0,e[\times]e,1] \cup]e,1] \times [0,e[$ any uninorm U satisfies :

$$\min(x,y) \leq U(x,y) \leq \max(x,y), \tag{59}$$

In other words uninorms show partially a compensating behavior. This property is particularly interesting when we think about the fact that if we use a t-norm the occurrence of an input small positive value would mean that the result is small, no matter what the other inputs are. They can for instance all equal one. A uninorm will allow to compensate these low values with other high values. We notice that as for the t-norms, the t-conorms do not have a compensating behavior. The appearance of a high value (i.e. close to one) will not be compensated using a t-conorm.

2.9.3 Minimal and maximal uninorms

We first note that for e = 1 or e = 0, there exists a large class of such uninorms (corresponding to the t-norms and t-conorms respectively). However, for the purpose of finding full reinforcement operators we need uninorms with neutral element other than 1 or 0. In [43] Fodor, Yager and Rybalov introduced two general classes of uninorms for any e. We will call them the minimal uninorms and the maximal uninorms.

The **minimal uninorm** is the weakest uninorm U given a t-norm T, a t-conorm S and a neutral element e. This operator will be defined by :

$$U_{\min}(x,y) = \begin{cases} e \cdot T\left(\frac{x}{e},\frac{y}{e}\right) & \text{for } x \leq e \text{ and } y \leq e \\ e + (1-e) \cdot S\left(\frac{x-e}{1-e},\frac{y-e}{1-e}\right) & \text{for } x \geq e \text{ and } y \geq e \\ \min(x,y) & \text{elsewhere} \end{cases} \tag{60}$$

The **maximal uninorm** is the strongest uninorm U given a t-norm T, a t-conorm S and a neutral element e. This operator will be defined by :

$$U_{\max}(x,y)=\begin{cases} e\cdot T\left(\frac{x}{e},\frac{y}{e}\right) & for\ x\le e\ and\ y\le e \\ e+(1-e)\cdot S\left(\frac{x-e}{1-e},\frac{y-e}{1-e}\right) & for\ x\ge e\ and\ y\ge e \\ \max(x,y) & elsewhere \end{cases} \tag{61}$$

We observe that the two families satisfy all the properties announced before, besides the compensation behavior, because we have a min or a max operator elsewhere. Another great disadvantage of these uninorms is that they have discontinuities around the neutral element. In fact for the maximal uninorm for any $a < e$, we have the following :

when we approach with values smaller than the neutral element, we have :

$$U_{\max}\left(a, x\to e^-\right)=e\cdot T\left(\frac{a}{e},\frac{x}{e}\to 1\right)=e\cdot\frac{a}{e}=a \tag{62}$$

and when we approach with values greater than the neutral element we have :

$$U_{\max}\left(a, x\to e^+\right)=\max\left(a, x\to e^+\right)=e \tag{63}$$

Comparing (62) and (63) we see that the value of the uninorm springs from $a<e$ to e, when we go from a little bit smaller than e to a little bit bigger than e.

In an analogous way, we have for the conjunctive uninorm for any $a > e$:

when we approach with values greater than the neutral element :

$$U_{\min}\left(a, x\to e^+\right)=e+(1-e)\cdot S\left(\frac{a-e}{1-e},\frac{x-e}{1-e}\to 0\right)=e+(1-e)\cdot\frac{a-e}{1-e}=a \tag{64}$$

and when we approach with values smaller than the neutral element we have :

$$U_{\min}\left(a, x\to e^-\right)=\min\left(a, x\to e^-\right)=e \tag{65}$$

Comparing (64) and (65) we see that the value of the uninorm springs from $a > e$ to e, when we go from a little bit bigger than e to a little bit smaller than e.

2.9.4 Generated uninorms

In [43] Fodor, Yager and Rybalov showed the following additive generated representation theorem, which gives an almost continuous uninorm :

Suppose U is continuous on $[0,1]^2 \setminus \{(0,1),(1,0)\}$ with neutral element $e \in]0,1[$. Then there exists a strictly increasing continuous function $g : [0,1] \to [-\infty,+\infty]$, with $g(e)=0$ such that the representation

$$U(x,y) = g^{(-1)}\left(g(x) + g(y)\right) \tag{66}$$

holds if and only if the following two conditions are satisfied :

- U is strictly increasing on the open unit square.
- U is self-dual with respect to a strong negation N with fixed point e.

In this case $g(0) = -\infty$, $g(1) = +\infty$ and $g^{(-1)} = g^{-1}$.

We remark that generated uninorms were already introduced as an interesting class of aggregation operators by Klement, Mesiar and Pap in [54], and were called the **associative compensatory operator** (see compensatory operators in section 2.8). Also Dombi [29] arrived to the same construction when presenting his **aggregative operator**.

2.8.5 Nullnorms

Nullnorms were found as solutions of the Frank equation for uninorms [11] :

$$U(x,y) + N(x,y) = x + y \tag{67}$$

From this follows that a nullnorm N is a commutative, associative and increasing operator, with an absorbent element $a \in [0,1]$ and that satisfies $\forall x \in [0,a] \quad N(x,0) = x$ and $\forall x \in [a,1] \quad N(x,1) = x$.

It can be shown that a nullnorm can be written under the following form :

$$N(x,y) = \begin{cases} a \cdot S\left(\dfrac{x}{a}, \dfrac{y}{a}\right) & \text{for } x \le a \text{ and } y \le a \\ a + (1-a) \cdot T\left(\dfrac{x-a}{1-a}, \dfrac{y-a}{1-a}\right) & \text{for } x \ge a \text{ and } y \ge a \\ a \quad \text{elsewhere} & \end{cases} \tag{68}$$

From this it is clear that this class contains t-norms (for a=0) and t-conorms (for a=1) as special cases.

Second Part

Studying the Aggregation of Uncertain Information

In this second part we present the new developments we propose for the aggregation operators. We base our work on the operators presented in the first part.

We start in **chapter 3** by studying how the different t-norms and t-conorms aggregate the uncertain. The idea is to focus our analysis on just one point, the "most fuzzy" one. This approach gives as a first result a classification method and as a second one, an interesting way of normalization of the parameters of the parameterized t-norms and t-conorm families. This new parameterization giving not only a meaning to the parameters, but also a common scale of work.

The results of the previous study suggested that the different t-norms and t-conorms have different attitudes with respect to the uncertainty. These attitudes are a more or less relaxed aggregation of the arguments (i.e. a higher or lower result for the same "to be aggregated" set). Based on this and on the observation that several aggregation operators were created by some simple mathematical operation, we study in **chapter 4** a way of creating new operators by reducing t-norms and augmenting t-conorms. Unfortunately we notice that the resulting operators are not in general t-norms (or t-conorms), because they were not associative. We finish the chapter by showing the example of the non-associative "Yager t-norm"-like operator, which illustrates the fact that these kind of constructions maybe interesting.

The prior studies on t-norms and t-conorms pointed out that the purpose of this aggregation is to compute the truth value of a logical phrase. In **chapter 5** we are interested in the aggregation of different truth values observed for the same logical phrase. We propose an axiom set for the aggregation of truth values, which leads to the characterization of two truth-aggregation families, a prudent and an enthusiastic. The first one has a cautious attitude choosing between two observed values the one which is more uncertain. The second one has an enthusiastic behavior and will reinforce the result if twice the truth or twice the falsity is observed. When observing falsity and truth the operator gives a compensated value. We finish the chapter by expounding the use of these operators and their relationship with the traditionally used truth-aggregation operators : the t-norms and t-conorms.

A main result of aggregation of truth and falsity values, presented in chapter 5 is the following : on the one hand we have a prudent aggregation that tends to the ignorance. So, if we repeat the aggregation several times, we are going quickly to finish with "total" ignorance at the end. On the other hand we have an enthusiastic aggregation operator that is in general not continuous. We proposed an almost continuous solution based in an additive generated operator. So, in **chapter 6** we propose a new framework (philosophy) for additive generated operators. This time we do not base our work on an axiomatic set but on the metaphor of a balance. We use a metaphor in order to allow intuitive representation (visualization) of the operator (section 6.4). Pushing further the established analogy, we also discover the notion of the sensitivity of an operator.

But the visualization is not the only use of the metaphor. In the second part of chapter 6 (section 6.5), we present a guide of how to construct an aggregation operator, using the balance. The metaphor allowing this time to make the right choices while conceiving an aggregation operator.

It is remarkable that the balance not only generalizes most of the aggregation operators presented in chapter 2, but also that it offers a larger framework that allows the interpretation not mathematical friendly properties, as for instance a non continuous additive generator. The balance framework also opens new perspectives for further research, in particular on new kind of weight association.

Chapter 3

Aggregating Uncertainty by means of t-norms and t-conorms

One goal of *fuzzy logic* is to extend the classical binary logic. When we talk about "fuzzy", we think about something between falsity and truth. If we denote the value truth by one and the value of false by zero, what is then more fuzzy, more central than 1/2? In fact, one of the first studies on multi-valued logic was done by Lukasiewicz, who originally worked with 3 truth values : 0, 1/2 and 1.

Important components of logic are the logical operators like negation, conjunction, disjunction and implication. In extending the binary logic to fuzzy logic an interesting and central question concerns the behavior of these logical operators at this middle point of truth-value. In this chapter we investigate this question.

T-norms and t-conorms were introduced to define the intersection and the union (respectively) of fuzzy sets. But these operations can also be used as a generalization of the boolean logic connectives. T-norms generalize the conjunctive 'AND' operator and t-conorms generalize the disjunctive 'OR' operator. We start here on the results presented in section 2.7. In this chapter we concentrate our attention in these two particular aggregation operators.

In order to study the t-norms at the "most fuzzy" point we will study the t-norms on (1/2,1/2). We show, taking into account the definitional constraints, how central this point is. We also indicate that defining a t-norm on this point can be a natural step after fulfilling the classical logic constraints. These results push us to suggest that t-norms can be classified observing their image on the (1/2,1/2) point. We consider some usual t-norms. We pursuit our study by observing what happens in the case of parameterized t-norms. We consider three different families.

Finally based on the presented classification method, we propose a uniform method for computing the parameters. This method allows not only to have the same parameter scale for all the families, but it also gives an intuitional sense to the parameters. The parameter is the expected value for the most uncertain aggregation.

3.1 The middle point for a t-norms

Let us here focus on the t-norms, but keeping in mind that analogous observations could be made for the t-conorms based on the duality between these operators.

We recall (see section 2.7.1) that formally, a t-norm $T : [0,1] \times [0,1] \to [0,1]$, which is a continuous, monotone, associative and has the unity as neutral element.

A first deduction that can be made from this is :

$$T(a,0) = 0 \qquad (69)$$

Proof : We know that $T(a,0) \in [0,1]$, so $0 \le T(a,0)$. And using axiom T2 (monotonicity) and T4 we have $T(a,0) \le T(1,0) = 0$, since $a \le 1$.

Another (well-known) property associated with this operator is $T(a,b) \le \min(a,b)$, and so, $T(a,a) \le a$. Viewed as a logical connective, 'AND' operator, the t-norm has the general tendency of making truth decrease.

Using the commutativity property, we have the boundary properties :

- $T(a,1) = a$ (B1) = axiom T4
- $T(1,a) = a$ (B2)
- $T(a,0) = 0$ (B3)
- $T(0,a) = 0$ (B4)

In other words the t-norms are completely defined on the edges of the unit square as shown in Figure 2.

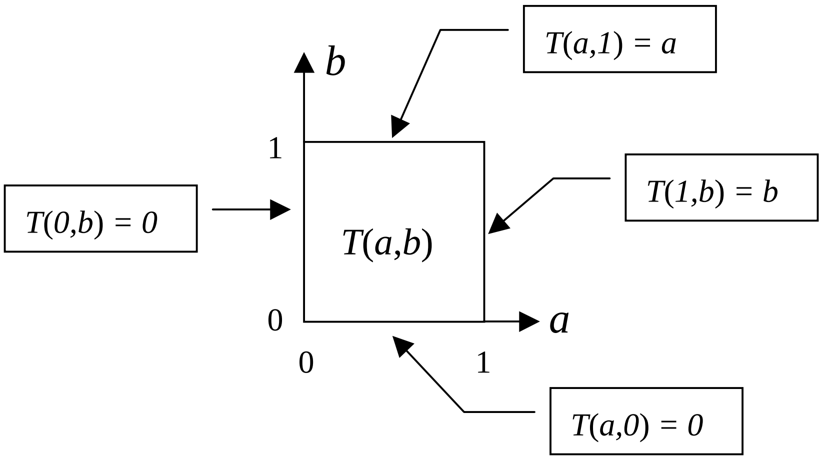


Figure 2 : Definition of t-norms on edges of the unit square

We notice that the vertices of the unit square correspond to the arguments of the classical binary logic and here the t-norm emulates the classical 'AND' operator :

- $T(1,1) = 1$ is the value of "True AND True" = "True"
- $T(1,0) = 0$ is the value of "True AND False" = "False"
- $T(0,1) = 0$ is the value of "False AND True" = "False"
- $T(0,0) = 0$ is the value of "False AND False" = "False"

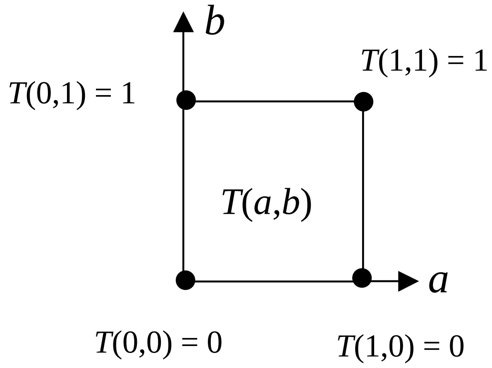


Figure 3 : Definition of t-norms on the vertices of the unit square

3.1.1 The middle point

As noted the t-norms are constrained not only to follow the classical behavior at vertices of the unit square, but also to satisfy the limit properties (B1-B4), on the edges of the unit square. We observe that in the middle area of the unit square we have the freedom of choice. It is in this area that we distinguish between different t-norm operators. An interesting point in this middle area, because of its central position is the point (1/2,1/2). We know that it is the gravitational center of the square, it is also the intersection of the diagonals, the intersection of the middle lines and the barycenter of the edges of the square (the classical points). It can also be shown that it is the point of the unit square that is at the maximal distance from the points of the border. In other words it is the "most distant" point of the already defined points and therefore it should be a useful point to characterize t-norm operators.

We can also attach a sense to this point. The value 1/2 is exactly the point between False (0 point) and True (1 point). So studying the image of the point (1/2, 1/2) we are examining the behavior of the t-norm on the more fuzzy point. More particularly if a is a truth value then $Fuz(a) = 1-|a-(1-a)|$ measures the degree of fuzziness of the value a. The degree of fuzziness for a pair of points (a,b) can be measured as the average of the two, $Fuz(a) = 1/2\ (Fuz(a) + Fuz(b))$. Thus on (1/2, 1/2) we are dealing with the most confused situation. It appears that specifying the value of a t-norm would be very useful to characterize a t-norm.

3.1.2 Usual t-norms

Let us now take a look at what the image of (1/2, 1/2) for some of the most typical t-norms (see section 2.7.3). First we note that while we have some freedom in selecting $T(1/2, 1/2)$ we have some restriction. In particular we note that $T(1/2, 1/2) \leq T(1, 1/2) = 1/2$ and $T(1/2, 1/2) \geq T(0, 1/2) = 0$, thus $T(1/2, 1/2) \in [0,1/2]$. We see that the common t-norms have different values for the image of (1/2, 1/2). The most common t-norm is the minimum t-norm defined by :

$$T_{\text{minimum}}(a,b) = \min(a,b) \tag{70}$$

We observe of course that all the conditions (B1-B4) are fulfilled. The value of this t-norm at the point (1/2, 1/2) is 1/2. This operator takes the most uncertain truth-value. Actually this is the largest possible value we can get at the point (1/2, 1/2).

Consider now the probabilistic t-norm, defined by :

$$T_{\text{probabilistic}}(a,b) = a \cdot b \tag{71}$$

The value of this t-norm at the point (1/2, 1/2) is 1/4.

Another interesting case is the Lukasiewicz t-norm, defined by :

$$T_{\text{Lukasiewicz}}(a,b) = \max(a+b-1,0) \tag{72}$$

Here the middle value is 0. And of course the limit points are fulfilled.

It would be nice if we could uniquely define a t-norm by giving its middle value. But things are not so easy. In fact we can have quite different t-norms for the same $T(1/2,1/2)$ value. A very good example is the Lukasiewicz t-norm (72) and the drastic t-norm defined by :

$$T_{\text{drastic}}(a,b) = \begin{cases} a \text{ if } \; b = 1 \\ b \text{ if } \; a = 1 \\ 0 \;\text{ anywhere else} \end{cases} \tag{73}$$

In both cases the middle point value equals 0.

As indicated above $T(1/2,1/2)$ is a value of the [0,1/2] interval. And this interval cannot be reduced, since we have shown an example for each extreme.

While we have shown that in general it is not possible to uniquely specify a t-norm by indicating its value at the middle point, for some classes of t-norms the specification of the middle can be used to uniquely identify a t-norm. In agreement with this goal we shall look at some parameterized t-norms.

3.2 Parameterized t-norms

Considerable interest in the literature on t-norms has focused on the study of parameterized families of t-norms. Klir and Folger [41] provide a comprehensive list of families of parameterized t-norms. We are going to study here just three families : the Hamacher, the Weber-Sugeno and the Yager t-norms.

3.2.1 Hamacher

The Hamacher t-norms [48] are defined for $\gamma \geq 0$ by :

$$T_{\text{Hamacher}}(a,b) = \frac{a \cdot b}{\gamma + (1-\gamma)\cdot(a+b-a\cdot b)} \tag{74}$$

This equation is reduced in the middle point to :

$$T_{\text{Hamacher}}(1/2,1/2) = \frac{1}{4\cdot\gamma + 3\cdot(1-\gamma)} = \frac{1}{\gamma+3} \tag{75}$$

If we plot the value of T(1/2,1/2) we obtain the following graph :

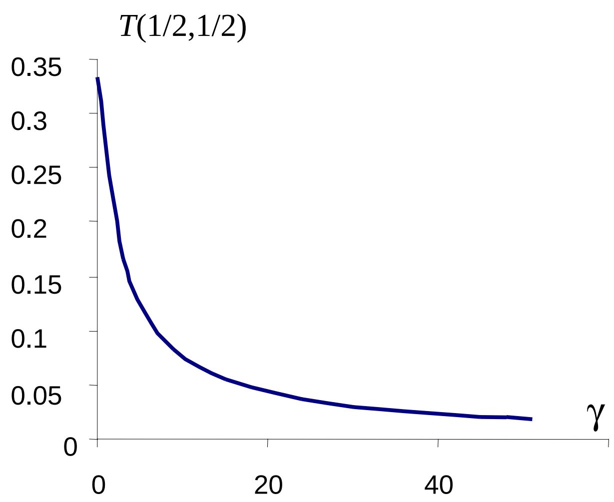


Figure 4 : T (1/2,1/2) for different parameter values of the Hamacher t-norm

We see that T(1/2,1/2) varies between 0 and 1/3. So we can only obtain t-norms with the middle point in this domain. We see that it is a bijection in this region. It means that for each γ we have a unique value of T(1/2,1/2) and vice-versa for each value of T(1/2,1/2) we can find a γ. The formula that allows us to obtain γ from the middle point value is :

$$\gamma = \frac{1}{T(1/2,1/2)} - 3 \tag{76}$$

We see that this formula allows us to obtain a γ by giving the value of $T(1/2,1/2) \in [0,1/3]$. We can now study interesting particular cases :

- $T(1/2,1/2) = 1/4$. With (76) we obtain γ = 1 and replacing this in (74) we obtain the probabilistic t-norm (71).
- $T(1/2,1/2) \to 0^+$. With (76) we obtain $\gamma \to +\infty$ and replacing this in (74) we obtain that the t-norm tends to the drastic t-norm (73).
- $T(1/2,1/2) = 1/2$. We observed in the graph that for γ > 0, $T(1/2,1/2) > 1/3$ is impossible. So we can immediately conclude that Hamacher t-norm cannot generalize even approach the minimum t-norm (70).
- $T(1/2,1/2) = 1/3$. This implies that γ = 0 and we obtain that Hamacher t-norm equals the quotient of the probabilistic t-norm and the probabilistic t-conorm.

3.2.2 Weber-Sugeno

The Weber-Sugeno t-norms [85] are defined for λ > -1 by :

$$T_{\text{Weber}}(a,b) = \max\left(\frac{a+b-1+\lambda \cdot a \cdot b}{1+\lambda}\ ,\ 0\right) \tag{77}$$

This equation is reduced in the middle point to :

$$T_{\text{Weber}}(1/2,1/2) = \max\left(\frac{\lambda}{4\cdot(1+\lambda)}\ ,\ 0\right) \tag{78}$$

If we plot the value of $T(1/2,1/2)$ we obtain the following graph :

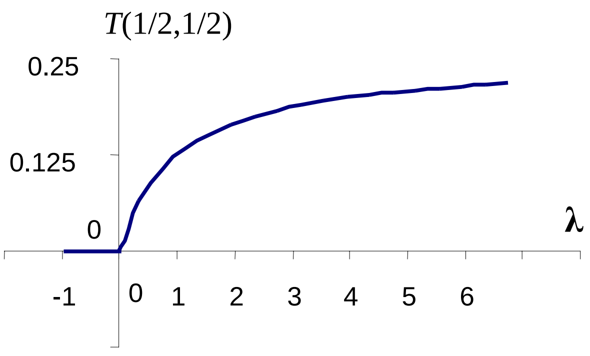


Figure 5 : *T* (1/2,1/2) for different parameter values of the Weber-Sugeno t-norm

We obtain that $T(1/2,1/2)$ lies between 0 and 1/4. We observe that for $\lambda \in]-1,0]$, $T(1/2,1/2)$ is constant and equals zero. For $\lambda \geq 0$, we have a bijection. It means that for each λ positive we have a different value of $T(1/2,1/2)$ and vice-versa for each value of $T(1/2,1/2) \in]0,1/4[$ we can find a λ. For $\lambda \in [-1,0]$, we note that $T(1/2,1/2)=0$, in other words we have an infinity of Weber-Sugeno t-norms having the middle point value equal to zero. The formula that allows us to obtain $\lambda \geq 0$ from the middle point value $t = T(1/2,1/2) \in]0,1/4[$ is :

$$\lambda = \frac{4 \cdot t}{1 - 4 \cdot t} \tag{79}$$

We see that this formula allows us to obtain a λ by giving the value of $T(1/2,1/2) \in]0,1/4[$. We can now note interesting particular cases :

- $T(1/2,1/2) = 1/2$. We observed in the graph that for $\lambda > -1$, $T(1/2,1/2) < 1/4$. So we can immediately conclude that Weber-Sugeno t-norm cannot generalize (even approach) the Zadeh t-norm (70).
- $T(1/2,1/2) \to 1/4$. With (79) we obtain $\lambda \to \infty$, we obtain that (77) tends to the probabilistic t-norm (71).
- $T(1/2,1/2) = 0$. With (79) we obtain $\lambda = 0$ and replacing this in (77) we obtain the Lukasiewicz t-norm (72).

We notice for $T(1/2,1/2) = 0$ we obtain the Lukasiewicz t-norm using the formula (79) to compute λ. But since we do not have a bijection for $-1 < \lambda \leq 0$ we cannot obtain using this formula, the t-norms with $\lambda \in]-1,0]$. In particular we cannot obtain the drastic t-norm, that is the particular case of (77) for $\lambda \to -1^+$.

3.2.3 Yager

The Yager t-norms [89] are defined for $p > 0$ by :

$$T_{\text{Yager}}(a,b) = \max\left(1 - \left[(1-a)^p + (1-b)^p\right]^{1/p} \ , \ 0\right) \tag{80}$$

This equation is reduced in the middle point to :

$$T_{\text{Yager}}(1/2,1/2) = \max\left(1 - 2^{\frac{1-p}{p}} \ , \ 0\right) \tag{81}$$

If we plot the value of $T(1/2,1/2)$ we obtain the following graph :

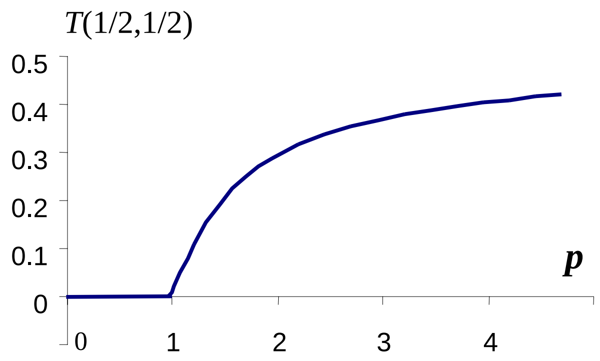


Figure 6 : *T* (1/2,1/2) for different parameter values of the Yager t-norm

We see that *T*(1/2,1/2) lies between 0 and 1/2. We observe that for $p\in\,]0,1]$, *T*(1/2,1/2) is constant and equals zero, in other words we have an infinity of Yager t-norms having the middle point value equal to zero. For $p\geq 1$, we have a bijection. This means that for each $p\geq 1$ we have a different value of *T*(1/2,1/2) and vice-versa for each value of $T(1/2,1/2)\in\,]0,1/2[$ we can find a distinct value of *p*. The formula that allows us to obtain $p\geq 1$ from the middle point value $t = T(1/2, 1/2) \in\,]0,1/2]$ is :

$$p = \frac{1}{1+\ln_2(1-t)} \qquad (82)$$

We see that this formula allows us to obtain a *p* by giving the value of *T*(1/2,1/2) $\in [0,1/2[$. We now note interesting particular cases :

- $T(1/2,1/2) \to 1/2$. With (82) we obtain $p \to \infty$ and replacing this in (80) we obtain that the t-norm tends to the minimum t-norm (70).
- *T*(1/2,1/2) = 1/4 . With (82) we obtain *p* =1.709 and replacing this in (80) we obtain a t-norm that is not the probabilistic t-norm (71), but is the closest one in the Yager family, in the sense that it equals the product on the boundary (B1-B4) and in (1/2,1/2).
- *T*(1/2,1/2) = 0. With (82) we obtain *p* = 1 and replacing this in (80) we obtain the Lukasiewicz t-norm (72).

We notice for *t* = 0 we obtain the Lukasiewicz t-norm using formula (82) to compute *p*. But since we do not have a bijection for $0< p \leq 1$ we cannot obtain using this formula the t-norms with $p\in[0,1[$. In particular we cannot obtain the drastic t-norm, that is the particular case of (80) for $p\to 0$.

3.3 The inverting functions

In the precedent section we introduced for each of the considered parameterized t-norms an inverting function : for Hamacher t-norm it was formula (76), for Weber-Sugeno t-norm it was formula (79) and for Yager t-norm it was (82).

t-norm family	Inverting function
Hamacher t-norm	$\gamma = \frac{1}{t} - 3$
Weber-Sugeno t-norm	$\lambda = \frac{4 \cdot t}{1 - 4 \cdot t}$
Yager t-norm	$p = \frac{1}{1 + \ln_2(1-t)}$

Table 9. Inverting function for parameterized families.

These inverting functions allow us to obtain the parameter required so that the parameterized t-norm has a particular $t = T(1/2,1/2)$ value. In other words if we impose the value of $T(1/2,1/2)$, using this formulas we can obtain the parameter that allows the t-norm to have this particular value.

From another perspective we can consider that the value of $T(1/2,1/2)$ is the parameter, through the inverting formulas. This perspective has the advantage of having a uniform parameter for all the parameterized t-norms :

- In all cases the parameter t satisfies $0 \leq t \leq 1/2$. So, we work always on the same scale, which is not the case when using directly the parameters.
- We observe that using this parameterization we will always evolve from the most drastic t-norm for $t = 0$ to the closest to the 'min' operator for $t = 1/2$. So, we have always the same kind of variation when the parameter t increases or decreases. This is not the case using directly the parameters, for instance we could compare Hamacher and Yager families.
- We also notice that in order to obtain for example the probabilistic t-norm the parameter t will have to equal 1/4, for any parameterized family. So, we have necessarily to use the same parameter for all parameterized t-norm families in order to obtain a certain particular case. For the minimum t-norm $t = 1/2$ and for the Lukasiewicz and the drastic t-norms $t = 0$.

Notes : The fact that $T(1/2,1/2)$ has a particular value does not guarantee that we generalize a behavior. For example, with this method we can obtain a Yager t-norm satisfying $T(1/2,1/2)=1/4$. We observed that it is not the probabilistic t-norm, but another t-norm of the class $t = 1/4$. But we can say that we have "the closest" Yager t-norm to the product, in the sense that they are equal on all the edges of the unit square and also in the middle point.

The value of $T(1/2,1/2)$ says only in which class of t-norm we are. But it does not say anything about which t-norm in this class we are generalizing. For example for T=0 we can have the drastic t-norm or the Lukasiewicz t-norm.

Another important point that we would like to point out here is that $T(1/2,1/2)$ in function of the parameters is usually only a bijection on a part of the [0,1/2] interval. The consequence of this is that the inverting function does not cover all the values of the parameter. In other words there are particular t-norms that are generalized by the usual definition of the parameterized t-norm family, that cannot be obtained using the inverting function. An example of that is that Yager t-norm generalizes t-norms that are more drastic than the Lukasiewicz t-norm (for $0<p<1$), but we cannot obtain these t-norms using the inverting formula (82) for $t \in [0,1/2[$. In a more general way we can say that the group of t-norms stricter than the Lukasiewicz t-norm are indistinguishable for our method, since all of these operators have all $T(1/2,1/2)=0$. But it seems in the literature that these operators are not often very used, besides the drastic t-norm that has a mathematical interest.

3.4 The middle point as a classification method

In the previous section we saw that the inverting function offers a uniform method for computing the parameter of each family on a parameter scale. Since we have a common and intuitive scale for all the t-norms we can compare and classify them. This classification translates their attitude with respect to the aggregation of 1/2 and 1/2. Here we draw in a same graph (Figure 7) the behavior of some usual t-norms (see section 2.7).

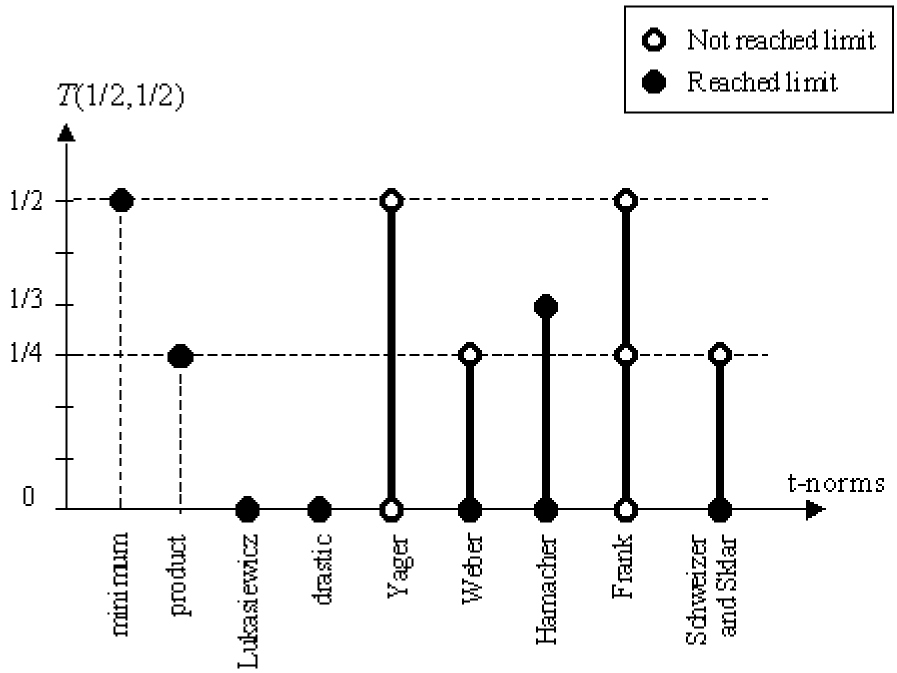


Figure 7. Classification of some usual t-norms

We can immediately conclude when looking at Figure 7 that the usual parameterized t-norms (see section 2.7.4) have a very strict attitude. Usually the aggregation of (1/2,1/2) is 1/4 or less. We can be disappointed, since a parameterized t-norm should primarily aim to cover the range [0,1/2], giving a large choice of attitudes. Instead the usual t-norms take care of the generalization of t-norms placed somewhere between the Lukasiewicz one and the drastic one, all having very similar attitudes.

Two particular families should be pointed out : the Frank family and Yager's one. Both cover the range]0,1/2[, the boundaries being not reached limit cases. The main difference is at $T(1/2,1/2) = 1/4$, the Frank t-norm has as a non reached limit case the product, while Yager t-norm presents another t-norm for this value (see section 3.2.3). This makes the Yager family more comfortable for application, but as shown at the end of the next chapter (see section 4.7) the computational effort is relatively high.

Chapter 4

Reducing t-norms and augmenting t-conorms

In the last chapter we noticed that t-norms have different attitudes with respect to the aggregation of the "most fuzzy point". In particular we remarked that in general the attitude of the parameterized families is very severe (usually the generalized t-norms are stronger than the product). In other words there are not so many parameterized families offering particular cases for the most interesting range [0,1/2], and specially on the upper half of this range. We are going to use this observation as a starting point.

We also remarked that the minimum operator is the biggest possible t-norm. Taking into account these two facts we propose to reduce this operator in order to obtain more drastic t-norms, having naturally a range starting at 1/2 and then being reduced until 0. Unfortunately, we discover that when reducing the t-norms, we usually lose the associativity. However, the resulting operators are similar to t-norm aggregation operators. So, we investigate the reduction of some typical t-norms. We also study the dual operator of the reduced t-norm, which can be interpreted as an augmented t-conorm, and again some particular cases.

We finish the chapter (section 4.7) by studying in detail the reduction of the most promising operator, the minimum. And we compare it to the most extensive and comfortable t-norms, the Yager t-norms. We note that both operators are very similar, besides the fact that the reduced minimum is not associative, but computationally much lighter. This leads to the conclusion that this operator should be used for repetitive t-norm like aggregation or for large number of arguments.

4.1 Reducing t-norms

Let us begin with the t-norms. As we just indicated, the largest t-norm is the min operator. Let us consider reducing the result obtained from this t-norm operator by subtracting a value from it, in here the value is computed as $f(a,b)$. This gives us :

$$\min(a,b) - f(a,b) \tag{83}$$

We insure that the resulting operator does not give any negative result by means of the max operator :

$$G(a,b) = \max[\min(a,b) - f(a,b),\ 0] \tag{84}$$

In order to obtain an interesting operator, we consider a function f that has the following properties :

- $f(a,b) = f(b,a)$ (F1) **Commutativity**
- $f(a,b) \geq f(c,d)$, if $a \leq c$ and $b \leq d$ (F2) **Monotonicity (decreasing)**
- $f(a,1) = 0$ (F3) **One is a null factor**

Then G is commutative, increasingly monotone and 1 is a neutral element. We notice that G is almost a t-norm, but this operator is not necessarily associative.

The function f can be expressed in terms of the negation of a t-conorm, a generalization of the classical *'nor'* function (also called the Pierce function) :

$$f(a,b) = \beta \cdot \overline{S(a,b)} \tag{85}$$

where $\beta \geq 0$ is a parameter and S is a t-conorm. Using (85), expression (84) becomes :

$$G(a,b) = \max[\min(a,b) - \beta \cdot \overline{S(a,b)},\ 0] \tag{86}$$

We can generalize this result to other t-norms than the min operator. In that case we will have the general expression :

$$R^{T-S}(a,b) = \max[T(a,b) - \beta \cdot \overline{S(a,b)},\ 0] \tag{87}$$

We observe that $R^{T\text{-}S}$ is almost a t-norm, but this operator is not necessarily associative. However this operator is an *aggregation operator* as defined in chapter 1.

> **Proof :** In fact, we can define $R^{T\text{-}S}(x) = x$. For the boundary conditions, we have $T(0,0) = 0$ and $S(0,0) = 0$ and consequently $R^{T\text{-}S}(0,0) = 0$. In the same way $T(1,1) = 1$ and $S(1,1) = 1$ and consequently $R^{T\text{-}S}(1,1) = 1$. The reduced t-norm is non-decreasing because :

$$T(a,b) - \beta \cdot \overline{S(a,b)} = T(a,b) + \beta \cdot S(a,b) - \beta \qquad (88)$$

The t-norm is not decreasing, the t-conorm is not decreasing and the addition or multiplication by a positive constant does not change the monotonicity. Neither does the truncation with the max.

Other interesting basic properties of this aggregation operator are the *commutativity* and having 1 as *a neutral element*.

Proof : In fact, the reduced t-norm is commutative, because the t-norm and the t-conorm are commutative. And, 1 is a neutral element, because $T(a,1) = a$. Using the DeMorgan law and the property of the t-conorms (47), we have that $1 = \max(1,a) \leq S(1,a) \leq 1$ and so $\overline{S(1,a)} = 0$. Consequently :

$$R^{T-S}(1,a) = \max[a - \beta \cdot 0,\ 0] = a \qquad (89)$$

To resume we have that the reduced t-norm is a commutative aggregation operator with neutral element 1. If looked more in detail this operator, we will discover that it also satisfies the following interesting properties :

Property 1 : The reduced t-norm is *continuous*, if the underlying t-norm and t-conorm are continuous.

Proof : In fact, if the t-norm and the t-conorm are continuous, their linear combination is still continuous. And the truncation by the max gives also a continuous operator.

Property 2 : Zero is an absorbent element of the reduced t-norms, mathematically :

$$R^{T-S}(a,0) = 0 \qquad (90)$$

Proof : We know that because of the introduced max, we have that $R^{T\text{-}S}(a,0) \geq 0$. And using the monotonicity and the fact that 1 is neutral element, we have that, $R^{T\text{-}S}(a,0) \leq R^{T\text{-}S}(1,0) = 0$, since $a \leq 1$.

Property 3 : The operator defined in (87) can be written also using only t-norms : Using the DeMorgan laws ($S(a,b) = \overline{T(\bar{a},\bar{b})}$), we can write the expression (87) under the form :

$$R^{T_1-S_2}(a,b) = R^{T_1-T_2}(a,b) = \max[T_1(a,b) - \beta \cdot T_2(\bar{a},\bar{b}),\ 0] \qquad (91)$$

We observe that it is not compulsory to have twice the same t-norm. We can have $T_1 \neq T_2$. This remark applies also to the definition (87) : there are no special constraints on the choice of S and T.

We can consider now that β is a parameter that we vary in order to obtain different aggregation operators. We will have then :

Property 4 : for $\boldsymbol{\beta} = \mathbf{0}$, we obtain for every t-norm T : $\boldsymbol{R^{T\text{-}S}(a,b) = T(a,b)}$.

Proof : Trivial.

Property 5 : for $\boldsymbol{\beta} \to +\infty$, we obtain for every t-norm T reduced by non saturated t-conorm S (i.e. that if a≠1 and b≠1 then $S(a,b) \neq 1$) : $\boldsymbol{R^{T\text{-}S}} \to \boldsymbol{T_{drastic}}$, the drastic t-norm (see section 3.1.2).

Proof : In fact, for $a \neq 1$ and $b \neq 1$, we have for a t-norm without zero divisors, that $T(\bar{a},\bar{b}) > 0$. So for β large enough we will have $T(a,b) - \beta \cdot T(\bar{a},\bar{b}) < 0$, and in this way $\max[T(a,b) - \beta \cdot T(\bar{a},\bar{b}),\ 0] = 0$. For a = 1, we have that $T(\bar{a},\bar{b}) = 0$ and $T(a,b) = b \geq 0$. In this way we obtain $\max[T(a,b) - \beta \cdot T(\bar{a},\bar{b}),\ 0] = b$. The case b = 1 can be shown with the commutativity property.

Property 5 : for $\boldsymbol{\beta} = \mathbf{1}$, for every t-norm and t-conorm pair from the Frank t-norm family [44], $\boldsymbol{R^{T\text{-}S} = T_{Lukasiewicz}}$, the Lukasiewicz t-norm (see section 3.1.2).

Proof : In fact,

$$\begin{aligned} T(a,b) - T(\bar{a},\bar{b}) &= T(a,b) - \overline{S(a,b)} && \text{(using DeMorgan law)} \\ &= T(a,b) - (1 - S(a,b)) && \text{(using the definition of negation)} \\ &= T(a,b) + S(a,b) - 1 \end{aligned}$$

The t-norms of the Frank family satisfy $T(a,b) + S(a,b) = a + b$. So we obtain then the reduced t-norm $R^{T\text{-}S} = \max(a + b - 1, 0)$, which is the Lukasiewicz t-norm.

Note : the Frank family is a very large class of t-norms including the most popular t-norms as the product, the minimum and the Lukasiewicz. The precedent result is also true for ordinal sum of Frank t-norms and their dual copulas.

4.2 Reducing some typical t-norms

Now that we have shown some general properties, let us take a look at some particular cases. We will study in this section what happens when we reduce some typical t-norms :

Minimum : As we said, at the beginning of this chapter, the biggest t-norm is the minimum. We can reduce this t-norm. We choose here to use the dual pair (min, max). Following definition (91), we only need to use the min and we obtain :

$$R^{min-min}(a,b) = \max[\min(a,b) - \beta \cdot \min((1-a),(1-b)),\ 0] \qquad (92)$$

Another form for this same operator is (see definition (87)) :

$$R^{min-max}(a,b) = \max[\min(a,b) - \beta \cdot (1-\max(a,b)),\ 0] \qquad (93)$$

We observe that we obtain the expected properties :

- for $\beta = 0$, we obtain the minimum.
- for $\beta = 1$, we obtain the Lukasiewicz t-norm.
- for $\beta \rightarrow +\infty$, we tend to the drastic t-norm.

The "reduced min" seems to be an interesting operator that strangely looks like the Yager t-norm, without being associative, but instead having a lower computational effort. For details see the fast non-associative "Yager t-norm"-like operator at the end of this chapter (section 4.7).

Probabilistic t-norm : Another very famous t-norm is the product. Reducing the probabilistic t-norm by the dual t-conorm and using (91), we obtain the following definition :

$$R^{proba-proba}(a,b) = \max[a \cdot b - \beta \cdot ((1-a) \cdot (1-b)),\ 0] \qquad (94)$$

We remark here that the reduced probabilistic t-norm is associative and in this way it is a t-norm.

We observe that we obtain the expected properties :

- for $\beta = 0$, we obtain the product.
- for $\beta = 1$, we obtain the Lukasiewicz t-norm.
- for $\beta \rightarrow +\infty$, we tend to the drastic t-norm.

Here an interesting point is that this operator generalizes the probabilistic and the drastic t-norm in the same way as Hamacher t-norm family [48] does. We also notice that the reduced probabilistic t-norm generalizes also the Lukasiewicz t-norm, which is not the case for the Hamacher t-norm.

Let us now compare the reduced probabilistic t-norm with the Weber-Sugeno t-norms [85], defined for $\lambda > -1$ by :

$$T_{Weber}(a,b) = \max\left(\frac{a+b-1+\lambda \cdot a \cdot b}{1+\lambda}\ ,\ 0\right) \tag{95}$$

We observe that both of these operators generalize the same typical operators. And if we take a look closer we will discover that in fact the reduced probabilistic t-norm is the Weber-Sugeno t-norm for the parameter $\beta = \frac{1}{1+\lambda}$. We notice that in this case the operator is associative and therefore a t-norm.

Lukasiewicz t-norm : The Lukasiewicz t-norm seems to be central because of the theorem affirming that for every t-norm from the Frank family [44], for $\beta = 1$ we obtain the Lukasiewicz t-norm. What happens if we try now to reduce the Lukasiewicz t-norm by the corresponding t-conorm? Using the definition (91), we obtain :

$$R^{Luka-Luka}(a,b) = \max\left[\max(a+b-1{,}0) - \beta \cdot \max((1-a)+(1-b)-1),\ 0\right] \tag{96}$$

We observe that if $(a + b - 1) \geq 0$, then $((1-a)+(1-b)-1) = (1 - a - b) \leq 0$. So, we can reduce the expression to :

$$R^{Luka-Luka}(a,b) = \max(a+b-1{,}0) = T_{Lukasiewicz}(a,b) \tag{97}$$

which is exactly the Lukasiewicz t-norm. In other words we cannot reduce the Lukasiewicz t-norm using the associated t-conorm.

Note : We can reduce the Lukasiewicz t-norm by using another t-norm that is not the associated Lukasiewicz t-conorm.

Drastic t-norm : Another interesting case is the drastic t-norm. We know that the drastic t-norm is the smallest t-norm. We have for any t-norm T the following property :

- $T(a,b) \geq T_{drastic}(a,b)$ (98)

So, what happens if we try to reduce the drastic t-norm? Let us try to reduce the drastic t-norm with any t-conorm (not only the associated). Using the definition (87), we obtain :

$$R^{drastic-S}(a,b) = \max\left[T_{drastic}(a,b) - \beta \cdot \overline{S(a,b)},\ 0\right] \tag{99}$$

We observe that for $a \neq 1$ and $b \neq 1$, $T_{drastic}(a,b) = 0$ and so $T_{drastic}(a,b) - \beta \cdot \overline{S(a,b)} \leq 0$, so the reduced drastic t-norm equal 0. For $a = 1$, $T_{drastic}(a,b) = b$ and for any t-conorm we have $\overline{S(a,b)} = 0$. In other words the operator is the drastic t-norm. We can conclude by saying that we cannot reduce the drastic t-norm.

4.3 The augmented t-conorms

Now that we have studied the reduced t-norms, let us take a look at their dual operators. We start with the expression (87) and by using the DeMorgan law, we obtain :

$$\begin{aligned} F(a,b) &= \overline{R^{T-S}\left(\bar{a},\bar{b}\right)} \\ &= \overline{\max\left[T\left(\bar{a},\bar{b}\right) - \beta \cdot S\left(\bar{a},\bar{b}\right),\ 0\right]} \\ &= \min\left[\overline{T\left(\bar{a},\bar{b}\right) - \beta \cdot T(a,b)}, 1\right] \\ &= \min\left[1 - T\left(\bar{a},\bar{b}\right) + \beta \cdot T(a,b),\ 1\right] \\ &= \min\left[S(a,b) + \beta \cdot T(a,b),\ 1\right] \end{aligned}$$

The obtained operator can be understood as being a t-conorm to which we add β times a t-norm, the result of this addition being limited to 1. For this reason we will call this operator the augmented t-conorm :

$$A^{S+T}(a,b) = \min\left[S(a,b) + \beta \cdot T(a,b),\ 1\right] \tag{100}$$

We remark that this operator is dual of the reduced t-norms by construction. So, it is a commutative aggregation operator and it has zero as neutral element. Once more we do not have necessarily the associativity. If the dual reduced t-norm is associative then the augmented t-conorm is also associative.

Besides these basic properties we have :

Property 1 : The augmented t-conorm will be *continuous*, if the underlying t-norm and t-conorm are continuous.

> **Proof :** In fact, if the t-norm and the t-conorm are continuous, their linear combination is still continuous. And the truncation by the min gives also a continuous operator.

Property 2 : One is an absorbent element of the augmented t-conorms, mathematically :

$$A^{S+T}(a,1) = 1 \tag{101}$$

> **Proof :** We know that because of the truncation with the min, we have $R^{T\text{-}S}(a,0) \leq 1$. And using the monotonicity and the fact that 0 is neutral element, we have, $A^{S+T}(a,1) \geq A^{S+T}(0,1) = 1$, since $a \geq 1$.

Property 2 : The augmented t-conorm can be written using only t-conorms :

$$A^{S+T}(a,b) = \min\left[S(a,b) + \beta \cdot \left(1 - S\left(\bar{a},\bar{b}\right)\right), 1\right] \tag{102}$$

Proof : We used the DeMorgan law to obtain this result.

Property 3 : The augmented t-conorm has also the generalization properties :

- **For $\boldsymbol{\beta}$ = 0**, we obtain for every t-conorm *S* and any t-norm *T* : $\boldsymbol{A^{S+T} = S(a,b)}$.
- **For $\boldsymbol{\beta} \to +\infty$**, we obtain for every t-conorm *S* augmented by a t-norm *T* without zero divisors (i.e. that if a≠0 and b≠0 then $T(a,b) \neq 0$) : $\boldsymbol{A^{S+T} \to S_{drastic}}$, the drastic t-conorm (see section 2.7.3).
- **For $\boldsymbol{\beta}$ = 1**, for every t-norm and t-conorm pair from the Frank t-norms family [44], we obtain that $\boldsymbol{A^{S+T} = S_{Lukasiewicz}}$, the Lukasiewicz t-conorm (see section 2.7.3).

Property 4 : The augmented t-conorms and the reduced t-norms are dual by construction, if the underlying t-norms and t-conorms are dual. More generally, we have :

$$A^{S_1+T_2}(a,b) = 1 - R^{T_1-S_2}\left(\bar{a},\bar{b}\right) \tag{103}$$

where (T_1, S_1) and (T_2, S_2) are dual pairs.

Proof : see the construction of the augmented t-conorm in section 4.3.

4.4 Augmenting typical t-conorms

Let us now analyze what happens when we augment typical t-conorms :

Maximum : Using the maximum and augmenting it with the associated t-norm (i.e. minimum), we obtain the following operator (using definition (100)) :

$$A^{max+min}(a,b) = \min[\max(a,b) + \beta \cdot \min(a,b), 1] \tag{104}$$

We notice that this operator is the dual operator of the reduced minimum. It is not associative, but generalizes the following t-conorms : the maximum for $\beta = 0$; the Lukasiewicz t-conorm for $\beta = 1$ and for $\beta \to +\infty$, it tends to the drastic t-conorm

Probabilistic t-conorm : Using the probabilistic t-conorm and augmenting it with the associated t-norm, we obtain the following operator (using definition (100)) :

$$A^{proba+proba}(a,b) = \min[a + b + (\beta - 1) \cdot ab, 1] \tag{105}$$

We notice that this operator is the dual operator of the reduced probabilistic t-norm. It is associative and in this way it is a t-conorm. We also observe that this operator generalizes the following t-conorms :

- for $\beta = 0$, we obtain the probabilistic t-conorm
- for $\beta = 1$, we obtain the Lukasiewicz t-conorm.
- for $\beta \to +\infty$, we tend to the drastic t-conorm.

We observe that this operator generalizes the probabilistic t-conorm and drastic t-conorm in the same way as the Hamacher t-conorm family [48] does.

We also notice that the augmented probabilistic t-conorm generalizes the same typical t-conorms in the same way as the Weber-Sugeno t-conorm family [85] does. In fact the expression (105) is another form of the Weber-Sugeno t-conorm (where $\beta = \lambda + 1$) :

$$A^{proba+proba}(a,b) = \min[a + b + \lambda \cdot ab, 1] = S_{Weber}(a,b) \tag{106}$$

This operator is associative and therefore is a t-conorm.

Lukasiewicz t-conorm : Like in the dual case, it is impossible to augment the Lukasiewicz t-conorm, using the Lukasiewicz t-norm. This does not mean that it is impossible to augment the Lukasiewicz t-conorm, but we will need to use another t-norm than the Lukasiewicz one (i.e. something like the product or the minimum).

Drastic t-norm : Like in the dual case it is possible to show that the drastic t-conorm cannot be augmented, which is a natural result since drastic t-conorm is the largest t-conorm.

4.5 Mixed reductions and augmentations

Here we are interested in doing reduction and augmentation with pairs of t-norm and t-conorm that are not dual. In other words the reduction of a t-norm will be done using another t-conorm than the dual one. And we will augment the t-conorm with another t-norm than the dual one.

This kind of manipulation is particularly interesting in the case of the Lukasiewicz pair. We saw that it is impossible to reduce the Lukasiewicz t-norm by the Lukasiewicz t-conorm or to augment the Lukasiewicz t-conorm by the Lukasiewicz t-norm. Here we give two examples of reducing the Lukasiewicz t-norm and two of augmenting the Lukasiewicz t-conorm.

We can reduce the Lukasiewicz t-norm using the minimum (107) or using the product (108) :

$$R^{Luka-\min}(a,b) = \max\left[\max(a+b-1,0) - \beta \cdot (1-\max(a,b)),\ 0\right] \tag{107}$$

$$R^{Luka-prod}(a,b) = \max\left[\max(a+b-1,0) - \beta \cdot (1-a)\cdot(1-b),\ 0\right] \tag{108}$$

We observe that both of these operators have the following properties :

- for $\beta = 0$, we obtain the Lukasiewicz t-norm.
- for $\beta \to +\infty$, we tend to the drastic t-norm. (since the max and the product have no zero divisors).

In other words, as expected, we can reduce the Lukasiewicz t-norm by changing β. And we see that we can go from the Lukasiewicz t-norm until the drastic t-norm.

We have two dual cases that correspond to the augmented Lukasiewicz t-conorm : the first one by the minimum (109) and the second one by the product (110).

$$A^{Luka+\min}(a,b) = \min\left[\min(a+b,1) + \beta \cdot \min(a,b),\ 1\right] \tag{109}$$

$$A^{Luka+prod}(a,b) = \min\left[\min(a+b,1) + \beta \cdot a \cdot b,\ 1\right] \tag{110}$$

We still have for the limit cases :

- for $\beta = 0$, we obtain the Lukasiewicz t-conorm.
- for $\beta \to +\infty$, we tend to the drastic t-conorm (because the min and the product have no zero divisors).

In other words as expected, we can augment the Lukasiewicz t-conorm by augmenting β until we obtain the drastic t-conorm.

In the precedent examples we have two associative operators and two non-associative ones. It seems that knowing the associativity in advance for every operator is not an easy task. To know more about this kind of problem there is an interesting paper of Ling [61].

We have that :

- The Lukasiewicz t-norm reduced by the minimum is *not associative*. Therefore its dual operator, the Lukasiewicz t-conorm augmented by the maximum, is not associative.

  In fact, let us use a counterexample for $\beta = 1$:

$$R^{Luka-min}\left(R^{Luka-min}(0.75, 0.8), 0.85\right) = R^{Luka-min}(0.35, 0.85) = 0.05$$

and

$$R^{Luka-min}\left(0.75, R^{Luka-min}(0.8, 0.85)\right) = R^{Luka-min}(0.75, 0.5) = 0$$

- The Lukasiewicz t-conorm augmented by the product is *associative*. Therefore their dual operator, the Lukasiewicz t-norm reduced by the probabilistic t-conorm, is also associative.

  In fact, augmenting the Lukasiewicz t-conorm by the product gives (110) exactly the Weber-Sugeno t-norm :

$$A^{Luka+prod}(a,b) = \min(a + b + \beta \cdot a \cdot b, 1) \qquad (111)$$

  In fact, for $a + b > 1$, we have then $a + b + \beta ab > 1$ since $\beta \geq 0$.

  And reducing the Lukasiewicz t-norm by the probabilistic t-conorm gives Weber-Sugeno t-norm :

$$R^{Luka-prod}(a,b) = \max[a + b - 1 - \beta \cdot (1-a) \cdot (1-b),\ 0] \qquad (112)$$

  In fact, for $a + b - 1 < 0$, we have $a + b - 1 - \beta(1-a)(1-b) < 0$ since $\beta \geq 0$.

In conclusion, augmenting the Lukasiewicz t-conorm by the product or reducing the Lukasiewicz t-norm by the probabilistic t-conorm gives known parameterized t-norm and t-conorm families.

4.6 The difficulty of combining t-norms and t-conorms

In this chapter we studied how to reduce a t-norm by subtracting to it a value given by a function. Using a commutative, monotone decreasing function that has one as null factor, we obtain an interesting aggregation operator, which is almost a t-norm besides the fact that it is not always associative. This very particular side effect shows the difficulty of reducing t-norms.

If we impose the associativity to the function being subtracted, we discover that it is actually β times the negation of a t-conorm. Then we study some interesting general properties and after that, we study some particular cases by reducing some typical t-norms. We obtained for each typical t-norm an interesting parameterized family. We compare them to some existing parameterized t-norms.

Then using the DeMorgan law we obtain a dual operator. We find that these kinds of operators are augmented t-conorms. In fact this operators are built by adding to a t-conorm, β times a t-norm. This result is very interesting because it suggests that in order to make more drastic a t-conorm we need to add a t-norm. Here, once again, we do not obtain always a t-conorm, because the resulting operator is not always associative. We also study some interesting particular cases by augmenting typical t-conorms.

We also take a look at the reduction and augmentation of non dual t-norms and t-conorms. This situation is particularly interesting for the Lukasiewicz case. Since we showed that the Lukasiewicz t-norm (or t-conorm) cannot be reduced (or augmented) by its dual operator. Augmenting and reducing Lukasiewicz by the minimum, we obtain an interesting non-associative aggregation operator. But when augmenting and reducing by the minimum we obtain a parameterized t-conorm and a parameterized t-norm.

4.7 The n-ary operators

We have until now studied only the aggregation of two arguments, but since in general the reduced t-norms and the augmented t-conorms are not associative (as pointed out in the previous paragraph), we need to specify how to aggregate *n* arguments.

There are several ways of extending a two arguments operator to *n* arguments. As for instance aggregate by packages of two or also aggregate the first argument with the second one, the result with the third one, and so on. We choose here to explicitly introduce a n-ary form, which allow us to have a pseudo-associativity.

The n-any reduced t-norm is defined by :

$$R^{T-S}(x_1,\ldots,x_n) = \max[T(x_1,\ldots,x_n) - \beta \cdot \overline{S(x_1,\ldots,x_n)},\ 0] \qquad (113)$$

It is pseudo-associative in the sense that if we keep in memory the values $T(x_1,\ldots,x_n)$ and $S(x_1,\ldots,x_n)$, we can profit from the advantages of the normal associativity. For example a nice benefit of having the associativity is that if we observe a new argument we can with the last result obtain a final aggregation that is equivalent to re-computing using all the arguments. Mathematically this can be translated by :

$$R^{T-S}(R^{T-S}(x_1,\ldots,x_n), x_{n+1}) = R^{T-S}(x_1,\ldots,x_n, x_{n+1}) \qquad (114)$$

We can obtain this by simply keeping the values $T(x_1,\ldots,x_n)$ and $S(x_1,\ldots,x_n)$. With them we first calculated $R^{T\text{-}S}(x_1,\ldots,x_n)$.

Then when we observe x_{n+1} we can use the associativity of the t-norm and of the t-conorm to obtain $T(x_1,\ldots, x_n, x_{n+1})= T(T(x_1,\ldots,x_n), x_{n+1})$ and $S(x_1,\ldots, x_n, x_{n+1})$. Which will lead to $R^{T\text{-}S}(x_1,\ldots,x_n, x_{n+1})$.

In the same way, we define the n-any augmented t-conorm :

$$A^{S+T}(x_1,\ldots,x_n) = \min[S(x_1,\ldots,x_n) + \beta \cdot T(x_1,\ldots,x_n), 1] \qquad (115)$$

This operator has also the pseudo-associativity property as long as we keep in memory the values $T(x_1,\ldots,x_n)$ and $S(x_1,\ldots,x_n)$.

4.8 Fast non-associative "Yager t-norm"-like aggregation

Here we study a parameterized aggregation operator that was obtained by reducing the minimum by its dual operator, the maximum. The resulting operator has properties very similar to the Yager t-norm. This operator is not associative, but we will see that it is computationally lighter than Yager t-norm. So, we recommend the use of this operator for the aggregation of a large number of arguments.

Here we will also see that among the common properties, the t-norm and our operator generalize the most basic t-norm : the Lukasiewicz t-norm, the drastic t-norm and minimum t-norm. The last one being attempt not as limit case (as for Yager's family), in other words it is more computational friendly.

The Yager t-norm

The Yager t-norm was introduced in [89] as a general form of a t-norm (see section 2.7.4),. The definition of this new operator is based on a parameter that changes the type of the t-norm. Since this operator is associative there is no ambiguity on its n-any form. For $p > 0$ we have :

$$T_{Yager}(x_1, \dots, x_n) = \max\left(1 - \left[\sum_{i=1}^{n} (1 - x_i)^p\right]^{1/p} , \ 0\right) \qquad (116)$$

We saw that this parameterized t-norm generalizes some of the usual t-norms :

- For $p \to 0$, the Yager t-norm ***tends*** to the drastic t-norm.
- For $p = 1$, the Yager t-norm is the Lukasiewicz t-norm.
- For $p \to +\infty$, the Yager t-norm ***tends*** to the minimum t-norm

Let us now take a look at the fast non-associative "Yager t-norm"-like operator. In fact it is what we called the reduced min (see 4.7).

The reduced min

Since this operators is not associative its two arguments definition (92) is limited. Here we propose a n-ary form based in the definition (113). We define the n-ary reduced min for $\beta \geq 0$ by :

$$R^{\min - \max}(x_1, \dots, x_n) = \max\left(\beta \cdot \left(\max(x_1, \dots, x_n) - 1\right) + \min(x_1, \dots, x_n), \ 0\right) \qquad (117)$$

4.8.1 Comparison with the Yager t-norm

As a result of the previous work, we know that, as the Yager t-norm, the new operator (117) is commutative, monotone, has the unity as neutral element (i.e. $R^{\text{min-max}}(a,1)=a$) and zero is an absorbent element (i.e. $R^{\text{min-max}}(a,0)=0$). The two last properties translate the fact that these two operators are equal on the edges of the unit square (see Figure 2).

Another result from the previous work is that it generalizes the same usual t-norms as the Yager t-norm. For seek of simplicity we will reason on binary operators, which will not affect the generality since the particular cases are associative :

- For $\beta = 0$ we have the **minimum** t-norm.

  In fact, for $\beta = 0$, the t-norm is reduced to

  $$R^{\text{min-max}}(a,b) = \max(\min(a,b),0) = \min(a,b).$$

- For $\beta = 1$ we have the **Lukasiewicz** t-norm.

  In fact, for $\beta = 1$, the t-norm is reduced to

  $$R^{\text{min-max}}(a,b) = \max(\max(a,b)+\min(a,b)-1,0).$$

  Now $\max(a,b)+\min(a,b) = a + b$, therefore $R^{\text{min-max}}(a,b) = \max(a + b-1,0)$.

- For $\beta \to +\infty$, the t-norm tends to the **drastic** t-norm.

  In fact, for $a \neq 1$ and $b \neq 1$, $\max(a,b)-1 < 0$.

  Therefore $\beta \cdot (\max(u,v) - 1) + \min(u,v) \to -\infty$, when $\beta \to +\infty$.

  And so, $R^{\text{min-max}}(a,b) = 0$.

  For $b=1$, we have $\max(a,b)=1$ and $\min(a,b)=a$, and therefore

  $R^{\text{min-max}}(a,b) = \max(0+a,0) = a$. Since the t-norm is commutative we have finished the proof.

Using the methodology presented in chapter 3 we see that the range of action is the same for the two operators : the smallest operator being the drastic t-norm and the biggest the minimum t-norm. We also observe that neither of these two operators generalize the probabilistic t-norm.

Another common property is the fact that both of these operators have zero divisors (i.e. there exist two numbers a and b both different from zero, so that $R^{\text{min-max}}(a,b) = 0$). This is true except when the reduced min equals the minimum.

So far, we have seen that the Yager t-norm and the new defined operator have similar behaviors. Let us now point out some differences.

Computational Effort

Let us now focus on the computational effort aspects. If we look at the mathematical definition of the Yager t-norm (116) and at the reduced min (117) we will observe that these two operators have the form :

$$\max\left(R(x_1,\dots,x_n),\ 0\right) \tag{118}$$

For the Yager t-norm, we have

$$R(x_1,\dots,x_n) = 1 - \left[\sum_{i=1}^{n} (1-x_i)^p\right]^{1/p} \tag{119}$$

For the reduced min, we have

$$R(x_1,\dots,x_n) = \beta \cdot \left(\max\left(x_1,\dots,x_n\right) - 1\right) + \min\left(x_1,\dots,x_n\right) \tag{120}$$

Now let us compare both equations from an arithmetic point of view :

Yager $R(x_1,\dots, x_n)$	**New $R(x_1,\dots, x_n)$**
1 division - 1/*p*	1 product
n-1 additions	1 addition
n+1 subtractions	1 subtraction
n+1 power operations	1 min and 1 max

Table 10. Effort comparison between Yager t-norm and the reduced min.

Looking at Table 10, we notice that the new operator is computationally lighter. In fact, we observe that Yager t-norm has a division that has the same complexity as a product.

We see that the more arguments we aggregate the more heavy the calculus of the Yager t-norm is. In fact we increase linearly the number of additions, subtractions and of power operations. For the new operator when we increase the number of arguments we increase the difficulty of computing the min and the max. In fact, both can be calculated with simply *n*-1 comparisons.

But even in the case *n*=2, Yager t-norm is more arduous. The two operators have then same amount of additions, but Yager t-norm has one more subtraction. This does not make a large difference. The main difference appears when we observe that Yager t-norm needs 3 power operations, while the new operator only needs one comparison : the bigger number will be used for the max and the other one for the min.

In other words the new operator is interesting for the calculus of large number of arguments, but also when aggregation several times just couple of values.

Other advantages

The fact that the reduced minimum is computationally faster than the Yager t-norm is interesting, but we may think that the price for this is that we lost the associativity property. In fact, the reduced minimum is not associative, but it is pseudo-associative in the sense presented in section 4.7. As we as when introducing the associativity (see section 1.2.4) the idea of the associativity is to allow the aggregation par packages. So, if we keep in memory the maximum and minimum of this packages we will witness the same features as when using the associativity.

From the generalization point of view, we noticed that both of the two families generalize the most common t-norm : the minimum t-norm. But what we did not say is that Yager t-norm generalizes the minimum t-norm as a limit case. In other words the parameter p has to take the value $+\infty$ to obtain exactly the expected behavior. We all know that this is not possible in a computer. If we look at the reduced min, we observe that the generalization of the most common t-norm is done for the particular case $\beta = 0$, which is not a limit case.

Note : We observe that for the generalization of the drastic t-norm, we have in both cases a limit case. This is normal since the two parameterized t-norms are families of continuous operators and the drastic t-norm is non-continuous.

Concluding remarks

The reduced minimum is a particularly interesting operator for efficient practical applications, where we have to use a t-norm. Its computational lightness combined to its pseudo-associativity, without forgetting its generalization property make of it a power-full tool. See chapter 7 for an illustration of its use.

Chapter 5

Aggregating Truth and Falsity Values

The problem of aggregating truth values is at the core of the studies in fuzzy logic. But it is to notice that the purpose of this aggregation is to compute the truth value of a logical phrase. Here we are interested in the aggregation of different truth values observed for the same logical phrase.

In the first case, we compute the truth value of a phrase of the type : "the figure *is* quadrilateral 'AND' the angle on the left *is* a right angle". Here it is clear that if one statement is true and the other false then the truth value of the whole assertion will be false, which follows the classical logic.

In the second case, we observe two different truth values for the same statement. For example we observe once that "the angle on the left *is* a right angle" is true and then we observe that it is false. This does not mean that the statement is completely false, we just can conclude that we do not know if it is true or false. In this chapter we try to give an answer by proposing operators that compute the degree of truth using the two observed degrees.

Before the construction of such operators we define a truth scale, on which we work. Then we propose an axiom set for the aggregation of truth values, which leads to the characterization of two truth-aggregation families, a *prudent* and an *enthusiastic*. The first one has a cautious attitude choosing between two observed values the one that is more uncertain. The second one has an enthusiastic behavior and will reinforce the result if it observes twice the truth or twice the falsity. When observing falsity and truth the operator gives a compensated value.

We determine that the *prudent* aggregation is completely determined, while the construction of an *enthusiastic* aggregation is more delicate, leading usually to a non continuous structure. We propose a particular solution that is continuous except for the full contradiction, which conducts us to study in general the full contradiction and the normal contradiction of the *enthusiastic* aggregation. We obtain a complete characterization of these two situations.

We finish by expounding the use of the truth and falsity aggregation operators and their relationship with the traditionally used truth-aggregation operators : the t-norms and t-conorms. Actually the presented operators should be used for the aggregation of different observed truth values for the same phrase vs. the calculus of the truth of a logical phrase.

5.1 Truth fuzzy set and falsity fuzzy set

Here we are going to assume the truth as a fuzzy set, where 1 is the full truth and 0 represents the zero degree of membership of being true. In the same way we can imagine a fuzzy set for the falsity, but this time we are going to use the negative scale and consider -1 the full falsity and 0 the zero degree of membership of being false. In this way we build a scale of "truth" we can pass progressively from the total falsity to the total truth. So we work on the scale [-1,1] and we will say that [0,1] is the degree of truth and [-1,0] the degree of falsity, where -1 is the full falsity.

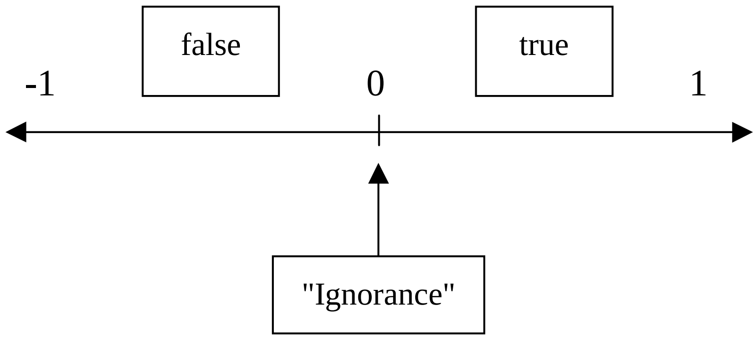


Figure 8. Scale of truth and falsity

In this construction an interesting point is 0. It is the middle point between the full truth and the full falsity. It is actually the null membership to the sets of truth and falsity. So we will consider that it represents the "ignorance", since it is neither true nor false.

We can relate these two fuzzy sets with a negation operator that transforms a truth value into a falsity one with the same degree : $n(x)= -x$. The negation of full truth is full falsity and vice-versa. We also observe that the fixed point of the negation is 0 corresponding to the total ignorance.

5.2 Basic properties of the operator

On this scale, how can we aggregate the truth values ? In order to build an operator we first dictate some *very useful* conditions for the aggregation (for more details about the properties see chapter 1). The operator should be :

- **Monotone :** The decrease of any of the arguments to be aggregated cannot produce an increase of the total value. This condition is related to the Pareto optimum.
- **Commutative :** This property ensures that the aggregation is symmetric, that is, indifferent to the order of the arguments to be aggregated.
- **Associative :** we consider that the total result should be independent of the grouping of the arguments. This may be regarded as a criterion of objectivity. This axiom also enables us to extend the operator from two arguments to more than two.

The associativity and commutativity taken together enable us to avoid some problems inherent to the nature of the systems :

- The architecture of the aggregation : if we aggregate the sources at different levels or steps the associativity and commutativity guarantee that the final result is independent of these levels or steps.
- The temporal aspects of obtaining the data : the order and the size of the parcels received in the time will not have any influence on the total result. We will obtain a final result equal to the aggregation considering that we disposed of all the information from the beginning.
- The computability capacity of the machine executing the aggregation : if the quantity of information is bigger than the memory of the machine processing the aggregation then we will need to do partial aggregations before getting the final result. The associativity and commutativity guarantee that the final result is independent of the method used to make these partial results.

5.3 The prudent aggregation

Now that we fixed these axioms relative to the general behavior, let us identify some attitudes.

For instance by saying that we want to be *prudent* (*cautious*) while aggregating, let *Prud*(x,y) denote this operator. We translate this condition as a simple constraint :

Let us imagine that we observe first a full truth and then for the same observation another truth value. What will be the aggregation of these two degrees? If we are *prudent* (*cautious*) we will consider that the final membership should be equal to the partial truth degree, since we observe first something completely true and then a doubt on this value, a *prudent* attitude is to keep the doubt. Mathematically we translate this by :

$$\text{for all } x \in [0,1] \quad Prud(1,x)=x \tag{121}$$

A simple choice like this can have many consequences. For instance :

The aggregation of the total ignorance with any truth value equals the total ignorance. Mathematically :

Proposition 1 : For all $x \in [0,1]$ $Prud(0,x)=0$.

Proof. Since the aggregation of two truth values should be a truth value ($Prud(0,x)\geq 0$), and using the monotonicity : $0 \leq Prud(0,x) \leq Prud(0,1) = 0$

The *prudent* aggregation of two truth values will always give a smaller (or equal) value than the smallest of any of the values being aggregated. It is exactly what we expect from a *prudent* (*cautious*) aggregation. Mathematically :

Proposition 2 : For all $(x,y)\in[0,1]^2$ $Prud(x,y) \leq \min(x,y)$

Proof. Using the monotonicity we have $Prud(x,y) \leq Prud(x,1) = x$

and by commutativity $Prud(x,y) = Prud(y,x) \leq Prud(y,1) = y$

Hence, $Prud(x,y) \leq x$ and $Prud(x,y) \leq y$, that is, $Prud(x,y) \leq \min(x,y)$.

Note : We notice here that the operator actually used on $[0,1]^2$ is a t-norm (see 2.7).

5.3.1 Prudent aggregation of the falsity values

Using the negation we can build the operator to be used for the falsity fuzzy set :

$$\text{For all } x,y \in [-1,0]^2 \quad Prud(x,y) = n(Prud(n(x),n(y))) \tag{122}$$

It is easy to show that this transformation keeps the associativity, the monotonicity and the commutativity.

Other consequences are :

If we observe something completely false and then a doubt on this falsity, the *prudent* aggregation will choose the doubt value. Mathematically :

Proposition 3 : For all $x \in [-1,0]$ $Prud(-1,x) = x$

Proof. Obvious using (122).

The *prudent* aggregation of "ignorance" and any falsity value will give full "ignorance".

Proposition 4 : For all $x \in [-1,0]$ $Prud(0,x) = 0$.

Proof. Obvious using (122).

The *prudent* aggregation of falsity values will always give a less or equally false value than the least false of the values being aggregated.

Proposition 5 : For $(x,y) \in [-1,0]^2$ $Prud(x,y) \geq \max(x,y)$.

Proof. Using (122) and proposition 2.

Note : We notice here that the operator actually used on $[-1,0]^2$ is a t-conorm (see section 2.7) shifted from $[0,1]^2$.

Another way of obtaining the operator to be used on the falsity domain is to assume proposition 3 as an axiom. But, we choose to use the negation for the construction in order to impose exactly the same behavior for the falsity and for the truth.

5.3.2 Prudent aggregation of falsity and truth values

Let us now take a look at what happens if we aggregate a truth value with a falsity value. In this case we obtain the ignorance, which corresponds to our *prudent* attitude.

Mathematically, we have :

Proposition 6 : For $(x,y) \in [-1,0]x[0,1] \cup [0,1]x[-1,0]$ $Prud(x,y) = 0$

Proof. Let us assume that $0 \leq x$ and $y \leq 0$. Using the monotonicity we obtain :

$0 = Prud(0,y) \leq Prud(x,y) \leq Prud(x,0) = 0$

and for $x \leq 0$ and $0 \leq y$, we have $0 = Prud(x,0) \leq Prud(x,y) \leq Prud(0,y) = 0$

A consequence of the precedent result is that the *prudent* operator is continuous on $[-1,1]^2$. (of course if and only if the underlying t-norm is continuous). In fact if the underlying t-norm is continuous, then the operator is continuous on $[-1,0]^2 \cup [0,1]^2$. Proposition 6 shows that the operator on $[-1,0]x[0,1] \cup [0,1]x[-1,0]$ equals 0 and using proposition 1 and 4, we have the continuity on $[-1,1]^2$.

This property is interesting since it translates the fact that the variation of the aggregated value do not jump from a value to another. It gives some stability to the result.

The general behavior : The associativity of the global operator was showed in [11]. We can resume the *prudent* aggregation on the following figure, where *T* is a t-norm :

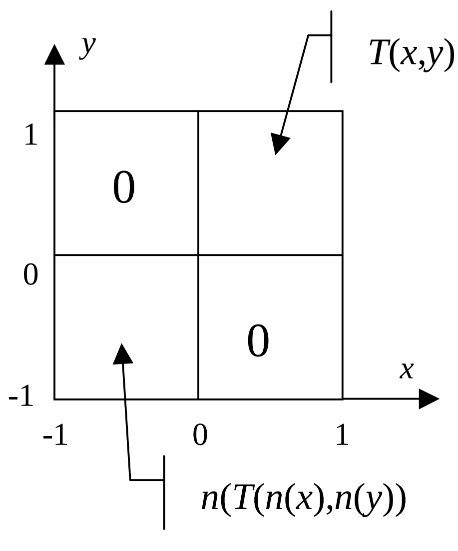


Figure 9. *Prudent* aggregation

Interpretation : We showed in proposition 1 that when aggregating two truth values the resulting value will be a truth value smaller than the smallest of the values to be aggregated.

When aggregating two falsity values, proposition 2 applies, which means that the result is always bigger that the maximum of the values. And the aggregation of two falsity values will be a smaller falsity value than any of the falsity values to aggregate.

As we already said in proposition 6, the aggregation of a truth value with a falsity value will give the total "ignorance".

Summing up, the general tendency of this operator is to converge to the "ignorance". If we have two truth values we will compute an aggregated value that is closer to the total ignorance than any of the two initial values.

5.3.2 Comparison with known operators

Here we work on the scale [-1,1], but generally the aggregation operators are defined on [0,1]. So, in order to compare this operator to others, we need to do a scale transformation.

We notice that the construction of the prudent operator is based on a t-norm and a negation. So, we can obtain t-norms and t-conorms by simple linear transformation :

$$\text{For all } x,y \in [0,1]^2 \quad S(x,y) = -Prud(-x,-y) \tag{123}$$

$$\text{For all } x,y \in [0,1]^2 \quad T(x,y) = Prud(x,y) \tag{124}$$

But maybe the most natural transformation is a linear transformation from [-1,1] to [0,1]. In this case we obtain a particular nullnorm (see section 2.9.5) :

$$\text{For all } x,y \in [0,1]^2 \quad N_{1/2}(x,y) = \frac{1}{2} \cdot (Prud(2x,2y) - 1) \tag{125}$$

This nullnorm is particular since the absorbent element *a* equals 1/2 and the underlying t-norms and t-conorms are dual. Note that it is easy to conceive a non linear transformation so that we obtain a nullnorm with any absorbent element *a* from a prudent operator.

5.4 The enthusiastic aggregation

We studied in detail the construction of a *prudent* operator, let us now take a look at the *enthusiastic* aggregation. Let us denote by $Enthu(x,y)$ such an aggregation operator. Of course we still request the operator to be associative, monotone and commutative. But this time we will impose that if we observe the total ignorance and a truth or a falsity value, then enthusiastically we will use this value as the final aggregation.

Mathematically we have

$$\text{For all } x \in [-1,1] \ \ Enthu(0,x) = x \qquad (126)$$

The choice of these characteristics for the operator has the following consequences :

The *enthusiastic* aggregation of the total truth with any other truth value will be the total truth. Mathematically :

Proposition 7 : For all $x \in [0,1]$ $Enthu(1,x) = 1$.

Proof. Using the monotonicity we have $1 = Enthu(0,1) \leq Enthu(x,1) \leq 1$

The *enthusiastic* aggregation of the total falsity with any other falsity value will be the total falsity :

Proposition 8 : For all $x \in [-1,0]$ $Enthu(-1,x) = -1$.

Proof. Using the monotonicity we have $-1 \leq Enthu(-1, x) \leq Enthu(-1,0) = -1$

We have a reinforcement when observing two truth values. That is to say that the observation of two truth values will be aggregated into a value bigger than the biggest of the observed values. Mathematically :

Proposition 9 : For all $(x,y) \in [0,1]^2$ $Enthu(x,y) \geq \max(x,y)$.

Proof. Using the monotonicity we have $x = Enthu(x,0) \leq Enthu(x,y)$ and by commutativity $Enthu(x,y) = Enthu(y,x) \geq Enthu(y,0) = y$.

Hence, $Enthu(x,y) \geq x$ and $Enthu(x,y) \geq y$, that is, $Enthu(x,y) \geq \max(x,y)$.

The same way, we have a reinforcement when observing two falsity values. We obtain a bigger falsity result than what was observed. Mathematically :

Proposition 10 : For all $(x,y) \in [-1,0]^2$ $Enthu(x,y) \leq \min(x,y)$

Proof. Since $(x,y) \in [-1,0]^2$ and using the monotonicity we have

$Enthu(x,y) \leq Enthu(x,0) = x$ and by commutativity

$Enthu(x,y) = Enthu(y,x) \leq Enthu(y,0) = y$

Hence, $Enthu(x,y) \leq x$ and $Enthu(x,y) \leq y$, that is, $Enthu(x,y) \leq \min(x,y)$.

When observing truth and falsity we have a compensatory behavior. In fact the final value will be closer to the total ignorance than any of the observed values. Mathematically :

Proposition 11 : For all $(x,y) \in [-1,0]\text{x}[0,1] \cup [0,1]\text{x}[-1,0]$ $\min(x,y) \leq Enthu(x,y) \leq \max(x,y)$

Proof. Let assume that $y \leq 0 \leq x$. Using the monotonicity we obtain :

$\min(x,y) = y = Enthu(0,y) \leq Enthu\,(x,y)$

and $Enthu\,(x,y) \leq Enthu\,(x,0) = x = \max\,(x,y)$

Note : We remark that we actually have a t-conorm on $[0,1]^2$ and on $[-1,0]^2$ we observe a t-norm shifted from $[0,1]^2$.

In order to obtain a similar behavior on the falsity and truth domain, we can request the operator to be self-involutive with respect to the negation. Mathematically we want :

$$\text{For all } x,y \in [-1,1]^2 \; Enthu(x,y) = n(Enthu(n(x),n(y))) \qquad (127)$$

If we reason in terms of t-norms and t-conorms, what we want is that these two operators with the negation form a DeMorgan triple.

5.4.1 Enthusiastic aggregation of truth and falsity

At this point we characterized the behavior of our operator on $[0,1]^2$ and on $[-1,0]^2$, which corresponds to the aggregation of truth with truth or of falsity with falsity. But how to aggregate falsity and truth, so that the operator keeps its properties on $[-1,1]^2$? An immediate solution is to use a construction similar to the ordinal sum [61]. This method consists in using a min or a max on $[-1,0]\mathrm{x}[0,1] \cup [0,1]\mathrm{x}[-1,0]$. We obtain, this way, two operators, a *negative enthusiastic* and a *positive enthusiastic*.

The *negative enthusiastic* will use the min in order to aggregate the truth with the falsity, and it will always choose the falsity over the truth. Figure 10 represents this kind of operator, where *S* is a t-conorm.

The *positive enthusiastic* will use the max and will always choose the truth over the falsity.

These two families seem to provide a nice aggregation operator. But if we take a look closer at the *negative* and *positive enthusiastic* operators, we will notice that they have an annoying characteristic. These operators are not continuous around the total ignorance.

For instance, a *negative enthusiastic* aggregation of a very true value with a quasi ignorance (but true) or with a quasi ignorance (but false), will give a very different result. In the first case, because of proposition 9 we know that the aggregation will be bigger that the maximum of the two true values (i.e. a very true value close to 1).

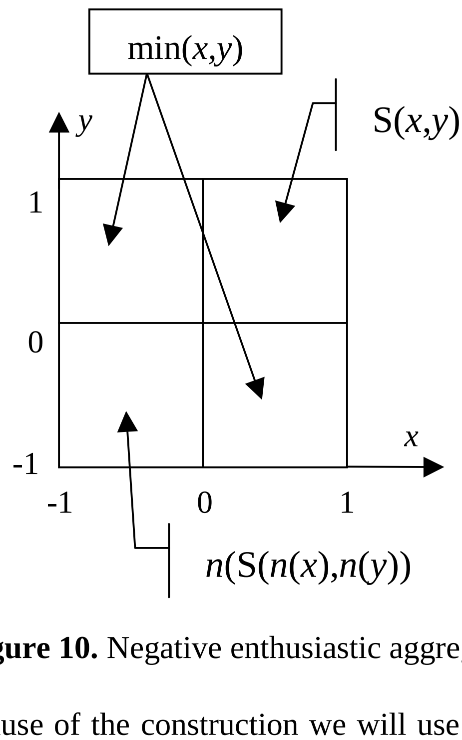


Figure 10. Negative enthusiastic aggregation

In the second case, because of the construction we will use the min, which will give a quasi ignorance but false, (i.e. a falsity value close to 0). So the result jumps for these two very close observations from a value close to 1 to a value close to 0. Mathematically this is translated by a discontinuity on $\{(x,0) \;/\; x\in]0,1]\}$ and on $\{(0,x) \;/\; x\in]0,1]\}$.

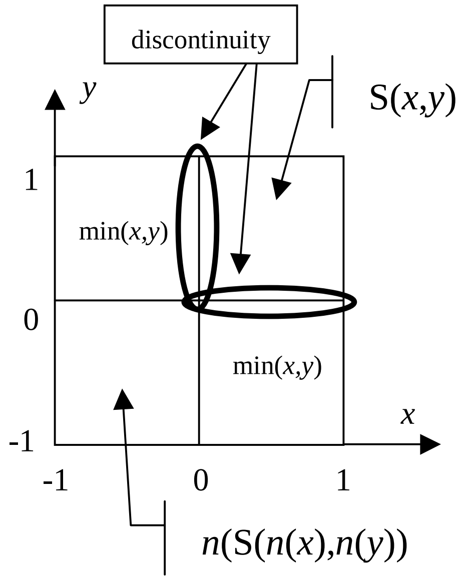


Figure 11. Discontinuity on negative enthusiastic aggregation

Even if this characteristic is not suitable for the aggregation it is clearly the consequence of the choice of our attitude. We are *enthusiastic*, so if we observe two truth values we take the best one, forgetting about the value close to the total ignorance. But if this quasi ignorance becomes a falsity value then, since we have a *negative* attitude we will choose the falsity over the truth.

We may think that the discontinuity is related to *negative* attitude of the operator. But unfortunately we observe the same problem for the *positive enthusiastic* operators. This time the aggregation of a total falsity value and two different quasi ignorance values will give very different results. In fact, we are *enthusiastic*, so if we observe two falsity values we take the certain one (i.e. the value close to -1), forgetting about the value close to the total ignorance (see property 10). But if this quasi ignorance becomes a truth value, then since we have a *positive* attitude we will choose the truth over the falsity.

Mathematically this is translated as a discontinuity on $\{(x,0) \setminus x \in [-1,0[\}$ and on $\{(0,x) \setminus x \in [-1,0[\}$.

5.4.2 Continuous enthusiastic aggregation

Since these discontinuities can be annoying, we can built a continuous operator using a method inspired by the representation theorem of the continuous Archimedean t-norms and t-conorms. In concrete terms we will base the construction of the continuous enthusiastic operator on the use of an additive generator *h*. Given that on $[0,1]^2$ the enthusiastic operator equals a t-conorm, our generator *h* should equal a generator *g* of a t-conorm on [0,1].

Mathematically we have that :

$$\text{For all } x \in [0,1] \quad h(x) = g(x) \tag{128}$$

where the function $g : [0,1] \rightarrow [0,+\infty]$ is continuous, strictly increasing, with $g(0) = 0$ and $g(1) = +\infty$

In order to obtain a similar behavior on the falsity domain, we use the negation to obtain the generator on [-1,0] :

$$\text{For all } x \in [-1,0] \quad h(x) = n(g(n(x))) = -\, g(-x) \tag{129}$$

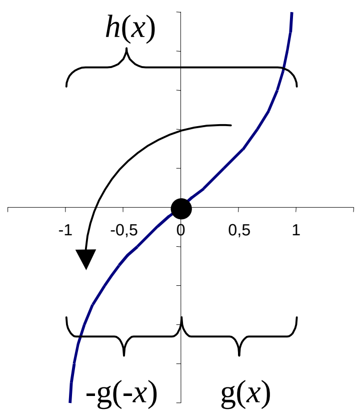


Figure 12. Enthusiastic aggregation generating function

We obtain a function $h : [0,1] \rightarrow]-\infty,+\infty[$, which is a continuous, strictly increasing bijection, symmetric with respect to the point (0,0) and satisfying $h(0) = 0$. The continuous enthusiastic operator being defined by :

$$Enthu(x,y) = h^{-1}(h(x)+h(y)) \tag{130}$$

We verify that we have an *enthusiastic* operator, since it is by construction associative, commutative and for all $x \in [-1,1]^2$ $Enthu(0,x) = x$ (because $h(0)=0$).

Automatically we inherit the properties 8, 9 and 10, which translate the reinforcement behavior when observing twice the truth (or twice the falsity) and compensation when confronting truth and falsity.

A particularity of this family is that when observing the total truth and then any other value (even if it is false), we will enthusiastically take the certain value (the total truth) as the aggregated value. Mathematically :

Proposition 12 : For all $x \in]-1,1]$ $\quad Enthu(x,1) = 1$.

Proof. Obvious taking into account that $h(1)=g(1)=+\infty$

The same happens, when we observe the total falsity and any other value, then we choose the total falsity.

Proposition 13 : For all $x \in [-1,1[$ $\quad Enthu(x,-1) = -1$.

Proof. Obvious taking into account that $h(-1) = - g(-(-1)) = - g(1) = - \infty$

Representation Theorem : We have shown that our construction gives an operator with a certain number of remarkable properties, but it is interesting to notice that it has been shown (based on the results of [1]) in [55] that if we request our operator to be associative, monotone, commutative, self involutive, with a neutral element and continuous on $[-1,1]^2$ except on the points (-1,1) and (1,-1), then it can be written under the form (130).

This result shows in particular that if we want an operator satisfying the basic properties (see section 5.2), having the same behavior for the truth and for the falsity (i.e. self involutive) and enthusiastic with respect to the total ignorance, then the only continuous solution is the one presented here.

Under the light of propositions 12 and 13 a natural question arises : what happens when confronting the total truth with the total falsity? Actually formula (130) does not give any answer. Mathematically we observe that (130) is continuous on $[-1,1]^2$ besides the limit points (0,1) and (1,0), where the function is undefined. This translates the impossibility of giving a value when observing a total truth and a total falsity. Let take us a closer look at this "full contradiction" in the next section.

5.4.3 The full contradiction

Let us assume that the enthusiastic operator gives a value k as result of a full contradiction :

$$k = Enthu(-1,1) \qquad (131)$$

The first thing we may think is that the total contradiction should equal the total ignorance. But this is impossible.

Proposition 14 : The full contradiction is not the total ignorance ($k \neq 0$).

Proof. Let us suppose that $k = 0$.

Let us now aggregate twice the total truth and a total falsity $Enthu(-1, 1, 1)$.

On the one hand using the associativity and the boundary condition property, we have $Enthu(-1, 1, 1) = Enthu(-1, Enthu(1, 1)) = Enthu(-1, 1) = k$.

On the other hand using again the associativity, the neutrality of the total ignorance in the enthusiastic case and the hypothesis ($k = 0$), we have

$Enthu(-1, 1, 1) = Enthu(Enthu(-1,1),1) = Enthu(k,1) = Enthu(0, 1) = 1$.

So, we obtain $k = 1$, which in contradiction with the hypothesis.

If the full contradiction do not equal the total ignorance, what can then it be? The answer is presented in the Proposition 15 :

Proposition 15 : The full contradiction equals either the total truth ($k = 1$) or the total falsity ($k = -1$) (*Note* : or is not defined).

Proof. We still suppose that the full contradiction gives a value k. We do again the same reasoning as for the previous proposition but this time we know that $k \neq 0$.

Let us suppose that the full contradiction is a truth value ($k>0$) and let us aggregate twice the total truth and a total falsity :

Using the associativity and the boundary condition property, and the enthusiastic property described in Proposition 7, we have

$k = Enthu(-1, 1) = Enthu(-1, Enthu(1, 1)) = Enthu(Enthu(-1,1),1) = Enthu(k,1) = 1$.

If we suppose now that the full contradiction is a false value ($k<0$) and we aggregate a total truth and twice the total falsity, we obtain, using the associativity and the boundary condition property, and the enthusiastic property described in Proposition 8, we have :

$k = Enthu(1, -1) = Enthu(1, Enthu(-1, -1)) = Enthu(Enthu(1,-1),-1) = Enthu(k,-1) = -1$

5.4.4 Normal contradiction with the total truth or with the total falsity

In the precedent section we study the values of the edges of the unit square for an enthusiastic aggregation operator (see Figure 13). The thick lines indicate the known areas. In the following we are going to study the remaining (lines A, B, C and D), which correspond to a contradiction with the total truth or with the total falsity.

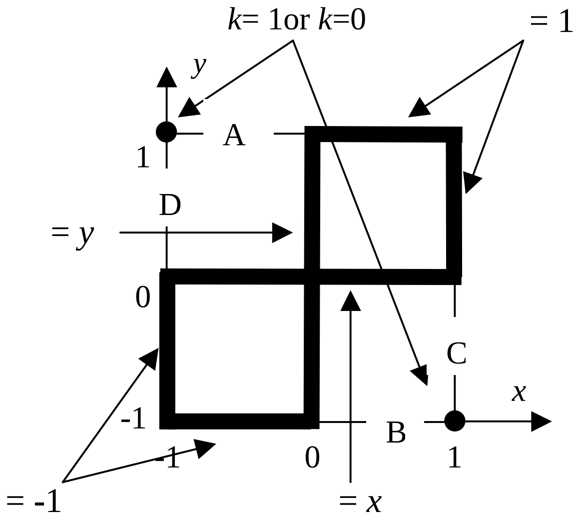


Figure 13. Know points for an enthusiastic aggregation admitting full contradiction

The result shown in Proposition 16 translates the fact that any contradiction with the total truth gives either the total truth or gives the false value.

Proposition 16 : For all $x \in [-1,1]$ *Enthu*$(x,1) = 1$ or *Enthu*$(x,1) = x$.

Proof. Let us call x the result of the aggregation of a truth value z with the total truth :

Enthu$(z,1) = x$. Now, using the associativity we have that,

Enthu$(x,1) =$ *Enthu*(*Enthu*$(z,1),1) =$ *Enthu*$(z,$ *Enthu*$(1,1)) =$ *Enthu*$(z,1) = x$.

Two cases appear :

- $x \geq 0$, then using Proposition 7, we have $x =$ *Enthu*$(x,1) = 1$.
- $x < 0$, then *Enthu*$(x,1) = x$.

Note : Proposition 7 fixes the values for the case where there is no contradiction. So, Proposition 16 is particularly interesting for the contradiction with the total truth.

With this result and using the monotonicity property we can draw Figure 14, where we show the possible construction for the contradiction with the total truth (p being a strictly negative value : $p \in [-1,0[$). It is to notice that we did not assume any hypothesis about the continuity.

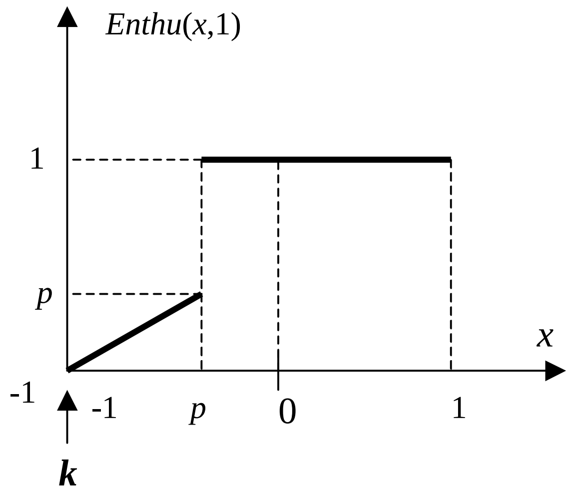


Figure 14. Enthusiastic aggregation of the total truth.

Now we can do an analogous reasoning in order to obtain the behavior of the contradiction with the total falsity. We find in this case that any contradiction with the total falsity gives either the total falsity or the truth value.

Proposition 17 : For all $x \in [-1,1]$ $Enthu(x,-1) = -1$ or $Enthu(x,-1) = x$.

> **Proof.** Analogous as in Proposition 16, but this time we use Proposition 8 instead of Proposition 7.

Taking into account Proposition 17 and Proposition 8, we can draw (Figure 15) the *enthusiastic* aggregation with the total falsity. This time q is a strictly positive value (i.e. $q \in]0,1]$).

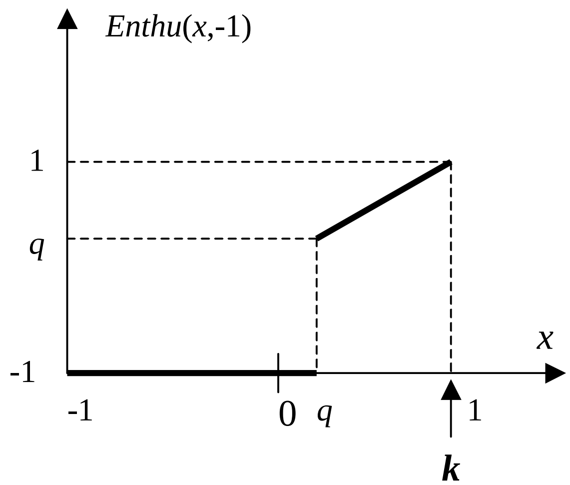


Figure 15. Enthusiastic aggregation of the total falsity.

Taking into account the results of the full contradiction (see previous section 5.4.3) and Proposition 16 and Proposition 17, we immediately see that there is no continuous *enthusiastic* aggregation operator.

Proposition 18 : There is no continuous *enthusiastic* aggregation operator.

Proof. If the full contradiction is not defined, we cannot even talk about continuity. And if it is defined, we know (Proposition 15) that it is either the total falsity or the total truth.

- If the full contradiction k is the total falsity (i.e. $k = -1$), then the enthusiastic aggregation with the total truth is not continuous (for any $p \in]-1,0[$).
- If the full contradiction k is the total truth (i.e. $k = 1$), then the enthusiastic aggregation with the total falsity is not continuous (for any $q \in]0,1[$).

This result points out that the "best" solution for an enthusiastic aggregation operator is the operator continuous enthusiastic presented in section 5.4.2. But taking into account Proposition 18, how can we talk about *continuous* enthusiastic operator ? The answer to this question is that the discontinuities in this case where pushed to the points (-1,1) and (1,-1), where the function is not defined. And if we define them, Proposition 18 applies and we will have discontinuities at the full contradiction points (i.e. (-1,1) and (1,-1)). But the function is continuous everywhere else. Rigorously we should speak about an "almost" continuous enthusiastic aggregation operator. Note that the continuity here is exactly the same as the continuity of the addition on the extended real line (or the continuity of the product on $[0,+\infty]$.

Another simple result relating the contradictions is that if the full contradiction equals the total falsity, then the *enthusiastic* aggregation with the falsity (and in particular the contradiction) equals the total falsity.

Proposition 19 : If $k = -1$ then for all $x \in [-1,1]$ $Enthu(x,-1) = -1$.

Proof. By definition $Enthu(x,-1) \geq -1$. And using the monotonicity

$-1 = k = Enthu(1,-1) \geq Enthu(x,-1)$. Which concludes the proof.

In the same way if the full contradiction equals the total truth, then the *enthusiastic* aggregation with the truth equals the total truth.

Proposition 20 : If $k = 1$ then for all $x \in [-1,1]$ $Enthu(x,1) = 1$.

Proof. By definition $Enthu(x,1) \leq 1$. And using the monotonicity

$1 = k = Enthu(-1, 1) \leq Enthu(x,1)$. Which concludes the proof.

5.4.5 Normal contradiction

The results of section 5.4.4 allow us almost completely to characterize the enthusiastic aggregation operators. In fact using the monotonicity it is trivial to show that any enthusiastic aggregation operator can be represented by Figure 16 or Figure 17, where $p\in[-1,0]$ and $q\in[0,1]$.

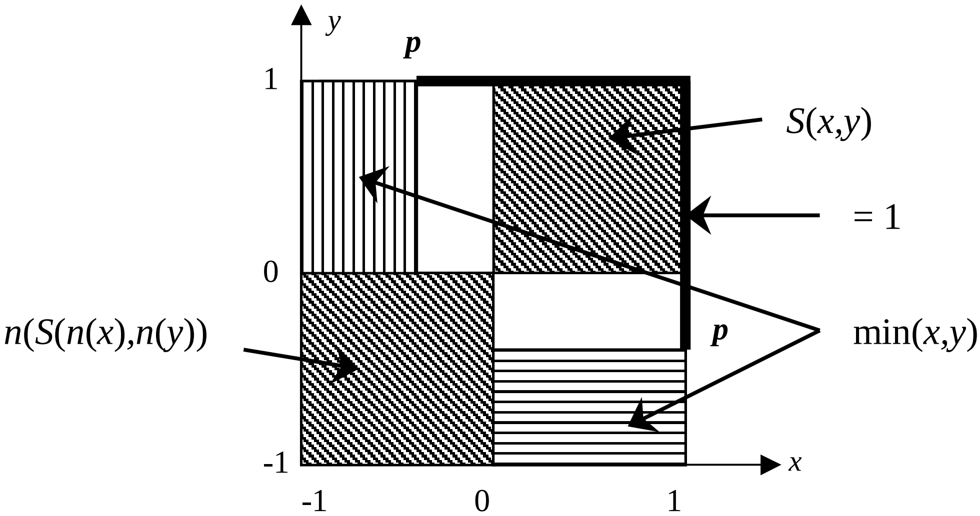


Figure 16. Enthusiastic aggregation with a negative approach for the contradiction.

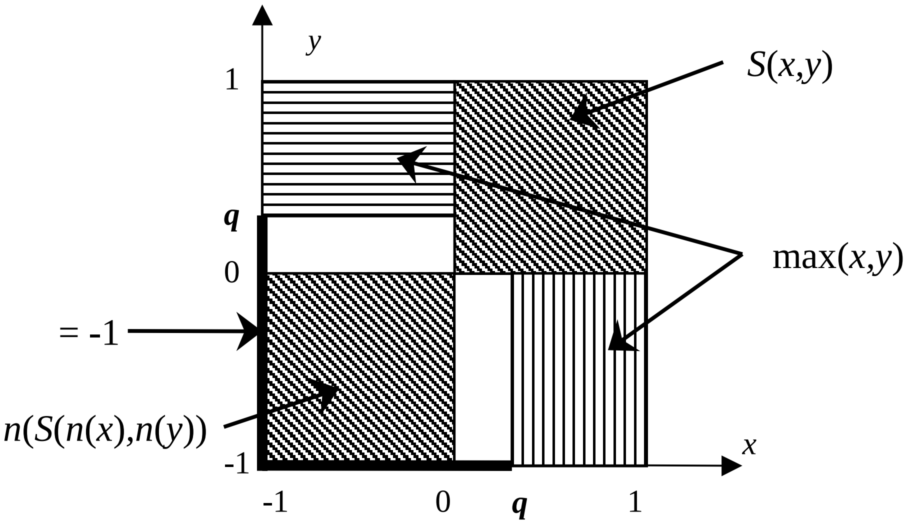


Figure 17. Enthusiastic aggregation with a positive approach for the contradiction.

Figure 17 represents what we called an enthusiastic aggregation with a *positive* approach for the contradiction, since if we presence a contradiction with a truth value strong enough ($\geq q$), then the action of the operator will be to forget the false value taking just the truth one. In the same way, Figure 16 stands for a *negative* approach, since we just take the falsity value when we observe a contradiction with falsity strong enough ($\leq p$).

Note : We recall that if q or p are different from 1, then the operator is not continuous.

5.4.6 Comparison with known operators

Here again we want to compare the enthusiastic operators to the existing ones and we need to do a scale transformation.

The construction of the enthusiastic operator is based on a t-conorm and a negation. So, by simple linear transformation we can obtain a t-norm or a t-conorm :

$$\text{For all } x,y \in [0,1]^2 \quad T(x,y) = -Enthu(-x,-y) \tag{132}$$

$$\text{For all } x,y \in [0,1]^2 \quad S(x,y) = Enthu(x,y) \tag{133}$$

But maybe the most natural transformation is a linear transformation from [-1,1] to [0,1]. In this case we obtain a particular uninorm (see section 2.9) :

$$\text{For all } x,y \in [0,1]^2 \quad U_{1/2}(x,y) = \frac{1}{2} \cdot (Enthu(2x,2y) - 1) \tag{134}$$

This uninorm is particular since the neutral element e equals 1/2 and the underlying t-norms and t-conorms are dual. Note that it is easy to conceive a non linear transformation so that we obtain a uninorm with any neutral element e from an enthusiastic operator.

5.5 Applications

At this point we characterized two operators devoted to the aggregation of truth values (enthusiastic and prudent). But actually, the aggregation of truth values is usually being done by conjunctive operators (t-norms) or disjunctive operators (t-conorms). So, what is the difference between these operators and the introduced construction ? The main difference is that we are not interested in the logical truth value of a logical formula, but in the aggregation of different values of truth for the same proposition. So, in order to compute the truth value of a logical formula of the form "a is A (degree of truth x) AND b is B (degree of truth y)", we use of course a t-norm T : "(a is A AND b is B) (degree of truth $T(x,y)$)". But if we observe first "a is A (degree of truth x)" and then we observe "a is A (degree of truth y)", then we will use a *prudent* or *enthusiastic* aggregation to obtain "a is A (degree of truth $Prud(x,y)$)" or "a is A (degree of truth $Enthu(x,y)$)".

The aggregation presented in this chapter should be used whenever we observe different truth values for the same proposition. A good example when this occurs is the case of the expert systems, where several rules point to the same conclusion. In fuzzy systems it is common to use in this case a t-norm (or a t-conorm), but actually the operator that should be used is a *prudent* or *enthusiastic* operator. Actually, if we examine the handling of uncertainty in non-fuzzy expert systems, we notice that the heuristics used to aggregate the truth values are nothing else than operators of the families presented in this chapter. For instance the medical expert system MYCIN [8] used an almost continuous *enthusiastic* operator, based on the following generator :

$$h(x) = \begin{cases} -\ln(1-x) & \text{for } x \geq 0 \\ \ln(1+x) & \text{for } x < 0 \end{cases} \tag{135}$$

Giving the global aggregation operator (for two arguments) :

$$MYCIN(x,y) = \begin{cases} x + y - (x \cdot y) & \text{if } x \geq 0,\ y \geq 0 \\ \dfrac{x+y}{1-\min(|x|,|y|)} & \text{if } x \cdot y \leq 0 \\ x + y + (x \cdot y) & \text{if } x \leq 0,\ y \leq 0 \end{cases} \tag{136}$$

Note: We would like to recall that this operator was proposed as an *ad hoc* solution based on a more or less probabilistic approach. And we obtained this operator as a particular case in a formal study on aggregation of truth and falsity values. This fact do not justify the ad hoc nature of the solution adopted in MYCIN, but presents a general framework for this kind of aggregation. We are just interested in the combination rule (i.e. aggregation of the truth values obtained by different rules for the same conclusion) and not in the deduction process or in the uncertainty propagation, which are sources of errors (see [33],[34]).

Obviously, the use of enthusiastic (or prudent) operators is not restricted to the expert systems. Their use is suitable in other fields as sensor fusion and decision making. More generally in the case of data fusion, we recommend to use these operators when several sources (expert, sensors, etc.) give different certainty values for the same fact (object, statement, etc.).

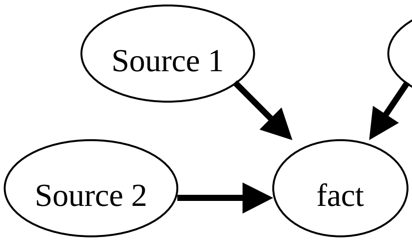


Figure 18. Multiple Source Aggregation

Another interesting case where the prudent and/or the enthusiastic operators are suitable is the case of data received in a temporal framework. Then the revision of the already received data should be performed with the proposed operators. We notice here that the associativity becomes particularly interesting.

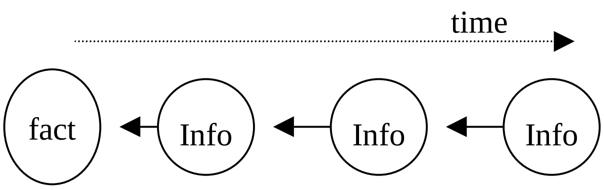


Figure 19. Temporal Data Aggregation

Concluding remarks

This chapter is a first attempt at the formalization of the aggregation of truth values in a non logical calculus way. With necessary conditions (axioms) we characterized two truth aggregation families : the *prudent* and the *enthusiastic*.

The first one has a *cautious* attitude choosing between two observed values the one that is more uncertain. This defines completely the operator, but its behavior seems not to be very interesting in the case of multiple aggregation. Since the general tendency of the *prudent* aggregation is to tend to the uncertainty, so for a lot of aggregations the final result is going to be probably the uncertainty. For instance if we observe a contradiction we are going to get stuck in the total ignorance (consequence of Propositions 1, 4 and 6).

The second one has an *enthusiastic* behavior and will reinforce the result if we observe twice the truth or twice the falsity. When observing falsity and truth the operator gives a compensated value. This seems to be a suitable way to do multiple aggregation, but we should keep in mind the consequences of our choice :

We should not forget that these operators are commutative. In fact we assume that the sources (sensors or rules) play symmetric roles and have the same importance. The use of this type of aggregation is suitable if we are in presence of *high priority sources* as for instance alarm detectors or exception rules.

We have a reinforcement behavior, so we are aware of the fact that if the information of one source (sensor or rule) is used several times we will be reinforced. We also are conscious of the fact that information of specialized sources will be reinforced with information coming from general sources.

It is clear that in order to use this aggregation the sources need to point to the same conclusion and have to be working well. In other words this aggregation method do not support *inconsistent* and *unreliable* sources.

All the precedent inconveniences are central issues in the fusion field. Some solution are presented in section 2.6. We can easily imagine to extend these solutions to our operators.

Finally we would like to finish by pointing out that in Proposition 18, we showed that there does not exist a *completely* continuous enthusiastic operator. This leads to the conclusion that the "best" operator is the *almost* continuous one (130). This particular case is based on an additive generated formula. In the next chapter, we present a proposal for the study and construction of this kind of operator, not based on axioms and mathematical formulas, but on a metaphor allowing intuitional interpretation.

Chapter 6

The Balance Framework

The problem of aggregating fuzzy sets in a meaningful way has been of a central interest since the late 1970s. In most cases (as we have observed in chapter 2), the aggregation operators are defined on a pure axiomatic basis, forgetting usually to give a global intuitive vision of the comportment.

This is why, in this chapter, we present a new way, based on a metaphor, to conceive new aggregation operators. The use of a metaphor offers the possibility to illustrate the mathematical and axiomatic choices, providing in this way an intuitive vision of the behavior of the operator.

The results of the aggregation of truth and falsity values (chapter 5) push us to particularly focus on additively generated operators. The balance metaphor is conceived specifically for this type of operators. We also establish that this new family is not only meaningful, but it is also enough general to include operators of the most common aggregation families presented in chapter 2.

We start this chapter (section 6.1) by establishing the metaphor between the balance model and a mathematical model. We proceed to explain how this new operator can take the form of classical operators (section 6.2). This provides evidence that we have a general framework, that allows us to compare the different existing operators and to create new ones.

In order to compare the global behavior of different operators, in section 6.4, we propose to use the balance framework as a method of visualization. In this way we avoid the problem due to the axiomatisation. In fact, it is generally not obvious, just by comparing sets axioms, to know the effect of their interaction on the global behavior. We first show how to represent the general comportment on the balance model. Then pushing the metaphor further we discover that we can add more information to our first scheme, as for instance the visualization of the sensitivity of the operator, based on the derivative of the constructing functions. And finally we illustrate on this scheme how we can analyze a particular data aggregation, providing a data mining tool.

For the construction matters, in section 6.5 we use the metaphor to guide the construction of an aggregation operator. This methodology is not only interesting because it introduces reflection during each step of the construction, but also because the natural choices conduce to interesting aggregation operators (see section 6.6). One advantage of using the metaphor for putting together an operator is that we do not need to identify the family to know the behavioral properties of the operator. Another advantage is that the operator can be built in a progressive way, where each step has an immediate intuitive consequence.

We conclude this chapter (section 6.7) by presenting concrete application of the balance framework. On the one hand we explain what kind of balance corresponds to the enthusiastic aggregation of truth and falsity. On the other hand we present the use of the metaphor as a starting point for the construction of original operators.

6.1 Construction of the metaphor

The construction of our new mathematical aggregation operator, as we said, is based on the metaphor of a balance. The idea is to establish a strong relationship between a real world object, on which we have natural intuition, with an abstract mathematical formula. In order to achieve this project we will proceed as follows : first we will take a physical model of a real balance. Then with the help of some basic physics and some mathematics we will obtain a mathematical model of the artifact. We will at this point remark that the formula that gives us the total weight is a general form of a lot of aggregation operators.

6.1.1 The physical model

The physical model of a balance we use is shown in the Figure 20, where d_i is the distance of the object i to the fulcrum, m_i his mass, g the acceleration due to the gravity, k is the constant of stiffness of the spring, and finally α is the angle between the lever and the horizon.

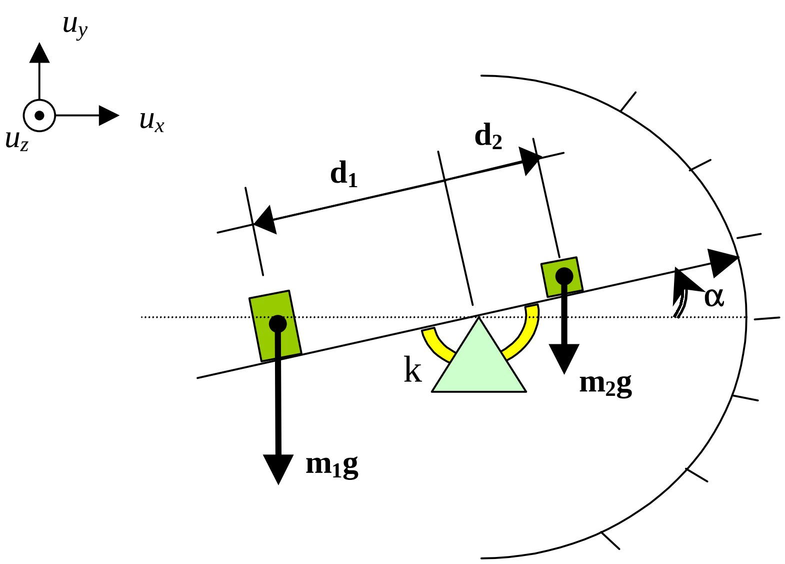


Figure 20. The Balance model

We can now, with the help of physics formulas, translate this scheme of a balance into mathematical equations.

6.1.2 The physics laws

Newton's Second Law says that the sum of the moments (forces) equals zero :

$$m_1 \cdot \vec{g} \times \vec{d}_1 + m_2 \cdot \vec{g} \times \vec{d}_2 + k \cdot \alpha \cdot \vec{u}_z = \vec{0} \tag{137}$$

Executing the vector product, we obtain :

$$\left(m_1 \cdot g \cdot \left\|\vec{d}_1\right\| \cdot \cos(\alpha + \pi)\right) \cdot \vec{u}_z + \left(m_2 \cdot g \cdot \left\|\vec{d}_2\right\| \cdot \cos\alpha\right) \cdot \vec{u}_z + (k \cdot \alpha) \cdot \vec{u}_z = \vec{0} \tag{138}$$

Making the projection on the *z*-axis, we have :

$$\left(m_2 \cdot g \cdot \left\|\vec{d}_2\right\| \cdot \cos\alpha\right) - \left(m_1 \cdot g \cdot \left\|\vec{d}_1\right\| \cdot \cos\alpha\right) + (k \cdot \alpha) = 0 \tag{139}$$

We can suppose without any restriction that α is very small (close to zero). In this case we can approximate $\cos(\alpha) \underset{\alpha \to O}{\approx} 1$ .Then the equation becomes :

$$m_2 g \left\|\vec{d}_2\right\| - m_1 g \left\|\vec{d}_1\right\| + k\alpha = 0 \tag{140}$$

Let e be the position of the fulcrum on the lever and x_i the position of the object i. Then the distance of the object i to the fulcrum is :

$$\left\|\vec{d}_i\right\| = |x_i - e| \tag{141}$$

So we obtain :

$$m_2 g \cdot |x_2 - e| - m_1 g \cdot |x_1 - e| + k\alpha = 0 \tag{142}$$

We remark that

$$x_2 \geq e \text{ and so } |x_2 - e| = (x_2 - e) \tag{143}$$

$$x_1 \leq e \text{ and so } -|x_1 - e| = (x_2 - e) \tag{144}$$

Then the formula (142) becomes :

$$m_2 g \cdot (x_2 - e) + m_1 g \cdot (x_1 - e) + k\alpha = 0 \tag{145}$$

We introduce the oriented distance :

$$\overline{d}_i = (x_i - e) \tag{146}$$

$\overline{d}$ is called in physics the oriented distance, because it gives the value of the distance and it is negative when we are on the left of the fulcrum and positive when we are on the right.

The formula (145) then becomes :

$$m_1 g \cdot \overline{d}_1 + m_2 g \cdot \overline{d}_2 + k\alpha = 0 \tag{147}$$

If we want to read the result of the weight, we only need to know α. So, we solve for that the equation on α :

$$\alpha = -\frac{g}{k}\left(m_1\overline{d}_1 + m_2\overline{d}_2\right) \tag{148}$$

We can easily generalize this result to *n* weights :

$$\alpha = -\frac{g}{k} \cdot \sum_{i=1}^{n} m_i \overline{d}_i \tag{149}$$

Let be $w_i = g \cdot m_i$, the weight of the object *i*, then the equation will be :

$$\alpha = -\frac{1}{k} \cdot \sum_{i=1}^{n} w_i \overline{d}_i \tag{150}$$

We obtained here the equation that computes the value of the angle using the weight and their positions. But what happens if we change the topology of the space ?

6.1.3 The balance equation with topology transformation

If we look closer to what we called the oriented distance, we will see that it is a linear function.

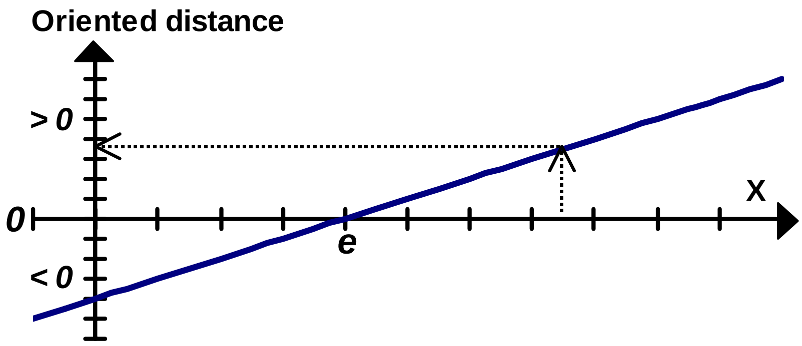


Figure 21 : Oriented distance as a function

This function associates to each point *x*, of the natural scale of the lever, a value of "distance with sign". As we just said it is a linear function, but we can imagine that we want to distort the topology of the scale by using a non-linear function $f : x \rightarrow f(x)$.

Taking this into account, the equation of the balance (150) becomes :

$$\alpha = -\frac{1}{k} \cdot \sum_{i=1}^{n} w_i \cdot f(x_i) \tag{151}$$

Let be $y = \sum_{i=1}^{n} w_i \cdot f(x_i)$, then the previous equation can be written :

$$\alpha = -\frac{1}{k} \cdot y \tag{152}$$

We observe here, once more, a linear function. This operator associates to each point *y*, a value of α. In other words it is the scale drawn on the balance. We can here also decide that the scale will not be linear. If we call $h : y \rightarrow h(y)$, the new non-linear function, the balance equation (152) becomes :

$$\alpha = h\left(\sum_{i=1}^{n} w_i \cdot f(x_i)\right) \tag{153}$$

We observe in formula that the weight *i* is automatically associated to the argument *i*. This does not need to be always true. We can imagine a function σ that associates each weight *i* to a particular argument *i'* = σ(*i*). The equation can be expressed by :

$$\alpha = h\left(\sum_{i=1}^{n} w_{\sigma(i)} \cdot f(x_i)\right) \qquad (154)$$

Here α can be understood as the value read on the weight-scale of the balance. In fact it is the transformation by *h* of the angle between the lever and the horizon. In other words it is the function that gives us the weights on the weight scale. We obtained in (154) the formula that gives us the total weight read on the scale using the single weights as input. And this is exactly the idea of an aggregation operator. We will call this formula the balance equation.

The balance equation is a general form of a mathematical operator; let us now show in the next section that the balance equation is pertinent because it is a general form of many classical operators, presented in chapter 2.

6.2 Particular Balances

We obtained a mathematical formula from a physical model of a balance (154). This formula looked like an interesting aggregation operator, in fact it is the general form of a lot of classical operators. In this section we will try to describe different particular cases of the balance formula that correspond to well-known aggregation operators. For each one of these examples, we recall the usual form of this operator (see chapter 2 for more details). Then we show why it is a particular case of the balance formula (154).

6.2.1 Quasi-arithmetic means

As we saw in section 2.2, the most common way to aggregate information is to use the quasi-arithmetic mean (24), where *f* is a strictly monotone function :

$$M_f(x_1, x_2, ..., x_n) = f^{-1}\left[\frac{1}{n}\sum_{i=1}^{n} f(x_i)\right] = f^{-1}\left[\sum_{i=1}^{n}\left(\frac{1}{n}\cdot f(x_i)\right)\right] \qquad (155)$$

We remark that (155) is a particular case of the balance equation (154), where *f* is a strictly monotone function, *h* is the inverse of *f* (i.e. f^{-1}), all the weights equal $1/n$ and the association of weights is the identity ($\sigma(i) = i$ for all i).

Note : any permutation will also give the same result since all the weights are equal.

Since the balance equation generalizes the quasi-arithmetic mean then it also generalizes all its particular cases :

Operator	**Lever $f(x)$**	**Weight Scale $h(x)$**	**Weights w_i**	**Weight association**
arithmetic mean	x	x	$w_i = 1/n$	any σ
quadratic mean	x^2	$\sqrt[2]{x}$	$w_i = 1/n$	any σ
geometric mean	$\log x$	e^x	$w_i = 1/n$	any σ
harmonic mean	$\frac{1}{x}$	$\frac{1}{x}$	$w_i = 1/n$	any σ
weighted mean	x	x	$\sum_{i=1}^{n} w_i = 1$	any σ

Table 11. Balance operator in the means family.

Note : It is usually claimed that the maximum and minimum are particular cases of the quasi-arithmetic mean. In reality they are limit cases, which means that we can approach them as much as we want, but we are never going to obtain the operator.

6.2.2 t-norms and t-conorms

We already introduced t-norms and t-conorms and worked with them. Please refer to section 2.7 for definitions and chapters 3 and 4 for detailed studies on these operators.

Here we are particularly interested in the Archimedean t-norms and t-conorms (see section 2.7.5), because of their representation based on the additive generator. It has been shown that for every continuous Archimedean t-norm *T*, there exists a continuous decreasing function f such that :

$$T(x_1,\ldots,x_n) = f^{(-1)}\left(\sum_{i=1}^{n} f(x_i)\right) \tag{156}$$

We remark that this representation of an Archimedean t-norm is a particular case of the balance equation (154), where f is a continuous decreasing function $f:[0,1]\rightarrow[0,+\infty]$ satisfying $f(1)=0$.

h being the pseudo inverse (50) of $f^{(-1)}$ defined by :

$$f^{(-1)}(z)=\begin{cases} f^{-1}(z) & \text{if } z\in[0,f(0)] \\ 0 & \text{if } z\in]f(0),+\infty] \end{cases} \tag{157}$$

Or in a more practical form :

$$f^{(-1)}(z)= f^{-1}(\min(z, f(0))) \tag{158}$$

All the weights equal the unity (i.e. $w_i=1$) and the association of weights is the identity ($\sigma(i) = i$ for all i).

For the continuous Archimedean t-conorms, we have an equivalent representation :

$$S(x_1,\ldots,x_n) = g^{(-1)}\left(\sum_{i=1}^{n} g(x_i)\right) \tag{159}$$

But this time the function f of the balance equation (154) corresponds to a continuous function $g:[0,1]\rightarrow[0,+\infty]$ satisfying $g(0)=0$.

The function h corresponds now to the pseudo-inverse $g^{(-1)}$, defined by *:*

$$g^{(-1)}(z)=\begin{cases} g^{-1}(z) & \text{if } z\in[0,g(1)] \\ 1 & \text{if } z\in]g(1),+\infty] \end{cases} \tag{160}$$

Or in a more practical form :

$$g^{(-1)}(z)= g^{-1}(\min(z, g(1))) \tag{161}$$

Again all the weights equal the unity (i.e. w_i =1) and the association of weights is the identity ($\sigma(i) = i$).

Note : any permutation σ will also give the same result for the t-norms or the t-conorms, since all the weights are equal.

In the following table we present some examples of continuous Archimedean t-norms and t-conorms and the different aspects for the construction of a balance.

Operator	**Lever $f(x)$**	**Weight Scale $h(x)$**	**Weights w_i**	**Weight association**
Probabilistic t-norm	$-\ln(x)$	e^{-x}	$w_i = 1$	any σ
Probabilistic t-conorm	$-\ln(1-x)$	$1-e^{-x}$	$w_i = 1$	any σ
Lukasiewicz t-norm	$1-x$	$1-\min(x,1)$ = $\max(1-x,0)$	$w_i = 1$	any σ
Lukasiewicz t-conorm	x	$\min(x, 1)$	$w_i = 1$	any σ
Yager t-norm	$(1-x)^p$	$1-\sqrt[p]{\min(x,1)}$ = $\max\left(1-\sqrt[p]{x},0\right)$	$w_i = 1$	any σ
Yager t-conorm	x^p	$\sqrt[p]{\min(x,1)}$ = $\min\left(\sqrt[p]{x},1\right)$	$w_i = 1$	any σ

Table 12. Balance operator in the continuous Archimedean t-norm and t-conorm families.

6.2.3 Ordered Weighted Averaging Operators (OWA)

In section 2.4, we presented the family of aggregation operators called the Ordered Weighted Averaging Operators (30) :

$$OWA(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} w_j x_{\sigma(j)} = \sum_{i=1}^{n} w_{\sigma^{-1}(i)} x_i \qquad (162)$$

where $\sum_{i=1}^{n} w_i = 1$, and σ is the permutation that orders the elements : $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \cdots \leq x_{\sigma(n)}$.

We remark that the equation of an OWA operator (162) is a particular case of the balance equation (154), where h and f are identity equations : $f(x) = h(x) = x$. The permutation σ is simply the way we associate the weights to the arguments. Here we associate the first weight (w_1) to the argument placed at first on the lever, looking from the left to the right (i.e. $a_{\sigma(1)}$). The second weight (w_1) will be associated to the next argument on the right of the first one ($a_{\sigma(2)}$), and so on.

Since the OWA operator is a particular case of the balance equation, all the following operators are particular cases of the balance :

Operator	**Lever $f(x)$**	**Weight Scale $h(x)$**	**Weights w_i**	**Weight association**
the minimum	x	x	$\begin{cases} w_1 = 1 \\ w_i = 0 \quad \text{if } i \neq 0 \end{cases}$	σ is the permutation that orders the elements : $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \cdots \leq x_{\sigma(n)}$
the maximum	x	x	$\begin{cases} w_n = 1 \\ w_i = 0 \quad \text{if } i \neq n \end{cases}$	
k-order statistics	x	x	$\begin{cases} w_k = 1 \\ w_i = 0 \quad \text{if } i \neq k \end{cases}$	

Table 13. Balance operator in the OWA operator case

6.2.4 Symmetric Sum

In section 2.3, we presented an interesting aggregation family characterized by the property of self-duality (26). We recall this property :

$$S(x_1, x_2, \ldots, x_n) = 1 - S(1 - x_1, 1 - x_2, \ldots, 1 - x_n) \qquad (163)$$

Equation (163) is translated in the balance framework by the fact that we would like the operator to act the same way if we invert the scales. A sufficient condition for this is first to have the same scale on the left and on the right of the lever. And second to have the same scale on the lower and higher part of the weight scale.

Since in the case of symmetric sums we are interested in aggregation in [0,1], the fulcrum should be placed at 1/2 (for symmetry reasons). And then the first condition is translated mathematically by :

$$f\left(\frac{1}{2} + x\right) + f\left(\frac{1}{2} - x\right) = 0 \qquad (164)$$

The second condition, corresponding to the symmetry of the weight scale, should translate the symmetry (based on the angles) of the scale, with respect to α =0. This is translated by :

$$h(\alpha) + h(-\alpha) = 1 \qquad (165)$$

These two conditions are sufficient to obtain a symmetric sum with a balance.

In fact, 1-h(f (1-x)+ f (1-y)) = 1-h(f (1/2+(1/2-x))+ f (1/2+(1/2-y)))
= 1-h(- f (1/2-(1/2-x)) - f (1/2-(1/2-y)))
= 1-h(-(f (x))+ f (y)))
= 1- (1 - h(f (x))+ f (x)))
= h(f (x))+ f (x))

Note : conditions (164) and (165) are more general than condition (29) presented by Silvert in [78] in a quasi-arithmetic mean context.

6.2.5 Generated Uninorms

Very interesting operators are the uniform aggregation operators (uninorms) (see section 2.9). But as already remarked, they are in general non continuous. In [43] Fodor, Yager and Rybalov proposed an almost continuous solution : the generated uninorms (see section 2.9.4) :

$$U(x_1,\dots,x_n) = g^{-1}\left(\sum_{i=1}^{n} g(x_i)\right) \tag{166}$$

It is clear that equation (166) is a particular case of the balance equation (154), where the lever function f is a strictly increasing continuous function $f = g : [0,1] \rightarrow [-\infty,+\infty]$, so that $g(0)= -\infty$, $g(1)= +\infty$. The weight scale function h is the inverse function of g (i.e. $h=g^{-1}$). The weights are all equal to one, and so any weight association σ gives the same result.

We remark that generated uninorm operators were already introduced as an interesting class of aggregation operators by Klement, Mesiar and Pap in [54], and were called the **associative compensatory operator** (see (55) in section 2.8). The authors based the construction of the function g on two additive generators, one of a continuous Archimedean t-norm (function t) and one of a continuous Archimedean t-conorm (function s). Then the lever function becomes (e being placed at the fulcrum) :

$$f(x) = g(x) = \begin{cases} -t\left(\dfrac{x}{e}\right) & \text{if } x \le e \\ s\left(\dfrac{x-e}{1-e}\right) & \text{if } x \ge e \end{cases} \tag{167}$$

We recall that t is a strictly decreasing continuous function satisfying $t(1)=0$. And s is a strictly increasing continuous function satisfying $s(0)=0$.

We can compute the inverse function in order to obtain the weight scale function h *:*

$$h(\alpha) = g^{-1}(\alpha) = \begin{cases} e \cdot t^{-1}(-\alpha) & \text{if } \alpha \le 0 \\ e + (1-e)\cdot s^{-1}(\alpha) & \text{if } \alpha \ge 0 \end{cases} \tag{168}$$

In the balance framework we can interpret equations (167) and (168), by saying that the idea is to build a t-norm for the elements put on the left side of the lever (with respect to the fulcrum e). If we actually put some elements there, the balance is going to point to the upper part of the weight scale ($\alpha \ge 0$). Since we want a t-norm, we need in this part the corresponding operator (i.e. the inverse function for the t-norm). The division and multiplication by e are introduced in order to obtain a full t-norm on $[0,e]$. The same reasoning can be made for the right side of the lever and for the lower side of the weight scale.

6.2.6 Monotonic Identity Commutative Aggregator

In the previous sections we presented well known families. Here we would like to describe an interesting operator introduced by Yager and Kelman in [53]. This aggregation operator was introduced after a long study in order to obtain the best operator for a real operational problem and as an illustration of what the authors called Monotonic Identity Commutative Aggregator. It was defined by :

$$K(x_1, x_2, \ldots, x_n) = \max\left(0, \min\left(1,\ k + \sum_{i=1}^{n} (x_i - k)\right)\right) \qquad (169)$$

It is noticed by the authors that if k =0, we obtain the Lukasiewicz t-conorm and if k =1, we obtain the Lukasiewicz t-norm. Note that this operator was also introduce by Mesiar in [68] as combination of Lukasiewicz t-norm and t-conorm.

This time the relationship with the balance operator is not so obvious, but if we consider the functions :

- $f(x) = x - k$.
- $h(x) = \max(0,\min(1,x+k))$

We obtain :

$$K(x_1, \ldots, x_n) = h\left(\sum_{i=1}^{n} f(x_i)\right) \qquad (170)$$

And now it is obvious that it is a particular case of the balance equation (154), where all the weights equal the unity (i.e. w_i =1).

In the balance framework we can see that the lever function f is nothing else than a normal linear scale (recall (150)). And the weight scale h is also a linear scale that blocks on 1 for total weights higher than 1 and for total weights smaller than 0 (see Figure 22). It is also to notice that the weight scale is built so that the empty balance shows the weight k (the neutral element).

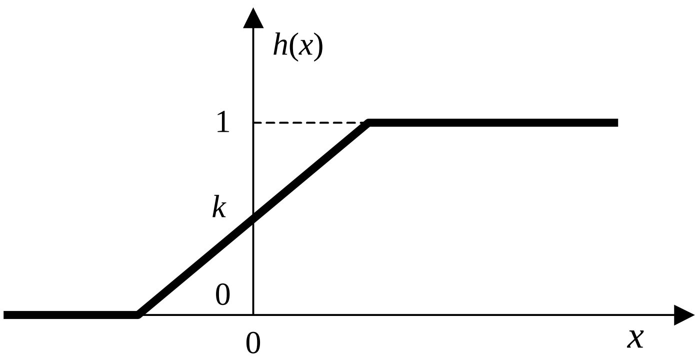


Figure 22. Weight scale h for Kelman's operator.

6.2.7 Other operators

Precedent sections have shown that the balance equation (154) covers a wide range of aggregation operators, where we find operators going from the t-norm family to operators like the quasi-arithmetic means, going through the OWA operators or even some uninorms. But we do not pretend here that we generalize every single family. In particular the presented balance model do not include the most important group of the fuzzy discrete integrals, the aggregators that take into account interaction. In fact, in the balance model, we clearly associate a weight to each argument, while in the fuzzy integrals we associate weights to the groups of arguments. However, we can imagine that an extended balance model will describe the association of weights to several arguments. This will lead to include most likely the Choquet integrals, since they are in an additive form. But certainly the Sugeno integrals are too different to be included under this model. We may just think about the weighted means.

It is to notice that a lot of work is currently being done on particular mathematical aspects relative to operators of the form (154) or some particular cases (mostly where $h = f^{-1}$) see [60]. For instance on conditions on the function so that the operator is associative (see [10]).

In the next section 6.3, we show that the balance equation can easily be put under a product form and so be used for all product based aggregation operators.

6.3 The balance equation in a product form

All the operators presented in the previous sections of this chapter are written in an additive form (154). Here, we show that the balance can also be put in a product form (171) :

$$\beta = \chi\left(\prod_{i=1}^{n} (\varphi(x_i))^{w_{\sigma(i)}}\right) \qquad (171)$$

In order to make the proof simpler, we start by showing the relation for *n*=2 and all weights equal to1.

Let *Aggreg* be an aggregation operator of the form :

$$Aggreg(x, y) = \chi(\varphi(x) \cdot \varphi(y)) \qquad (172)$$

And let us define now the function *f* by :

$$f(x) = -\log(\varphi(x)) \qquad (173)$$

So, we have :

$$\varphi(x) = \exp(-f(x)) \qquad (174)$$

Let define the function *h* by :

$$h(x) = \chi(\exp(-x) \qquad (175)$$

If we take now equation (172) and we replace the function φ by a trivial form :

$$\varphi(x) = \exp(\log(\varphi(x))) \qquad (176)$$

Equation (172) becomes :

$$Aggreg(x, y) = \chi(\exp(\log(\varphi(x))) \cdot \exp(\log(\varphi(y)))) \qquad (177)$$

Equation (172) becomes :

$$Aggreg(x, y) = \chi(\exp(\log(\varphi(x)) + \log(\varphi(y)))) \qquad (178)$$

If we use now the definition (173)of the function *f*, equation (172) becomes :

$$\begin{aligned} Aggreg(x, y) &= \chi(\exp(-f(x) - f(y))) \\ &= \chi(\exp(-(f(x) + f(y)))) \end{aligned} \qquad (179)$$

If we now use the definition of *h* (175), equation (172) becomes :

$$Aggreg(x, y) = h\big(f(x) + f(y)\big) \qquad (180)$$

We obtained the classical formulation of the balance equation, in the particular case of two elements (*n*=2) and all the weights equal to 1. The aggregation operator *Aggreg* is associative, so the generalization of this demonstration to *n* elements is trivial. The problem with the weights can be solved doing once again the demonstration, but this time with weights, and using the property of the logarithm :

$$\log(a^b) = b \cdot \log(a) \qquad (181)$$

We have in this way shown that all the functions of the form (171) take the form of the balance equation (154) with the lever function *f* defined by (173) and the weights scale function *h* defined by (175).

In other words we have shown that not only all operators defined on an additive form can exploit the metaphor, but also all the operators defined on a product basis.

6.4 The visualization of Aggregation

Aggregation operators are usually defined by mathematical and axiomatic choices. This method of construction and study allows irrefutable reasoning, but very often we forget that axioms are on the basis of the construction and that their influence (of each one or their combination) on the final operator is not always obvious.

This is why we propose to use the balance framework as a method of visualization that allows us to draw on a scheme the global behavior of an operator. With this global vision, we will automatically obtain an intuitional understanding of the operator and its global comportment.

In this section we show how we build the scheme of the general comportment. Then we see that we can add more information to our scheme, as for instance the visualization of the sensitivity of the operator. And finally we illustrate how the construction of the scheme can be used to visualize a particular data aggregation.

6.4.1 The visualization of the general comportment

As we insisted before, the visualization is fundamental for the good understanding of the global behavior of an operator. If we want to visualize (with our metaphor) a particular aggregator, at first we have to show that we can put it on the balance equation form.

While establishing the analogy between the physical balance model and the balance equation (154) we introduce the notion of a topology transformation (see section 6.1.3). This concept emerges with the apparition of the functions f and h. Let us analyze first the function f.

The topology transformation by *f*

As we already said, the function f associates to each position x, of the natural scale of the lever, a new value, in other words a new position. In order to visualize the action of this functions, we can draw the transformation of a linear scale done by the function f. So we project, using f, a constant step from the x-axis to the y-axis. This way, we obtain the new-scaled lever on the y-axis. (See Figure 23)

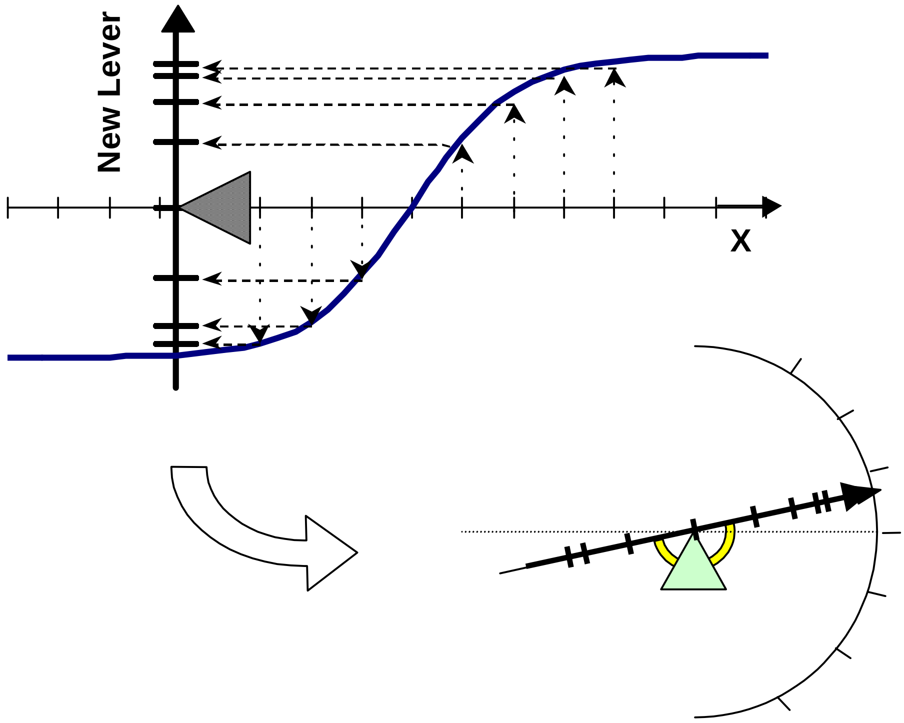


Figure 23. Scale transformation induced by *f*

What we actually do is the topological transformation of a linear scale by *f*. We can now translate this action in mathematical terms and we will obtain a direct procedure to draw the topological transformation. In order to visualize the action of the topology transformation *f* on the interval [*a*,*b*], on a *n* point scale, we need only to draw on an oriented axis the points :

$$\forall i \in \{1 \ldots n\} \; f_i = f\left(a + (b-a)\frac{i}{n} \right) \qquad (182)$$

In order to actually draw a balance, we need to re-scale the points to the size of the drawn lever. For a lever of length $2 \cdot L$ and where the fulcrum is placed at the zero point of the axis, we re-scale the points by using :

$$2 \cdot L \cdot \frac{f_m - f_i}{f_m - f_M} - L \qquad (\forall i \in \{1 \ldots n\}) \qquad (183)$$

where f_m is the lowest value attempted by the function on the *n* points and f_M is the highest one. In the case when *f* is monotone, $f_m = f(a)$ and $f_M = f(b)$ for the increasing case and $f_m = f(b)$ and $f_M = f(a)$ for the decreasing one.

When the scale is already constructed, we suggest to show the value *x* on its new position (i.e. *f*(*x*)). In this way it is clear where each value is located (see Figure 24).

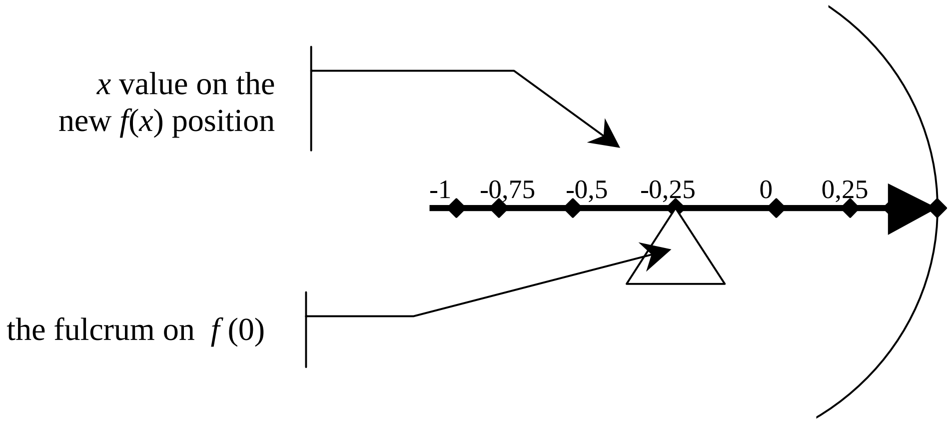


Figure 24. A balance with a topological transformed lever

The topology transformation by *h*

If we look now at the function *h*, we will observe that it transforms the angle between the lever and the horizon into the value read on the weight-scale of the balance. Once more *h* transforms the natural linear scale onto a new one. That is the reason why we talk about topology transformation.

However here the construction of the new scale is a bit different. The scale we built with *f* was constructed in order to illustrate the new positioning of arguments separated by a constant step. Here the new weight scale has to be built in order to show the right total aggregation with the natural functioning of the lever. This last condition is fundamental if we want to preserve our intuition on the balance model. To be more precise, we want that, when the lever makes an angle α with the horizon, the weight shown is $h(\alpha)$. We also expect that the numerical variation of the points appearing on the weight scale remains linear. In other words we want to deform a linear scale of $h(\alpha)$ and put it on the α-space. To do so, we use a similar method as for *f*, but this time we will project a regular step from the *y*-axis onto the *x*-axis, and then distort the obtained scale into a part of a circumference (see Figure 25).

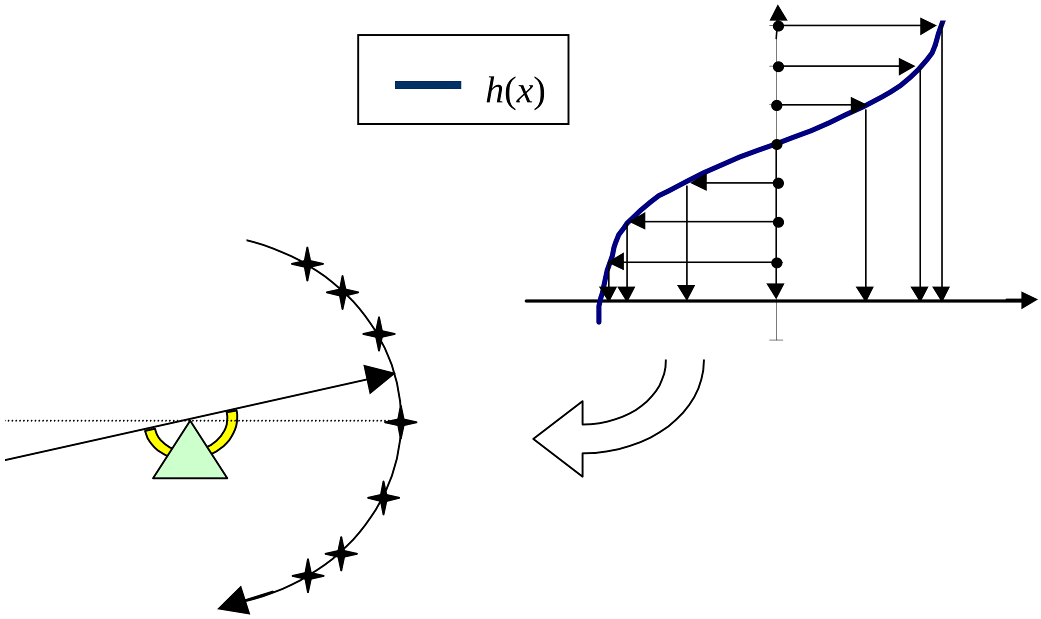


Figure 25. Scale transformation induced by *h*

What we actually do is the projection of a linear scale by the inverse function of *h* (i.e. h^{-1}). We remark that very often the aggregation operators require $h = f^{-1}$, which makes the calculus trivial because we already know *f*. For the remaining cases, we have almost always a continuous strictly monotone function *h*, so the inverse function can be determined. And if it is not the case, the projection can generally be done by doing a piecewise treatment.

The mathematical way to obtain the action of the topology transformation *h* on the interval [*a*,*b*] with a *n* points scale, is to draw the *n* points on an arc of circumference of rayon R :

$$\forall i \in \{1 \ldots n\}\ h_i = h^{-1}\left(a + (b-a)\frac{i}{n}\right) \tag{184}$$

In other to draw the scheme we need to place these *n* points on a circumference of rayon *L*, corresponding to the size of the lever. We place the *n* points by characterizing their (*x*,*y*) coordinates. Let h_m be the lowest value attempted by the function on the *n* points and h_M be the highest one. And let be $H = \max(|h_M|,|h_m|) \times sign(\max(|h_M|,|h_m|))$.

Then we use the formula :

$$\begin{cases} x_i = L \cdot \cos\left(\dfrac{-\pi}{4} \cdot \dfrac{h_i}{H}\right) \\ y_i = L \cdot \sin\left(\dfrac{-\pi}{4} \cdot \dfrac{h_i}{H}\right) \end{cases} \quad (\forall i \in \{1 \ldots n\}) \tag{185}$$

When the scale is already constructed, we suggest to put the value $a + (b-a)\frac{i}{n}$ next to its new h_i -position, in order to show the total aggregation value that will be computed for this position of the lever. (See Figure 26)

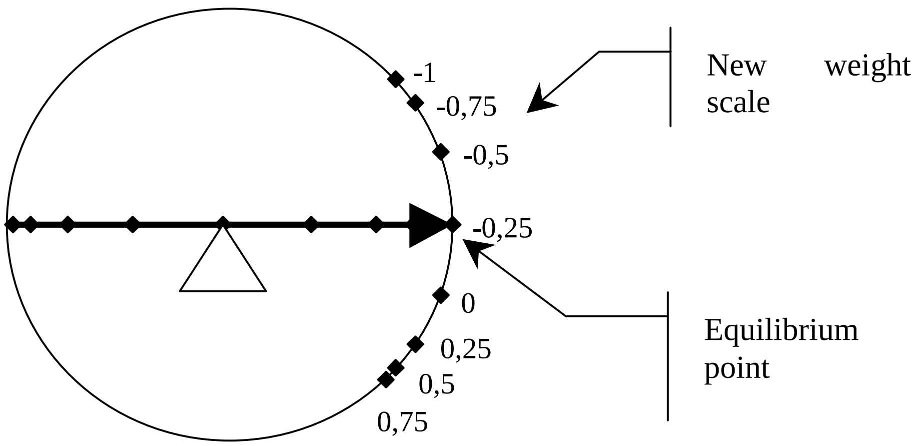


Figure 26. A balance with a topological transformed weight scale

We have presented here the mathematical formulas that allow us to construct a balance model with its topologically transformed scales. This visualization is fundamental for the good understanding of an operator. Nevertheless another useful tool is the visualization of the sensitivity.

6.4.2 Visualization of the Sensitivity

In order to understand the notion of stability we will first focus our attention on Figure 23 : Scale transformation induced by *f*. The function *f* associates to each position *x*, a new position *f*(*x*). If we analyze more precisely the form of the function, we remark that the steeper the function *f*, the less serried the lever scale. This means that, for example, a small variation of *x* in a region where *f* is steep will produce an enormous variation in the aggregation. A simple way to quantify this "steepness" and so the sensitivity, is to use the derivative of *f*. In Figure 27, we have drawn on the same *x*-axis the function *f* and his derivative function. We see that where *f* is steep the derivative function (sensitivity) is high.

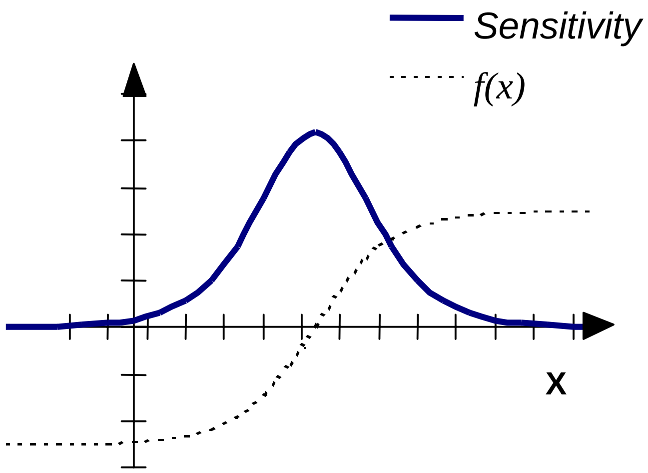


Figure 27. Derivative of *f* shows the sensitivity

Now, we can directly study the sensitivity of the aggregation operator using the derivative functions : where the value of the derivative is high, the sensitivity is high.

Let us now show the way of building the visualization of the lever scale *f* and of the weight scale *h*.

Sensitivity of the lever scale.

The derivative of *f* quantifies the sensitivity of the lever for a small variation of the value of an argument (x_i), in other words the intensity of the translation applied to a weight on the lever by the topology transformation because of this variation.

The simplest way to visualize the sensitivity of *f* on the interval [*a*,*b*] is to draw the derivative on this interval. But we do not have then any reference to the position of the fulcrum and to the lever. So we propose to draw the derivative f' on the deformed lever. To do this we draw the parametric function (variable *t*) :

$$\begin{cases} x = f(t) \\ y = f'(t) \end{cases} \qquad (186)$$

We can then place the derivative on the balance model. (See Figure 28)

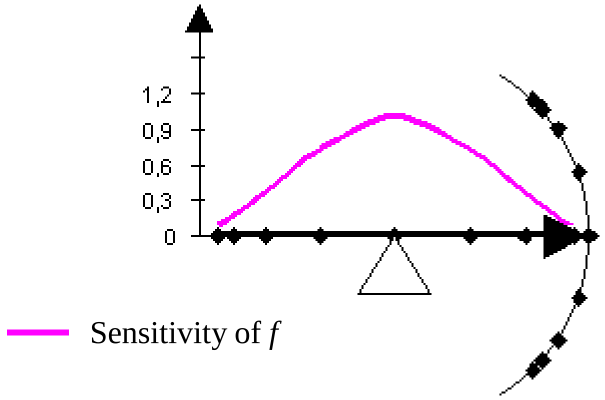


Figure 28. A balance with the sensitivity of his lever

We obtained then a deformed lever and its derived function. The transformed scale shows the new position for each point and the derivative shows for each point the sensitivity with regard to a variation of this value. We have to keep in mind that the translation of a weight on the deformed lever is a variation of $f(x)$ and not of x.

Sensitivity of the weight scale

The derivative of h quantifies the sensitivity for a small variation of the total aggregation. The simplest way to visualize the sensitivity of h on the total aggregation interval $[a,b]$ is to draw the derivative on the interval $h^{-1}([a,b])$. We propose to draw the derivative h' on a vertical rectilinear transformed scale next to the balance weight scale. To do so, we draw the parametric function (variable t) :

$$\begin{cases} x = h'(t) + R \\ y = h(0) - h(t) \end{cases} \qquad (187)$$

The translation $\begin{cases} +R \\ h(0) \end{cases}$ has been introduced to position at the right place the rectilinear scale.

We will obtain then, a re-scaled weight scale and next to it the derived function. The transformed scale shows the total aggregation values for each position of the lever and the derivative shows for each point the sensitivity of a variation on this area. (See Figure 29).

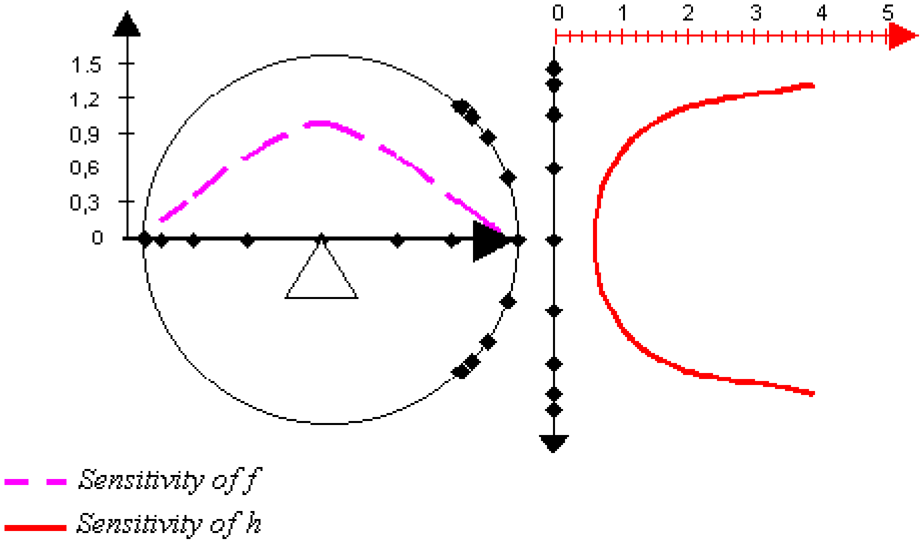


Figure 29. A balance with the sensitivity of the lever and of the weight-scale

Figure 29 sums up the visualization aspects presented in the previous sections, by showing in one figure the general comportment and the sensitivity. Now, we propose to take a look at the visualization of a particular aggregation.

6.4.3 The visualization of a particular aggregation

We talk of a particular aggregation, when we have a specific set of values to be aggregated, and also their weights, the aggregation formula and therefore the aggregated value. We assume that the aggregation operator can be put in the balance equation form (154) or (171), so that we can construct a balance model, as we indicated before. What we want to prove here is that the positioning of the specific values on the model can be useful in the understanding of the aggregation. The idea is to place the weights on the balance and to analyze the resulting physical situation.

Again if we want to actually draw a scheme, we need first to compute the inclination of the lever caused by the placement of the weights (i.e. the angle α). In order to compute this angle we use the balance equation without the topology transformation h (i.e. the function that gives us what we read on the scale). And we normalize the angle so that in the extreme case we obtain an angle of $\pm\frac{\mu}{4}$. Using the notations introduced before, the normalized angle will be :

$$\alpha = \frac{-\pi}{4} \cdot \frac{\sum_{i=1}^{n} w_{\sigma(i)} \cdot f(x_i)}{H} \qquad (188)$$

With this angle the lever will point at the total aggregation value on the transformed weight scale. The balance has now the right position, but it does not have any object on. The weights can be represented by a rectilinear line of a length proportional to the weight associated to the argument. This object will be placed with the help of the function *f*, in the same way as explained before. Taking into account the rotation of the lever, the position of the object $(w_{\sigma(i)}, x_i)$ will be :

$$\begin{cases} f(x_i) \cdot \cos(\alpha) \\ f(x_i) \cdot \sin(\alpha) \end{cases} \qquad (189)$$

Using formula (189), we can draw all the weights that have been used for the particular aggregation, and look at their distribution. We obtain a graph like the one shown on Figure 30.

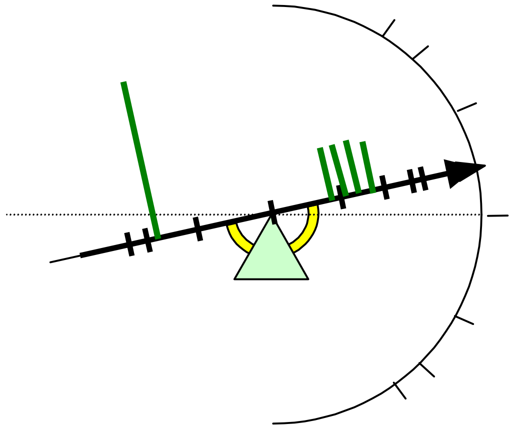


Figure 30. A balance with the sensitivity of the lever and of the weight-scale

On this particular example we observe that there is only one argument that is on the left of the fulcrum and all others on the right. We note that the weight of this argument is so important that the lever balanced onto the left. We understand that there is a strong argument against a lot of small arguments. Subsequently, it may be interesting to take a closer look at this "big" argument, because it can be for example an error.

We can also look at the sensitivity of the place where it is located, and in this way we will know if a small variation of this value will change a lot (or not) his position. We can also take a look at the sensitivity of the area of the total weight, this time we will know if a small variation of any of the arguments will change a lot the total aggregation.

6.5 Building an Aggregation Operator with a Balance

We started the chapter by establishing the metaphor between a balance and a mathematical aggregation operator (154). We showed that not only this new family is meaningful, but also it is enough general to include operators of the most common aggregation families and their specific comportment. Then we took advantage of the metaphor to study the operators of this form, by developing some visualization methods.

In this section, we present a methodology, based on the metaphor, to conceive new aggregation operators. The use of a metaphor offers the possibility to illustrate the mathematical and axiomatic choices, providing an intuitive methodology for the construction of an aggregation operator.

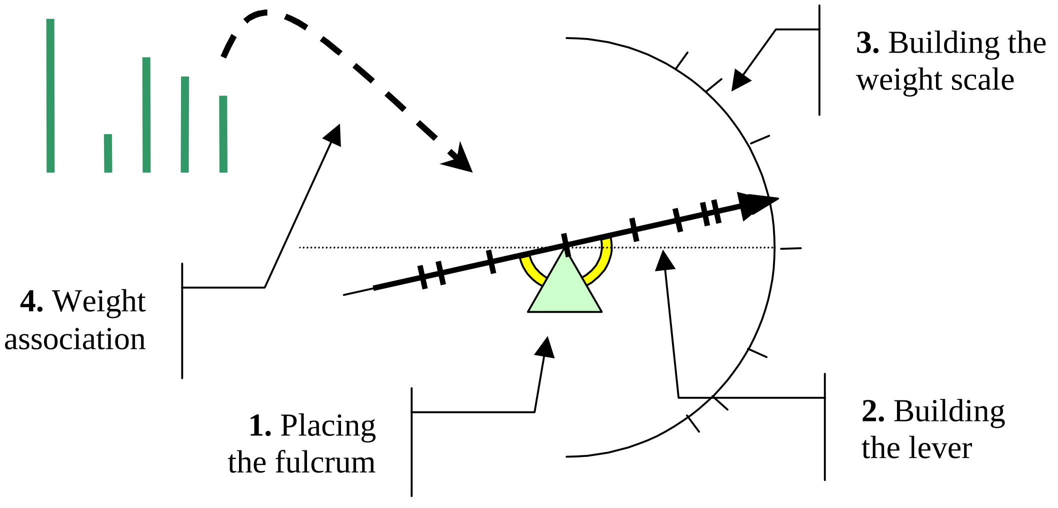


Figure 31. Steps to build a balance

The idea of the methodology is to follow the natural way of building a balance (see Figure 31). We will first place the fulcrum, which corresponds in some situations to the neutral element. Then we will put the lever and its scale by fixing the function *f*. The construction of the total-weight will be done by the choices of the *h* function. We will finish by discussing how to associate the weights to the different arguments.

6.5.1 Placing the fulcrum : the *neuter* element.

The first thing to do is to place the *neuter* element *e*, which corresponds in our metaphor to place of the fulcrum. The *neuter* element is a value associated to an argument that should not have any influence in the aggregation. Once the *neuter* element is fixed, we remark that if we have two arguments on the same side (both on the left of *e* or on the right) we have a reinforcement comportment. But if they are disposed on both sides we will have a counterbalance behavior. So the position of the *neuter* element not only implies the *neuter* value but also the distribution of negative and positive reinforcement areas. Once the value of *e* is fixed, then the next step is to choose the function *f*.

6.5.2 Constructing the lever : the function *f*.

Fixing the image of the *neuter* element : The first thing that we have to do is to take into account our previous choice (the *neuter* element). Then, taking into account our notation, the function *f* we choose has to equal zero for the *neuter* element *e* :

$$f(e) = 0 \qquad (190)$$

The form of the function : Now that we fixed one point, we have to choose the form of the function *f*. We know that *f* performs a topological transformation on the scale of the lever. We can interpret the action of this function, saying that it positions the arguments on the balance. In other words, the form of this function will change the placement of the weights. To illustrate this, let us consider for example that *f* is constant on an interval, then all arguments on this interval will be placed at the same position. Another important aspect related to the form is that the farther the elements are placed from the *neuter* element the more influence they will have. An element is placed far if the value of the function *f* at this point is high.

The steepness : If we analyze more precisely the form of the function, we will remark that the less the function *f* is steep in a region, the more the elements in this region will be placed in a serried way. This means that, for example, a small variation of *x* in a region where *f* is steep will produce an enormous variation in the aggregation. A simple way to quantify this "steepness" and so the sensitivity, is to use the derivative of *f*. For more details about this particular aspect, see section 6.4.2. In mathematical terms, when the derivative of *f* is high then the sensitivity is high. To illustrate this, let us consider the limit case, when the value of the derivative is zero. This implies that the function *f* is constant. We have seen that in this case all arguments are placed at the same position.

The monotonicity : Taking into account the fact that *f* is a topological transformation of a scale, a natural choice is to consider that the function *f* is monotone. This means that you will have a gradual comportment : the more you move away (or approach) an argument from the *neuter* element, the more (or the less) it will have an influence on the result. This characteristic seems for some authors as compulsory for the aggregation operators [58] and appears in their definitions (axioms). But after all, we may decide that we do not need this property. This may be interesting if, for example, we want that after a certain threshold the influence of the arguments decrease, because for instance the arguments are too extreme. We can perform this behavior by changing at this point the tendency of the monotonicity.

The continuity : Another natural choice is to use a continuous function. This means that you will have a regular positioning on the lever. Indeed the continuity translates the fact that if you change a little an argument, its position will change just a little, without jumping. Again we may decide that, after a threshold, the arguments have to be placed at another location on the lever. This can be performed by a discontinuity at the threshold. An example of such type of functions is a step function. In this case groups of arguments are placed on precise positions.

6.5.3 The weight scale : the function *h*.

We can interpret the action of the function *h*, saying that it associates to the angle between the lever and the horizon the value read on the weight-scale of the balance.

Weight of an empty balance : Taking into account that we fixed the *neuter* element, we should now choose the value we would like to obtain if the aggregation is performed with only *neuter* elements. This can be interpreted as choosing the value that will be showed by an empty balance. Mathematically this is translated by the choice of the value of $h(0)$.

The form of the function : After we fix this point we must decide the form of *h*. This function performs a topological transformation on the total-weight scale, in the same way as *f* does on the scale of the lever. So, we can now apply all the remarks for *f* to *h*, taking into account that this time the function *h* acts on the aggregated value of all weights and not on the individual weights as *f* does. For instance, if *h* is constant on an interval, this will mean that the total weight on this interval will not change. More generally, if we now consider the sensitivity we observe that, on a point where the value of the derivative of *h* is high, a small variation of *any* of the weights will produce a significant variation of the total-weight.

The monotonicity : Here again it is natural to choose a monotone function. But if we do not, we will observe some strange behaviors : for example if we keep increasing the weight of one of arguments then the total weight may change its tendency (from increasing to decreasing or vice-versa). This may be interpreted by saying that we consider that after a threshold the total-weight is considered extreme and we decrease its value.

Note that if we use non monotone functions we may obtain non monotone aggregation operators. Even if this maybe need in particular case, this does not fit anymore with our set of axiom for an aggregation operator (see section 1.1)

The continuity : Another natural choice is the continuity. But a discontinuity can be interpreted as a jump on the total weight scale. For example a step function implies that the balance round off the final weight.

The global behavior : Since the function *h* arrives at the end of the construction we may have some global behavior. For instance we may be interested in the aggregation of a single element *a* (or its equivalent : the aggregation of an element with a *neuter* element). It seems natural to expect then to have a total weight equal to *a*. For some authors [58] this seems so fundamental that it is included in their definition of an aggregation operator. But this constraint has strong consequences. Mathematically we have :

$$\forall a \quad h(f(a)) = a \tag{191}$$

It implies that on the domain where the aggregation is performed, the function *h* is the inverse function of f :

$$h = f^{-1} \text{ on the aggregation domain} \tag{192}$$

This constraint reduces considerably the liberty regarding the construction of the aggregation operator. But if we do not respect this restriction, how could we interpret the loss of property (191)? The first thing to remark is that if the aggregation is done on a great number of elements it may not be shocking that the aggregation of an element *a* lost among a large set of *neuter* elements do not give *a* as a final result. An interesting way to relax (191) is to reduce the domain of the constraint from *all a* to the only *neuter element* and *the boundaries*.

6.5.4 Association of Weight : the function σ

Now when the construction of the balance is completely defined, we can examine the method to associate the weights to the elements. We should not confuse the association method with the weight calculation procedure. These two questions are closely related and may easily be merged. Here we assume that we know already the values of the weights and we study the association method. We distinguish 3 different types :

Object association : This method of association consists in assigning a weight to each argument. This is the most often used procedure. On the balance equation (154) it corresponds to the absence of the function σ. Since here σ has none particular properties we usually assume that the weights have the same index as their corresponding argument (i.e. $\sigma(i) = i$). Figure 32 illustrates this by associating the weight number 1 to the argument (position) number 1, the weight 2 to the position 2, and so on. A good example where object association is used is the case of the weighted means.

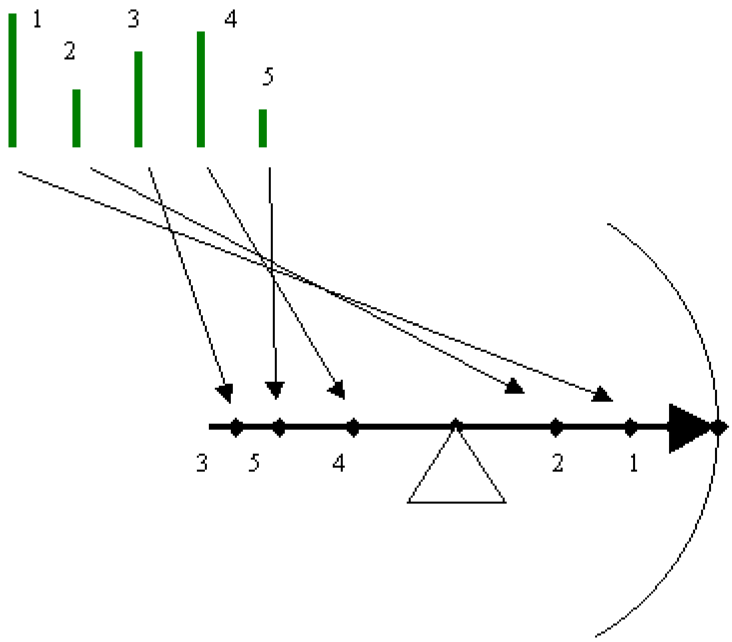


Figure 32. Object association of weights

Ordinal association : This method consists in ordering (by some method) the arguments and then to associate the first weight to the first argument, the second to the second one and so on. On the balance equation (154) it corresponds to using a permutation σ that orders the arguments in an increasing (or decreasing) way, the associating function being then σ^{-1}. A good example of aggregation operators where this is used are the OWA operators. From a mathematical point of view this method corresponds to the symmetrization of the first case. Figure 33 illustrates this method and we see that the first weight is associated to the smallest argument (argument 3) (i.e. $\sigma(1) = 3$). The weight number two is associated to the next one (argument 5) (i.e. $\sigma(2) = 5$). Note that here σ^{-1} orders the arguments in an increasing form.

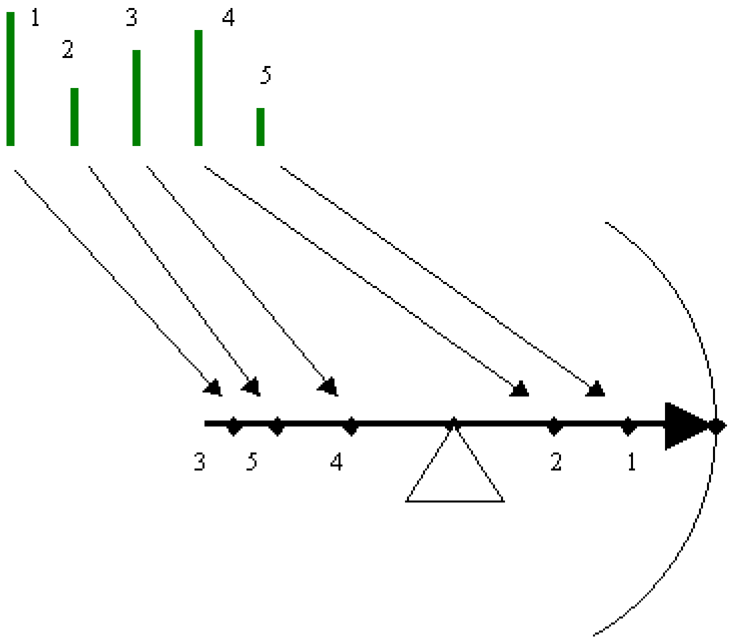


Figure 33. Ordinal association of weights

Cardinal association : This method consists in placing the elements on a scale and associating the weights according to their position. This new way of associating weights can be mathematically translated by saying that the associating function σ depends not anymore on i, as written in the balance equation (154), but on x (i.e. $\sigma(x)$). Figure 34 illustrates this method and we see that for example arguments 3 and 5 are obtaining the same weight 1, while argument 4 is associated to weight number 2.

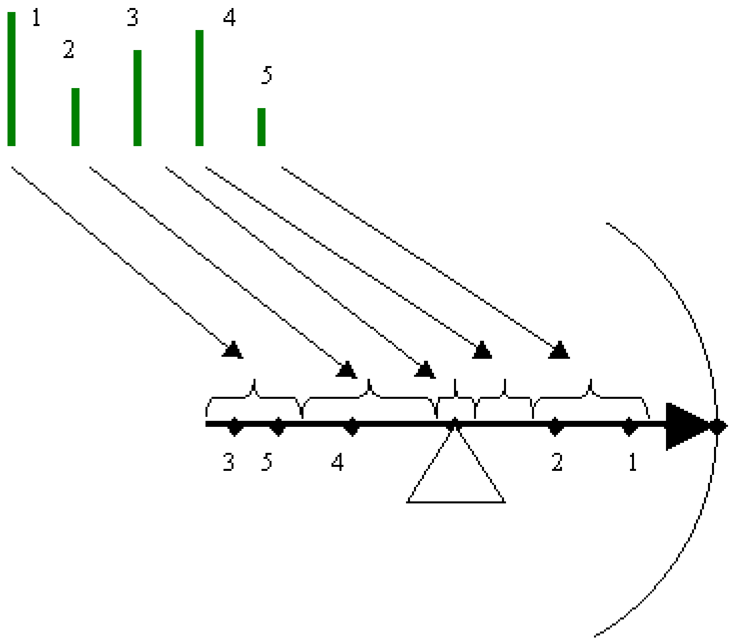


Figure 34. Example of Cardinal association of weights

Other associations : The metaphor allows the introduction of other kinds of association as for instance to associate the lightest weight to the smallest argument, the second lightest to the second smallest, and so on (see Figure 35). This will give an operator that insists on the "best" (i.e. highest) arguments. The symmetric operator is also an interesting operator. We can also imagine more sophisticated associations as for instance associating the lightest weight to the closest argument to the fulcrum, the second to the second and so on. This time the operator insists on the arguments far from the neuter element.

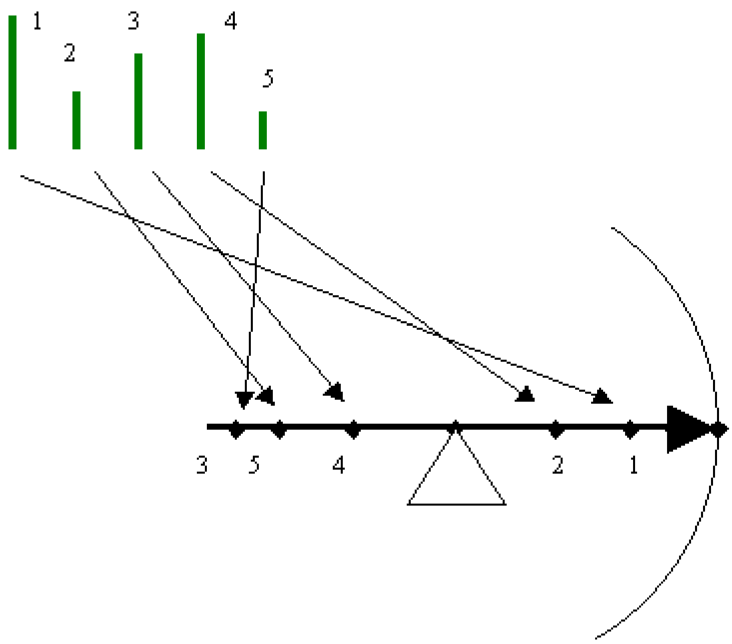


Figure 35. Example of other type of weight association

The study of the association of weights is clearly an important issue. The balance model clarifies and opens new perspectives in this area. However it is important to notice that in the balance model we associate one weight per argument and there are no "interaction-weights" attributed to sets of arguments, as it is the case for instance for the Choquet integral.

6.6 Why is the construction interesting ?

The first reason of why the balance is interesting is of course the fact that using a physical model for the metaphor we obtain an intuitive representation of the operator. This induces that we may simply understand the behavior without using mathematics. It also may inspire (intuit) new interesting properties and constructions.

6.6.1 Behavioral properties

Now, if we forget about the metaphor and think about our model as just an aggregation operator, we will notice the interesting properties (see chapter 1) :

Weights of parameters : the weights are introduced in a natural way and do not perturb the commutativity. In fact, equation (154) without weights is clearly commutative. Since we consider an association function for the weights, this means that we can put the arguments wherever we want and the function will manage to associate the right weight to the right argument.

Reinforcement and Counterbalance : These properties seem to be interesting [90] in fuzzy modeling and decision making. And they appear naturally on the balance. We intuitively know that arguments put on the same side of the balance will create reinforcement and arguments put on both sides (with the fulcrum in the middle) will create counterbalancement.

Interpretability of the parameters : Since we use a metaphor, the parameters have a real physical meaning. In the previous sections we show the analogy between fulcrum and neuter element, lever and function *f*, weight scale and *h*, etc.

***Neuter* element** : It represents the value associated to an argument that has no influence in the aggregation. This again seems to be an interesting property in fuzzy modeling and decision making.

6.6.2 Mathematical properties

Let us now take a more mathematical point of view. Since usually we aggregate finite real numbers, we will restrict this discussion to operators acting on the unit interval (i.e. [0,1]). It has been shown [29] that if we require the aggregation operator satisfies some reasonable properties (see below), then *all* these operators can be written in the form :

$$\alpha = f^{-1}\left(f\left(x_1\right) + f\left(x_2\right)\right) \qquad (193)$$

where $f(x)$ is continuous and strictly increasing. And the reasonable properties are the following :

- Continuity on the unit square except on the point (0,1) and (1,0).
- Requirements related to the Pareto optimum (monotonicity).
- Associativity.
- Self-duality with respect to a strong negation.

Another reasonable property set leading to the same result was proposed in [55]. The authors required the cancellation law on the open unit square (i.e. $]0,1[^2$), instead of the self-duality.

6.6.3 The natural choices

We observe that (193) is a *natural* construction of the balance equation (154) where all the weights equal the unity. By "natural" we mean the fact of using $h = f^{-1}$, even if it is not a constraint. In fact if we take into account the constraint (191), then we have automatically this property.

Another *natural* choice usually performed while constructing a balance is to place the *neuter* element somewhere within the unit interval. This property combined with the previous choice (see (192)) seems to give interesting operators. In fact from a theoretical point of view, several authors have proposed this operator under different names : associative compensatory operators [55], aggregative operator [29], or generated uninorms [43].

Several authors [55],[29] have shown under several forms that if we take the following natural choices for the *f* function :

- $f:[0,1]\rightarrow[-\infty,\infty]$
- $\lim_{x\rightarrow 0^+} f(x)\in\{-\infty,0,+\infty\}$
- $\lim_{x\rightarrow 1^-} f(x)\in\{-\infty,0,+\infty\}$
- *f* is continuous
- *f* is strictly monotone
- $h = f^{-1}$

Then the balance operator will be :

- associative
- continuous, it may not be the case for the only points (0,1) and (1,0)
- strictly monotone increasing on the open unit square
- $h(0)$ is a neutral element
- commutative (if we consider that the weights are associated to the arguments and not to their position in the mathematical formula).

This result seems very encouraging for the use of this model, since its *natural* choices seem to give interesting operators. But these *natural* choices are just preferences and not constraints, in fact the balance model tolerates very strange constructions.

6.7 Applications

Aggregating Truth and Falsity

The aggregation operators obtained with the balance can belong to known families. But we are particularly interested in the almost continuous enthusiastic aggregation operators (see section 5.4.2) since it seems to be the suitable way to aggregate truth and falsity values in a non logical-phrase calculus way.

We observe that an almost continuous enthusiastic aggregation (130), corresponds to the very nice balance that has the fulcrum at zero point, the lever scale is symmetric with respect to the fulcrum and is not bounded. The weight scale is built so that the total weight of a single argument equals its value, which leads to $h = f^{-1}$.

Using the balance framework we identify a particularly interesting operator, because of its mathematical simplicity. This almost continuous enthusiastic operator is generated with the function $f : [-1,1] \to [-\infty,+\infty]$:

$$f(x) = \ln\left(\frac{1+x}{1-x}\right) \tag{194}$$

The inverse function being :

$$f^{-1}(y) = \frac{e^y - 1}{e^y + 1} \tag{195}$$

Using this generator, we obtain a very simple (mathematically and computationally) formula. We have for n arguments :

$$Enthu_f(x_1,\dots,x_n) = \frac{\prod_{i=1}^{n}(1+x_i) - \prod_{i=1}^{n}(1-x_i)}{\prod_{i=1}^{n}(1+x_i) + \prod_{i=1}^{n}(1-x_i)} \tag{196}$$

Because of the associativity formula (196) can be used under a two-arguments form :

$$Enthu_f(x,y) = \frac{x+y}{1+x\cdot y} \tag{197}$$

In section 5.5, we indicated the particular operator used in a famous expert system (MYCIN) and we proposed a generator. Here using the step by step construction methodology presented in section 6.5, we were able to build a parameterized almost continuous enthusiastic operator. The parameter controlling the sensitivity, that corresponds to intensity of the reinforcement in the aggregation of truth and falsity values.

The operator presented in section 5.5 is based on the function "-ln(1-*x*)". Using a balance visualization and some tries-out, we arrive to the conclusion that one way to control the sensitivity is to introduce a power function. We obtain like this the generator *h*: [-1,1]→ [-∞,+∞] (with parameter *r* >0) :

$$h(x)=\begin{cases}-\ln(1-x^{\frac{1}{r}}) & for \;\; x\geq 0\\ \ln(1-(-x)^{\frac{1}{r}}) & for \;\; x<0\end{cases} \tag{198}$$

And we obtain for two arguments the following almost continuous enthusiastic operator :

$$Enthu_h(x,y)=\begin{cases} sign(x+y)\cdot(|x|^{\frac{1}{r}}+|y|^{\frac{1}{r}}-(x\cdot y)^{\frac{1}{r}})^{r} & for \;\; x\cdot y\geq 0\\ sign(x+y)\cdot\left(\dfrac{\max(|x|^{\frac{1}{r}},|y|^{\frac{1}{r}})-\min(|x|^{\frac{1}{r}},|y|^{\frac{1}{r}})}{1-\min(|x|^{\frac{1}{r}},|y|^{\frac{1}{r}})}\right)^{r} & otherwise\end{cases} \tag{199}$$

Even if (199) looks particularly complex, we would like to emphasize the fact that this operator is not only *continuous* on $[-1,1]^2$ (besides the extreme points (-1,1) and (1,-1)). But also, as an enthusiastic operator, it is associative, commutative and monotone.

The parameter *r* translates the "intensity of the reinforcement property". So, we have that :

- When $r \to 0$, then we tend to "no reinforcement". We obtain at the limit case an operator that for the aggregation of two truth values, will just choose the most truth one. (i.e. there is no reinforcing that will give a greater value than the most truth one). Mathematically, we have that for *x,y* ≥0, when $r \to 0$, then $Enthu_h(x,y) \to \max(x,y)$.
- For *r* =1, we have that $Enthu_h$ is nothing else than the operator used in MYCIN.
- When $r \to +\infty$, we tend to the drastic reinforcement. The limit being an operator that for two truth values (≠0) gives the total truth and for two falsity values (≠0) gives the total false.

We use this particular operator in an application related to video querying. For more information on this, refer to chapter 8. Here, the metaphor helped us to identify and to establish interesting truth and falsity aggregation operator, but it also provides a rich support for interpretation and introduction of new ideas.

The introduction of new operators and new problems

In this chapter we presented an analogy between a general aggregation operator and the physical model of a balance. The faculty of the balance operator to take the form of most operators of the classical families allows us not only to have the behavior of each family, but also to compare it on the same general framework. But without taking into account the generalization ability, the balance operator can be considered as a good operator, because the parameters are interpretable, we can use weights, we can control the behavior, it presents a *neuter* element and it has other interesting mathematical properties as reinforcement, counterbalancement and commutativity. Also several mathematical results (see section 6.6.3) point out that when doing the natural *choices* we obtain very interesting operators. These *choices* are not compulsory constraints, leaving a particular freedom to new construction and/or problems.

We can also obtain new operators. We understand by "new", operators that have not been precisely studied. This is obvious since all of the operators obtained in this framework will have the known mathematical form (154). Examples of this new operators which will lead us to further works are operators where the constraint (191) will be relaxed to only some points, as for instance to the only neutral element *e* :

$$h(f(e)) = e \tag{200}$$

Even if this relaxation may look like strange in section 6.2.6, we presented an operator (used in a real problem) that only satisfies (200) and not (191).

We can also study operators having a cardinal method for the weight association :

$$WA(x_1,\ldots,x_n) = h\left(\sum_{i=1}^{n} w_{\sigma(x_i)} \cdot f(x_i)\right) \tag{201}$$

A particular attention should be given to the association of weights, where we have a mine of ideas and problems available. We can also take a look at any other construction using crossing methods from well-known families, as for instance making ordinal association to operators other than the weighted mean. As for instance t-norms leading to ordered weighted t-norms.

$$OWT_{w_1,\ldots,w_n}(x_1, x_2, \ldots, x_n) = \mathop{T}_{i=1}^{n}\left[S\left(1 - w_{\sigma(i)}, x_i\right)\right] \tag{202}$$

We should not forget either the possibility of constructing strange operators, where we use non-monotone or non-continuous functions. As for example the function that rounds numbers to one decimal.

$$RA(x_1,\ldots,x_n) = round_{0,1}\left(\sum_{i=1}^{n} w_{\sigma(i)} \cdot f(x_i)\right) \tag{203}$$

Clearly the balance model can be seen not only as a study framework or a construction method. It is a mine of ideas, it is a rich metaphor that we should exploit.

Third Part

The Video Querying

The information revolution we are participating in, is making of the video a main actor, because of its richness,. But the problem of finding the information we want is complex issue. Here we are not dealing with text where we can easily extract the more frequent keywords, here we are dealing with imprecise, uncertain and incomplete information.

In this third part we present how the theoretical results, shown in the second part, can be implemented in order to solve the problem of the aggregation of imprecise, uncertain and incomplete information in the multimedia case. Our multi-media being the video channel. We divided this section into two chapters :

In **chapter 7**, we present a model that allows to browse a video using imperfect temporal queries. We propose a fuzzy query system based on fuzzy continuous annotations. This model follows the spirit of Zadeh's idea "Computing with words". We introduced a dictionary with the basic concepts and the way to construct new ones. All the aggregations being done in a logical way, we use t-norms and t-conorms. But the choice of the operator is not an easy task, so we use results of chapter 3 to help us. We also notice that in the case of the video the number of aggregation can easily explode, so we propose to use the fast operator proposed as result of chapter 4.

Although, when aggregating different sources indicating the same result, we should use another kind of aggregation than t-norms and t-conorms, as explained in chapter 5. In **Chapter 8**, we illustrate this and we present a Java based in-video search engine that uses a continuous enthusiastic operator (see section 5.4.2) to aggregate the matching query criteria. The operator was studied and tuned using the balance framework, presented in chapter 6.

Chapter 7

Browsing with Words

In this chapter, we focus on how to navigate in an annotated video by making temporal queries. The annotations may be in a database with other information. We assume that the annotations are precisely time-indexed, but their attached information may be uncertain. In other words, we know precisely at what time of the video something happens, but we are not completely sure about everything associated with the event. We note that this information may come from automatic or manual indexing. In order to handle this unavoidable uncertainty we propose to use fuzzy methods. We work here on scale the traditionally used scale (i.e. [0,1]) and all the aggregations are of the logical type (i.e. t-norms and t-conorms).

We introduce, following the spirit of Zadeh's idea [99] of "Computing with words", a dictionary with the basic concepts and methods for constructing new ones. Since in this approach we just have logical Computing with words allows us to have a human friendly interface. With this vocabulary and the logic tools introduced, the user will be able to realize human type queries. We focus on the time related queries [84],[13],[88]. We remark that our dictionary contains notions such as time positioning, time descriptors and time relationships. We show how to combine them in a uniform way in order to construct the queries. However the choice of the operators being a difficult task, we use the results of chapter 3 to make a choice.

We also notice that the stream of information in the video channel is so high that for the simplest calculi we have to do a untold set of aggregations. Results of chapter 4 help us to face this problem, by using a non associative approximation of Yager t-norm.

Let us start by explaining the most common way information is attached to a video, the annotations. And what we exactly understand by a fuzzy continuous annotation.

7.1 Fuzzy continuous annotations

7.1.1 Definition and examples

The actual works (see [50],[64],[49],[79],[88]) on query systems for video are based on the use of annotations. These annotations can be considered as information contained in a database associated to the video and indexed by the time. These annotations can be extracted from the video manually or automatically. With the automatic way it is clear that a lot of uncertainty arises. Just to give an example, the automatic face recognition is not, at this time, a hundred percent process.

But even the manual indexing, which can be considered as the most reliable way of obtaining the annotations, contains uncertainty. Not because of human errors, but because of the complexity of the world. For example when annotating night and day scenes, we can have a smooth passage from the day to the night. So we propose here to use fuzzy annotation to enrich the descriptions.

We call fuzzy annotation a classical annotation accompanied by a degree of certainty of the information (and not of the time indexing this annotation). This degree is usually a value between 0 and 1 (zero for completely uncertain and one for completely certain). So, for example an annotation can be : "At minute 6 the actor on the scene is Robert with a degree of certainty 0.75". Which means that we think that the actor is Robert but we are not totally sure. We notice here that we assume the indexing time (6 minutes) to be certain.

We speak about *continuous* annotations because we consider that we have the information for every time. Now, we can represent this information on a graph, where the *x*-axis is the indexing time of the film and the *y*-axis is the degree of certainty. Note that the actor appears for a period of time so that we have a curve and not a point.

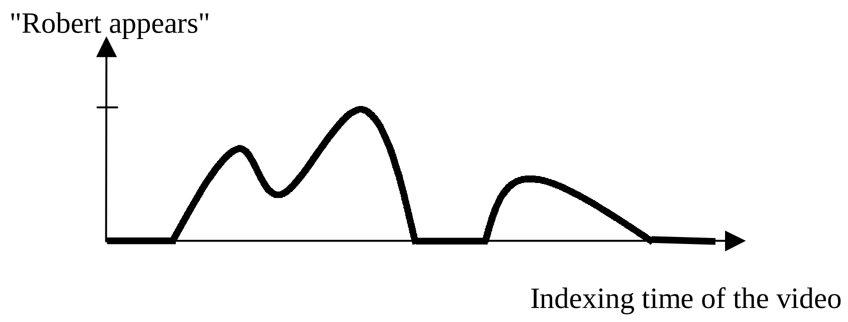


Figure 36. Fuzzy Annotation : "Robert appears".

In an analogous way we can represent on the indexing time axis an annotation like : "Between the minutes 10 and 11, we are sure that there is a dialog, but the beginning and the end of the dialog are not precise".

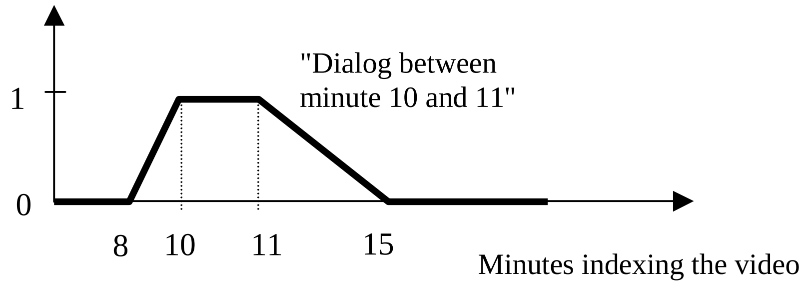


Figure 37. Fuzzy Annotation of a dialog

We remark that fuzzy annotations are a generalization of classical ones. We can obtain classical annotations by using only the degrees 0 and 1 of certainty. Now that we have these annotations (fuzzy or not), we want to use them in order to extract information. So in the following we propose a fuzzy time related query system. We introduce first what we call the fuzzy vocabulary.

7.1.2 Placing the video player

Placing the video player at the starting time is not a trivial issue. Since we have fuzzy annotations, we do not know exactly when the event starts. If we use just certain information, the video player will always be placed after the real start of the event we are looking for. This will force the user to rewind in order to see the beginning, and it is not something we want. Taking this into account, we may think that a good solution is to start at the point where the membership function starts (i.e. not null). This time the video player will be placed too far in advance and the user will have to wait until the event happens. This is clearly not a good solution either.

The action of indicating an exact start time, can be seen as a defuzzification process. We use an approach, based on the alpha-cuts. The idea is to work by alpha-cuts. Here, we propose to simply take as starting time(s) the minimum(s) of the (intervals) of the 1/2 cut . This gives us a point(s) where we are more or less sure that it starts. We pre-select this alpha-cut, but we leave the possibility to the user to change its attitude for the defuzzification. For more details on this kind of defuzzification method refer to our paper [18].

However, we leave to the user the possibility of increasing and decreasing the alpha value. In Figure 38, we see that the user can choose the alpha value with a scroll bar and observe directly the consequence of its choice.

In the previous lines we used the parenthesis to point out that they may be several moments in the movie answering to the question.

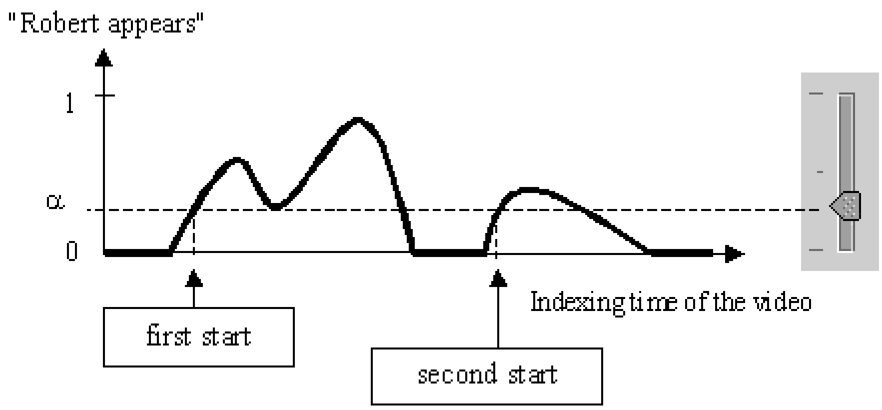


Figure 38. Choosing alpha for the defuzzification

7.2 Fuzzy time vocabulary

In the spirit of Zadeh's idea of computing with words [99] and Yager's work in [88], we propose to construct a fuzzy time related dictionary. This thesaurus will allow us to handle imprecise time querying. Using this we will be able to use time positioning definitions such as *beginning*, *end* and *middle*, to use imprecise time duration such as *about five minutes*, *long* and *short time* and to use time relationship like *after*, *before* and *close*. We will also see how to modify and combine them.

7.2.1 Time positioning

One of the things, which we may want to express, is a fuzzy positioning (of an event) in the time. This concept gives us an imprecise location in the time, to which we refer. Examples of this kind of concept are *beginning, middle* and *end*.

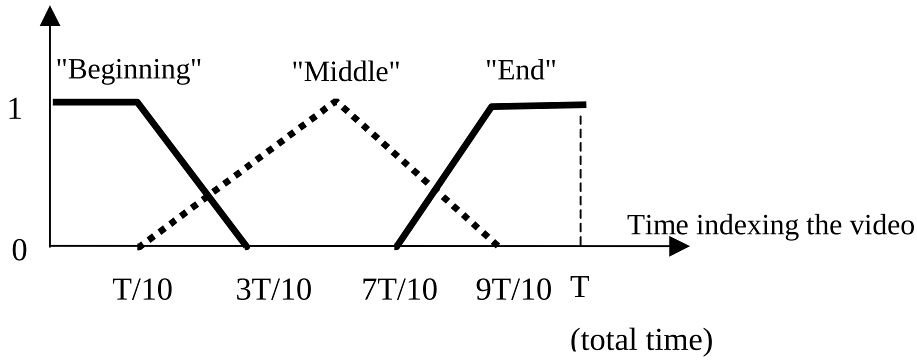


Figure 39. Beginning, Middle and End.

We remark here that the fuzzy *beginning*, *middle* and *end* are defined in relationship to the total reference time. We note that these notions are personal and so the user should personalize his translations, but always on a reference scale.

7.2.2 Duration descriptors

We may also want to describe the duration of some event. Here notions like long time and short time appear. Once more, even if it is not explicit, we assume that there is a reference. It is not the same "*long time*" in a 5 minute video and in a two hours film. We can then define this notion in a time scale having the total reference time.

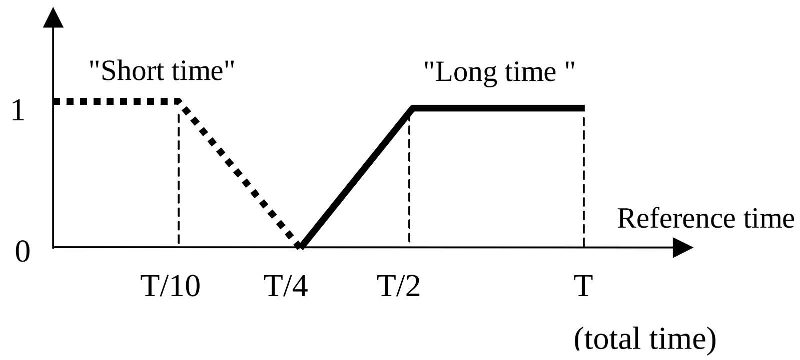


Figure 40. Long and Short time

We can also define on the time reference scale the notion "*about X minutes*". Here, once again we think that this notion is relative. We believe that the uncertainty grows with the value *X*. So, for example if we consider that the precision on "about five minutes" is a single minute, then the precision on "about 60 minutes" is at least 5 minutes (certainly more that 1 minute). We propose to translate "*about X minutes*" into "*between* (*X* minutes less 20 %) *and* (*X* minutes plus 20 %)" with the certainty of 1 for *X* minutes and a linear decreasing certainty between the extremes and the *X* minutes (see Figure 41).

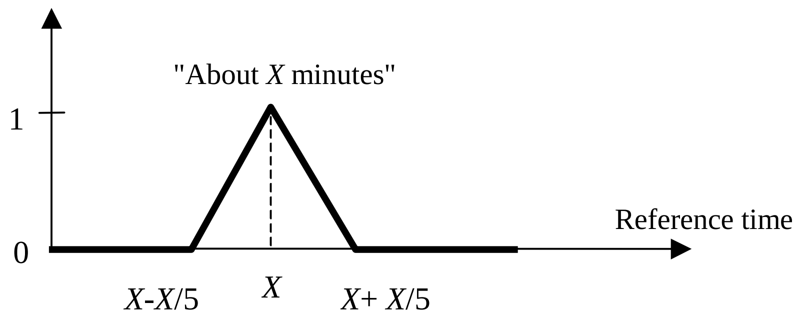


Figure 41. About *X* minutes

7.2.3 Time relationship

We may also want to define relationships between time events, like for example *after* and *before*. In this spirit Yager introduced in [88] a general framework : the relative temporal relationship (RTR). In this framework we have for example that the definition of "*after*" will be : If *X*-*Y* <0 then the degree of satisfaction of the concept "*X after Y* " is 0 and if *X*-*Y* ≥0 the degree will be 1. In a symmetric way we can define the notion "*before*". We propose here to use the time descriptors introduced before to generate new notions as for instance : "*About 10 minutes after*" or "*About 10 minutes before*".

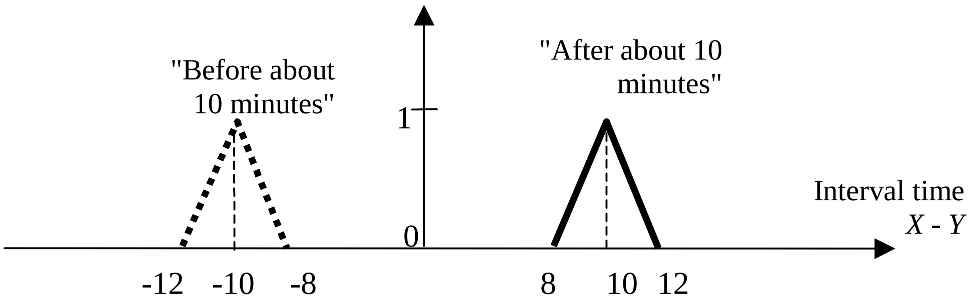


Figure 42. About 10 minutes after (or before)

7.3 Combining the fuzzy time vocabulary

7.3.1 Using time relationship

Now, that we have defined the basic relationships we may want to use them to indicate a particular moment in the video. So we may point to : "*About 10 minutes after the crash*".

Let *Crash*(*y*) be the membership degree of the event "*the crash*" at the time *y*. Then the membership function of "*About 10 minutes after the crash*" indexed by the time *x* will be obtained by following formula, where *T* is a t-norm :

$$About\_10m\_after\_Crash(x) = \max_y \left[T\left(after\_about\_10m(x-y), Crash(y)\right)\right] \quad (204)$$

This function allows to calculate simply the degree of membership at the time *x* of "*About 10 minutes after the crash*".

Let us generalize this result and let *R* be the membership function of a time relationship and *E* the membership function of an event, then we can point to a new moment of the video by using the general formula :

$$R \circ E(x) = \max_y \left[T\left(R(x-y), E(y)\right)\right] \quad (205)$$

We remark that the event *E* can also be a time positioning, like *beginning*, *middle* or *end* of the video.

7.3.2 Choosing the t-norm

As we just said, formula (204) allows to calculate simply the degree of membership at the time *x* of "*About 10 minutes after the crash*". However the simplicity is on the computing part, but the choice of the t-norm *T* is not clear. Here we propose to use the methodology and the results of chapter 3.

Looking at equation (204) we notice that we are trying to obtain the "best answer" for the logical conjunction of "after about 10 minutes" AND the "crash" event. We may immediately think about what should be the result for this conjunction, when we are not really at a point about 10 minutes after another point that is more or less a crash. This question translates the aggregation of the most fuzzy situation (i.e. *T*(1/2,1/2)).

The answer to this question translates the tolerance to the uncertainty and has several consequences. If we choose for the most fuzzy situation to attribute the value 0, this means that we are not tolerant to the uncertainty. We are going to obtain then a reduced area for the conclusion. In the opposite way, if we choose the largest value (i.e. 1/2), we will consider the largest area for the pointed event (see Figure 43).

In this figure we point out "about 5 minutes after an event that happened more or less between 3 and 5 minutes". Here "about 5 minutes" is a triangular fuzzy number being tolerant for more or less one minute. We notice here that clearly, when choosing the product (where *T*(1/2 ;1/2)=1/4), we obtain a considerable reduction, with respect to the minimum (where *T*(1/2 ;1/2)=1/2), for the uncertainty area.

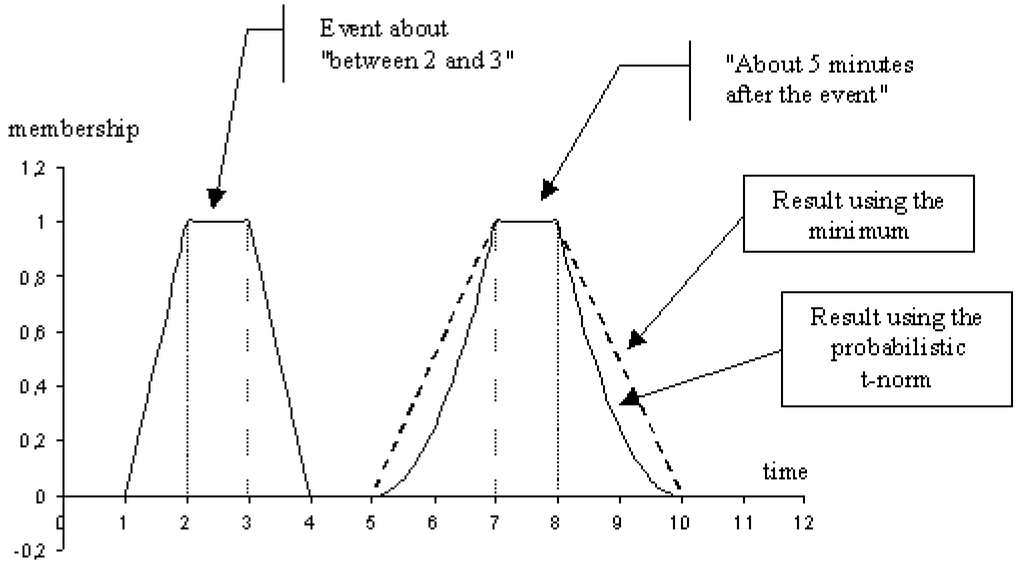


Figure 43. Pointing to "about 5 minutes after an event that happened more or less between 3 and 5 minutes".

It is to notice that we are differentiating the t-norms by their attitude when aggregating the uncertainty, since we showed (see Figure 2) that all the t-norms have the same behavior when aggregating a certain value (i.e.1). This induces here that no matter what t-norm we use, we will always obtain the same certain interval (i.e. the same core).

We can conclude that if we do not care about the uncertainty (i.e. everything that is not the core), then we can use any t-norm. This may happen if we just want to point the certain areas. However, we are convinced that this approach is deficient, since it is too strict. With just this information the video-player will be placed far after the beginning of the event, forcing the user to rewind.

In order to take into account the uncertainty, we propose to place the video-player at the beginning of the 1/2 cut, leaving always the possibility to the user to change the value 1/2.

In conclusion, the attitude with respect to the "full" uncertainty is the central notion. So, the results from chapter 2 help us to identify the t-norm. If we choose a parameterized because of the variety of attitudes it offers, again the results of chapter 2 will help us to select and interpret the influence of the parameter. Using the inverse function we can even leave the choice of the attitude to the user.

7.3.3 Using fast operators

It is clear that an interesting solution for the previous choice of the t-norm is to pick out a parameterized t-norm with a large *T*(1/2, 1/2) image space. Like this we have always the possibility of choosing (in a large spectrum) the attitude with respect to the full uncertainty. So, we may select the Yager t-norm.

However, looking at (205) we notice that we have to compute for every time *x* of the video, the aggregation (by the t-norm) of every time *y* in order to take the maximum. Taking into account that at least we have 25 frames per second, we have very quickly a great number of calculi. For instance just for *one hour* video, and for *one* time relationship we will have to compute 8.100.000.000 times the t-norm.

Looking again at equation (205) we observe that we never use the associativity property of the t-norms. In fact each time we just aggregate two values. Here the fast "Yager t-norm-like" operator (see section 4.7) may be interesting. In fact, this operator is computationally lighter than the Yager t-norm. We recall here its definition (117) :

$$R^{\min-\max}(u,v) = \max\big(\beta \cdot (\max(u,v) - 1) + \min(u,v),\ 0\big) \tag{206}$$

In order to be coherent with the results of chapter 2, we compute here the image of the "most fuzzy point" :

$$t = R^{\min-\max}\left(\frac{1}{2},\frac{1}{2}\right) = \beta \cdot \left(\frac{1}{2} - 1\right) + \frac{1}{2} = \frac{1}{2} \cdot (1 - \beta) \tag{207}$$

So, if we inject the inverse function into (206), we obtain equation (208), where the parameter *t* is the value expected for the most fuzzy aggregation :

$$R^{\min-\max}(u,v) = \max\big((1-2t) \cdot (\max(u,v) - 1) + \min(u,v),\ 0\big) \tag{208}$$

We impose to *t* the range [0,1/2], which reduces a bit the generalization capacity of the equation but we gain in intelligibility, without losing t-norms having all the possible attitudes with respect to the aggregation of the full fuzziness. Note that we lose just some very strong t-norms (stronger than the Lukasiewicz one).

Now, if we choose that the attitude with respect to the "total fuzziness" should be relaxed (i.e. *t* =1/2) then the operator (208) becomes the minimum (the largest t-norm). And for the strictest attitude (i.e. *t* =0), (208) becomes the Lukasiewicz t-norm.

7.3.4 Constructing more complex notions

We may also want to use the richness of the logic operators to create more precise notions. It is important to note that we can only combine notions having the same time scale. We can in this way combine two different relations, two different time descriptors, two different events, two different time positioning, but also an event and a time positioning.

In order to built these new notions we use of course the fuzzy logic operators. We use the t-norms to translate the logic 'AND' and the t-conorms to translate the logic 'OR'. Since we propose to use for the negation the function $x \rightarrow not(x) = 1 - x$, we advise to use dual t-norms and t-conorms.

Here again the question of the choice of the t-norms and t-norms arises. And again result of chapter 2 gives us an answer. The main question is : what is the attitude we want for the aggregation of the "most fuzzy" values ? And once more if we know the answer to this question, chapter 2 will guide us to choose the right couple of operators.

Chapter 8

Simple Constrained Queries over Fuzzy Shot-annotations

The amount of information stored in computers is growing, not only because of the richness of the video channel, but also because this richness is making of it the favorite media. In this context a main question arises : "How to get the information you want?" We propose here a new tool leading the user to make the right question in order to retrieve the information he wants inside the video.

Useful information may be automatically extracted from the multimedia streams. For instance, cuts and camera motion can be detected from the video, while cues such as applause, silence and speaker identity can be found from the audio. However at the actual state of the art, these indications are not rich enough, so we still use human annotation.

We notice that it is not always easy to aggregate the confidence scores coming from automatic programs and the human ones. Here we point out this problem and suggest some solutions. In particular we use the scale [-1,1] and the result presented in chapter 5.

Here we do not work anymore on continuous annotations since the humans annotate by shot. A shot corresponds to the segment of video captured by a continuous camera recording and is classically used in all the annotation systems. This can be explained by the fact that shots are "sense units", provide some structure to the video, similarly to paragraphs in text document and are easy to extract automatically [52], [100], [5], [97], [78], [3], [73]. Actually, the excellent result in this field make that most of the systems being developed now use this granulation. The annotation by shot will reduce the complexity and the size of the search domain, but will introduce more uncertainty and imprecision due to extraction methods.

We finish this chapter by describing the technology we use to create our query prototype. We also present the structure of our annotations written in XML, which is becoming a standard in video applications and more generally in multimedia and web technology.

8.1 Fuzzy Annotations per Shot

The present works on query systems for video is based on the use of shot annotations [64][49],[79],[88]. These annotations can be considered as information contained in a database associated to the video and indexed by the identification number of the shot. These annotations can be extracted from the video manually or automatically.

The automatically derived information can be generally described as a time-dependent value that is synchronous with the source media. For instance, annotations might come from output of a face-recognition or speaker identification algorithm. In this case it is clear that a lot of uncertainty arises, due to algorithmic imperfections. But, we would like to stress again (see section 7.1.1) on the fact that even the manual indexing, which we can be considered as the most reliable way of obtaining the annotations, presents some uncertainty, not because of human errors, but because of the complexity of the world and the subjectivity of the annotator.

Here we propose to use again fuzzy annotations (see section 7.1) to enrich the descriptions. But this time the annotations are going to be indexed by shot (and not of the time indexing this annotation).

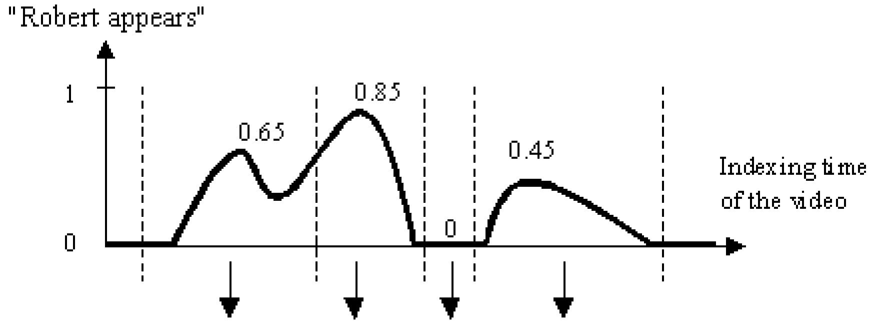
We assume that for each shot we have the start time and the end time. This information could also be considered as fuzzy annotations since they are usually coming out from an automatic system. Here we do not take into account the uncertainty attached to these values. We consider that the given times are the best we can obtain. Further research can be done taking into account these uncertainties.

We also do not use the classical [0,1] range of degree of certainty, but a more natural scale [-1,1], where –1 is the complete falsity, 0 the total ignorance and 1 the complete truth (see chapter 5). This scale allows us to point out a classical mistake in aggregation of annotation information. Automatic detection programs are positively oriented, which means that they are confident in their good results, but not in their bad results. A degree 0 in a detection program means "we do not know" and not "certainly the object is not there". So for automatic indexing we use the [0,1] range. But, since human annotation is more reliable, when the user does not annotate we assume that the object is *not* present (and this observation being sure).

The manual annotations are instinctively done per shot, but the automatic ones are continuous (see section 7.1). So, we need to aggregate the information obtained during the shot into a unique value. Because of the positive attitude of the automatic algorithms we aggregate their confidence degrees per shot by taking the best obtained value. For example, if we are recognizing a particular actor, then we will say that the actor is present in the shot (with the degree of confidence c) if the best recognition provided by the algorithm for this shot is c. Our aggregation formula is then simply :

$$\textit{shot-annotation}(\#n) = \max_{t \in \text{shot}(\#n)} (\textit{automatic-annotation}(t)) \qquad (209)$$

We can illustrate this in the following figure :



Shot #	23	24	25	26
Robert appears	0.65	0.85	0	0.45

Figure 44. Aggregation per shot continuous annotation

8.1.1 Extraction

Useful information may be automatically derived from multimedia streams. For example, we detect cuts and camera motion from video. The cuts are typically found by computing an image based distance between consecutive frames of the video. Over a certain threshold we consider that there is a cut. The distance between frames can be based on statistical properties of pixels [52], histogram difference [100], compression algorithms [5], edge differences [97] or motion detection [78]. We use an automatic shot boundary detection developed in our laboratory [3],[73]. And we automatically annotate the camera motion.

Using the resulting video segmentation, we annotate each shot with keywords and a degree of certainty. We use an annotation program developed in our laboratory [81] that uses our specific XML DTD (described in the next section) to guide the manual annotation process.

8.1.2 Storage and transfer of the fuzzy annotations (XML)

In order to store and to transfer the annotations we attach to the video file an XML file.

XML (Extensible Markup Language) is a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere. For example, video producers might agree on a standard or common way to describe the information about the videos (length of the video, copyrights, and so forth) and then describe the product information format with XML. Such a standard way of describing data enables the user to use the same program for all the video independently of the video producers. For instance, an intelligent agent sent to any video producer web site, may query the database and then make a valid comparison of the results. XML can be used by any individual or group of individuals or companies that wants to share information in a consistent way.

XML is similar to the language of today's Web pages, HTML. Both XML and HTML contain markup symbols to describe the contents of a page or file (as for instance a video file). HTML, however, describes the content of a Web page (mainly text and graphic images) only in terms of how it is to be displayed and interacted with. As with HTML, identification of data is done with tags (identifiers enclosed in angle brackets). For example, a <P> starts a new paragraph. XML describes the content in terms of what data is being described. For example, a <VTITLE> could indicate that the data that followed is the video title. This means that an XML file can be processed as data by a program or it can be stored with similar data on another computer or, like an HTML file, it can be displayed. For example, depending on how the application in the receiving computer wants to handle the video title, it could be stored, displayed, or translated.

In our case we store for each <annotation> the shot number, the start time and the end time. Inside each annotation we can put information as for instance a <personage> with its name, the actor interpreting it and the degree of confidence on this information. We may also have object descriptions <object> and automatically detected camera motions <cameramotion>.

Here we have an example of an annotation in XML form, describing shot 16 :

```
   ( … )

   <annotation shot='16' startTimeCode='00 :01 :22 :75'
endTimeCode='00 :01 :27 :73'>

          <personage name='Falbala' actor='Laëtitia Casta' value='0.95'>
          </ personage >
          <personage personnage='Obélix' actor='Gérard Depardieu' value='1.0'>
          </ personage >
          <object description='Horse' value='1.0'>
          </object>
          <cameramotion description='zoom in' value='0.75'>
          </cameramotion>

   </annotation>

   ( … )
```

Table 14. Fuzzy Annotations per Shot in XML

XML is "extensible" because, unlike HTML, the markup symbols are unlimited and self-defining. This is what allowed us to define the new tags describing the annotations. Since it is self defining we could leave this file like this, but since it has to work with our application we need to be sure that some particular rules were respected (our standard). To achieve this we use a Document Type Definition. A DTD defines a class of valid XML documents, i.e. it defines which tags, attributes and elements are valid. Our DTD is the following :

```
<Doctype -
        <!ELEMENT annotation (personage * | object  *) >
        <!ATTLIST annotation
             plan                CDATA    #REQUIRED
             startTimeCode       CDATA    #REQUIRED
             endTimeCode         CDATA    #REQUIRED
        >
        <!ELEMENT personage EMPTY >
        <!ATTLIST personage
             personage           CDATA    # IMPLIED
             actor               CDATA    # IMPLIED
             value               CDATA    # REQUIRED
        >
        <!ELEMENT object EMPTY >
        <!ATTLIST object
             description         CDATA    # REQUIRED
             value               CDATA    # REQUIRED
        >
        <!ELEMENT cameramotion EMPTY >
        <!ATTLIST cameramotion
             description         CDATA    # REQUIRED
             value               CDATA    # REQUIRED
        >
>
```

Table 15. DTD for our fuzzy annotations.

As we just saw it is clear that some world standards are needed. Currently the World Wide Web Consortium (W3C) is the organization that coordinates the standard creation and provides some recommendations. Some examples of the existing formats are the Microsoft's Channel Definition Format (CDF), which describes a web channel. Or also the ChartWare, which uses XML as a way to describe medical charts so that they can be shared by doctors. Applications related to banking, e-commerce ordering, personal preference profiles, purchase orders, litigation documents, part lists, and many others are anticipated. In particular the standardization of the information attached to the video is known under the name MPEG7.

For more information on XML see the web site www.w3c.org/XML. And for more details on MPEG7 see the www.cselt.it/mpeg/standarts/mpeg7/mpeg7.htm site.

8.2 Querying

Because of the richness of the video media, the set of possible queries is incredibly large. In this situation, we consider that some restriction should be set. For instance for the same video the user may want to see the shot that uses this or that technique, while another user may look for an actor and a third one may want to find where the actor says : "Hasta la vista baby".

Our approach consists in parsing the XML-annotation file in order to extract its content and then guide the user to query the available annotations. Our program detects automatically the annotations corresponding to our DTD even if they are structured and inserted between other annotations such as for instance copyright, video format, etc.

The idea is to extract the available annotation tags and present them some how to the user so that he queries on the offered information. Our approach is to guide the user during the construction of the query. We are interested here on simple queries. And we understand by a simple query a set of keywords that will identify the shot.

Figure 45. Constructing the query with a combo box.

The construction of the query is incremental. The user has to choose (with a combo box) one after the other the keywords he is going to use for the query. This kind of querying prevents the user to request any not available information, since the combo box is built with the information extracted from the annotation file.

Then pressing the query button we launch the search and the ranking processes.

8.3 Ranking the results

After we have constructed the query we launch the research process. We select the matching attributes and we aggregate their certainty values in order to obtain global certainty value.

We rank then the shots from the most positive certain to the last one. Consequently, the quality of the result depends directly on the aggregation operator. The theoretical research done in chapter 5 suggested that we should use a continuous enthusiastic operator. In the next section we supply some explanation about this choice.

8.3.1 The aggregation of the certainty values

We work with truth and falsity values. The automatic-obtained values being all degrees of truth (i.e. in [0,1] while values provided by human beings use the whole truth scale (i.e. [-1,1]).

We also notice that we are not trying to compute the logical truth value of the conjunctive phrase built from the keywords of the query. In fact this kind of aggregation would lead to problems as for instance the complete elimination of the shots having just one false matching argument.

Instead we are performing an aggregation of the type described in chapter 5. We saw there that for this kind of aggregation there are two operators doing the work : the prudent and the enthusiastic. We saw that the prudent leads to an extreme augmentation of the uncertainty, so we propose to use an enthusiastic operator.

Besides that, being aware of the fact that we are aggregating truth and falsity values, we will not do the mistake of using the same scale for aggregating the automatic obtained values and the manual ones.

In addition to the rational arguments announced before, we will try here to show on two examples why this operator is more suitable than others classically used : the means [59], and the t-norms [66],[59]. In order to compare these operators we should work on the same scale. We use a linear transformation to transform our degrees (range [-1,1]) into the [0,1] range.

Let us first compare the enthusiastic aggregation operator with a t-norm. Let us imagine that we have the following description for the shot number 11 :

	Shot 11			
	Actor 1	Actor 2	Camera Motion Type 2	Main background Color = Black
Our scale [-1,1]	0.8	0.9	0.7	- 0.5
Equivalence on [0,1]	*0.9*	*0.95*	*0.85*	*0.25*

Figure 46. Description of Shot Number 11

Using a t-norm will discard this shot from the results, since the low degree for the background color 0,25 imposes to the aggregated value to be smaller than 0,25 (on the [0,1] scale (see property (46)). For this case the enthusiastic operator will compensate this negative matching (-0,5) with the other positive matching. The compensation property of the enthusiastic operator is the one used here.

Let us now compare the enthusiastic operator to a mean type operator : every time we do not know we annotate with the value 0 (i.e. the total ignorance). It is clear that when using the mean all these zeros will have an influence on the final score, while for the enthusiastic operator we will not influence the result, since 0 is the neutral element.

The neutral element also solves the problem of the missing values, because if there is a value missing then we replace it by the neutral element, which has no influence on total value. People using the mean may imagine that the problem of missing values (or even of total ignorance) can be solved by aggregating just the available values. This is a solution but it is equivalent to replacing the missing value by the mean-value (of the non-missing values). In other words you will give a high score to a missing attribute value if the other attributes have high scores. We do not think that this is the right approach.

8.3.2 Presenting the results

After obtaining the global scores we rank the shots from the most positive certain to the last one. We present the results showing keyframes (i.e. still images extracted from the video) of the best shots. Since the user based on just this image has difficulties to find his way about, because for instance the keyframes may be very similar. We choose to present the 4 best shots keeping the video time order. This means that on the left we have the first shot appearing in the video independently of it is the best matching result or not. However, we indicate over the keyframes the score for that particular shot. In the same way the second one is the second one appearing in the video and so on.

8.4 ExploriX Interface.

We have prototyped a query program that uses the above ideas, towards the application of browsing and finding musical video clips. All the video clips are low quality (TV quality), MPEG-encoded and made available via network. Our program is being developed in Java language and is able to read any of the current video and sound formats. ExploriX is a first step towards intelligent retrieval and use of fuzzy annotations, which demonstrates the feasibility of the proposition.

Figure 47 shows the user interface of our browser prototype. To the top left is a classical video playback window (with controls), that listens to the other windows. To the top right there is the "constructing" query window. The user builds his query there, using the combo box. Between these two windows we see a control panel, where we can read all the messages transiting between the objects (these need not be visible for the user). To the middle of the screen we see the result window. This window contains 4 keyframes of the 4 best shots. They are ordered on the time scale, respecting the order of appearance in the video.

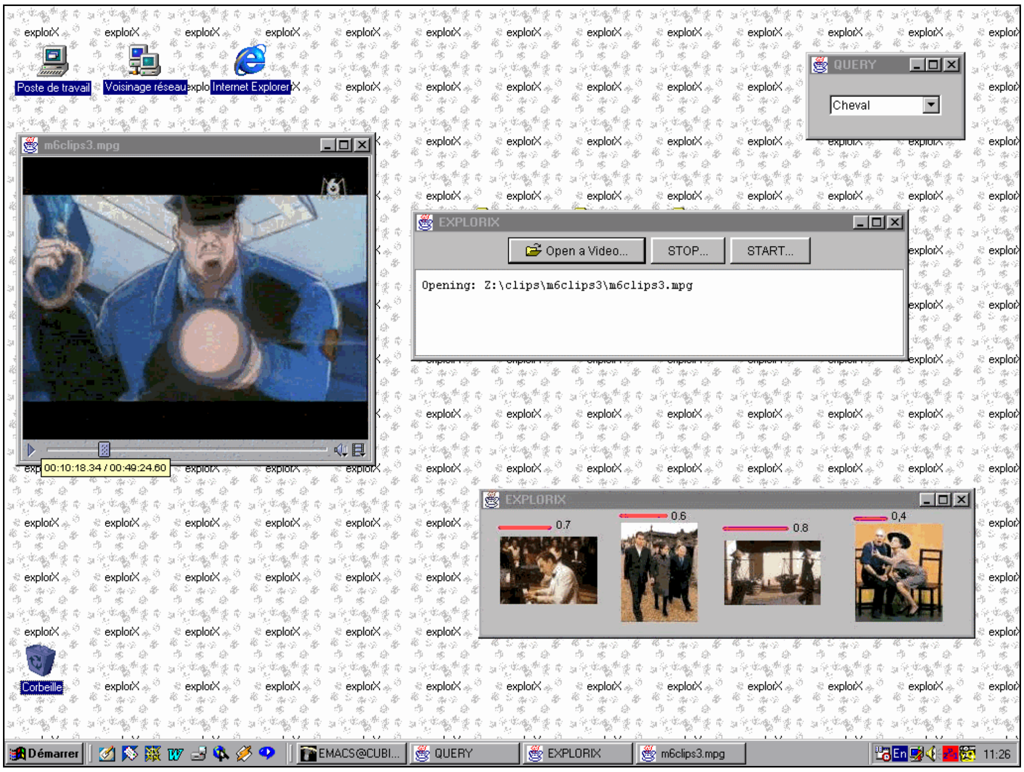


Figure 47. Screen shot of ExploriX

ExploriX was developed in the multimedia indexing group at LIP6 (University Paris 6). The group is partially funded by the AGIR project.

Conclusion and Perspectives

The explosion of the information technologies is bringing massive amounts of knowledge that are hidden in numerical data. In this context it is clear that the mathematical aggregation should play a main role in the reduction of this information into meaningful and representative subsets.

In this concluding section we would like to summarize the main results obtained around the mathematical aggregation. We present both theoretical and applied investigation on these operators. We will discuss the main achievements and the possible perspectives chapter by chapter.

The first part : The Mathematical Aggregation Operators

We start this manuscript by offering a global overview of the existing state of art. Here we try to brighten and to put some order in the existing definitions and properties of the mathematical aggregation operators.

In **chapter 1**, we propose, a minimal set of mathematical conditions that define an aggregation operator. We proceed by presenting other possible mathematical properties and we suggested interpretations for the mathematical characteristic.

In **chapter 2**, we present a catalogue of the existing operators. We portray their characteristics and advantages, but we try to be objective by giving also their disadvantages, which is rarely done.

Second part : Studying the Aggregation of the Uncertain

In the second part of this document, we present the new developments we propose for the aggregation of the uncertain. We start in **chapter 3** by studying how the different t-norms and t-conorms aggregate the uncertain. The idea is to focus our analysis on just one point, the "most fuzzy" one. This approach gives as a first result a classification method and as a second one, an interesting way of normalization of the parameters of the parameterized t-norms and t-conorms families. This new parameterization gives not only a meaning to the parameters, but also a common scale of work on which we can compare them. In particular we notice that the range of attitude with respect to the uncertainty is not usually completely fulfilled, in particular there are not many relaxed t-norms.

Based on this observation, we propose in **chapter 4** a way of creating new operators by reducing t-norms and augmenting t-conorms. Unfortunately we notice that the resulting operators are generally not t-norms (or t-conorms), because they are not associative. However, we find a particularly interesting operator, which seems to be a non-associative approximation of the parameterized Yager t-norm, with a special computational efficiency.

The prior studies focus on the t-norms and t-conorms that compute the truth value of a logical phrase. In **chapter 5** we are interested in the aggregation of different truth values observed for the same logical phrase. We propose an axiom set for the aggregation of truth values, which leads to the characterization of two truth-aggregation families, a prudent and an enthusiastic. The first one has a cautious attitude choosing between two observed values the one that is more uncertain. We show that this family is completely characterized, but its attitude tends to the total uncertainty, which is not very interesting for assisting human decisions. The second one has an enthusiastic behavior and reinforces the results if it observes twice the truth or twice the falsity and when observing a falsity and truth the operator gives a compensated value. Unfortunately these operators are usually not continuous. We propose an additive generated solution that provides an almost continuous solution, although it is not defined for the full contradiction. We proceed by characterizing for the enthusiastic case the full and the normal contradiction (notice the prudent case is completely defined).

After working on an axiomatic basis, in **chapter 6** we present a new perspective for the additive generated operators. This new point of view is based on the metaphor of a balance. A metaphor allows intuitive interpretation of the operators. And we show how to use it in order to visualize the global behavior of an operator. Pushing further the established analogy we discover the notion of sensitivity and the use of the derived function. We also use the metaphor to guide the construction of aggregation operators. The balance model allows to make the right choices while conceiving an aggregation operator. It is remarkable that the balance not only generalizes most of the aggregation operators presented in chapter 2 and it is particularly adapted for the operator highlight in the aggregation of truth and falsity values, but it also offers a larger framework that allows the interpretation of not mathematical friendly properties, as for instance a non continuous or non monotone additive generator. The balance framework also opens new perspectives for further research, in particular on new kind of weight association.

Third part : The Video Querying

The third part illustrates how the theoretical results can be implemented in order to solve practical problems of aggregation in the case of the video query systems. In **chapter 7**, we present a model that allows to browse a video using imperfect temporal queries. We describe a query system based on fuzzy continuous annotations. Since we are "computing with words", we establish a dictionary with the basic concepts and the way to construct new ones. Since all the aggregations are done on a logical background, we use the t-norms and the t-conorms. But the choice of the particular cases to be used is not clear, however the results of chapter 3 provide some answers. We also notice that in the case of videos the number of aggregation can easily go off, so we propose to use the fast operator proposed as result of chapter 4.

Nevertheless, when aggregating different sources indicating the same result, we should, as explained in chapter 5, use another kind of aggregation than t-norms and t-conorms. In **Chapter 8**, we illustrate this result with some examples. We also prove the feasibility of the method by presenting a Java based in-video search engine that uses a continuous enthusiastic operator studied and tuned using the balance framework, presented in chapter 6.

This work lead to several published results, which are not all described in this manuscript, since we opt to focus just on the mathematical aggregation aspects. We invite the reader to refer to [19] and [28] for aspects related to t-norms and t-conorms, to [21] for aggregation of truth values in non-logical way, to [22] and [24] for studies on additive generated operators, to [23], [20] and [76] for video aspects, to [18] and [96] for ranking methods, to [27] and [25] for defuzzification techniques and to [26] for a fuzzy comparison study.

We would like to conclude by stressing on the fact that the theoretical results obtained here are not particular to the multimedia problematic, they can be applied to any field that need an meaningful aggregation (as we did for the video querying). We also claim that this work may certainly be a starting point of further research on the aggregation of logical degrees in a non phrase calculus way and on new aggregation forms, in particular the ones based on additive generators.

Reference s

[1]. Aczel, J., *Lectures on Functional Equations and Applications*, Academic Press : New York, 1966.

[2]. Aczel, J., On Mean Values, *Bulletin of the American Mathematical Society 54*, 392-400, 1948.

[3]. Aigrain P and Joly P., The automatic real-time analysis of film editing and transition effects and its applications, *Computers and Graphics, Vol. 18, Nr.1*, 93-103, 1994.

[4]. Alsina, C., Trillas, E. and Valverde, L., On some logical connectives for fuzzy set theory, *Journal of Mathematics Analysis & Applications 93,* 15-26, 1983.

[5]. Arman, F., Hsu, A., Chiu, M-Y., Image Processing on Encoded Video Sequences, *Multimedia Systems, Vol. 1 No 1*, 211-219, 1994

[6]. Bonissone, P.P., Selecting uncertainty calculi and granularity : An experiment in trading off precision and complexity, *Proceedings of the first Workshop on Uncertainty in Artificial Intelligence,* Los Angeles, 57 - 66, 1985.

[7]. Bouchon-Meunier B., Ed., *Aggregation and Fusion of Imperfect Information*, Physica-Verlag, Heidelberg, Germany, 1997.

[8]. Buchanan, B., and Shortliffe, E., Ruled Based Expert Systems, The MYCIN Experiment of the Stanford Heuristic Programming Project, *Addison-Wesley*, Reading, MA, 1984.

[9]. Calvo T. and Mesiar R., Generalized medians. *Proceedings of AGGREGATION'99, Palma de Mayorca*, 159-165, 1999.

[10]. Calvo T. and Mesiar R., Continuous generated associative aggregation operators. *Proceedings of AGGREGATION'99, Palma de Mayorca,* 149-158, 1999.

[11]. Calvo T., De Baets B. and Fodor J., The functional equations of Frank and Alsina for uninorms and nullnorms, *Fuzzy Sets and Systems,* to appear.

[12]. Choquet, G., Theory of capacities. *Annales de l'Institut Fourier 5*, 131-295, 1953.

[13]. Cobb, M. and Petry, F., Fuzzy querying binary relationships in spatial databases, *Proceedings of the 1995 IEEE*, International Conference on Cybernetics and Society, Vancouver, 1452-1458, 1995.

[14]. Cooke R. M., *Experts in Uncertainty*, Oxford Press, Oxford, UK, 1991.

[15]. Davis, M., Knowledge representation for video, *Proceedings of the 12th National Conference on Artificial Intelligence*, AAAI Press, 120-127, 1994.

[16]. De Baets B. and Fodor J., On the structure of uninorms and their residual implicators, *Proceedings of the Eighteenth Linz Seminar on Fuzzy Set Theory,* Linz, Austria, 81-87, 1997.

[17]. De Baets B. and Fodor J., Van Melle's combining function in MYCIN is a representable uninorm : an alternative proof, *Fuzzy Sets and Systems 104*, 133-136, 1999.

[18]. Detyniecki M. and Yager R., Ranking fuzzy numbers using α-weights, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, to appear in 2000.

[19]. Detyniecki M., Yager R. and Bouchon-Meunier B., Specifying t-norms based on the value of T(1/2,1/2), *Mathware & Soft Computing vol.VII*, number 1, 2000.

[20]. Detyniecki M., Browsing a Video with Simple Constrained Queries over Fuzzy Annotations, *Flexible Query Answering Systems FQAS'2000*, Warsaw, Poland, October 2000.

[21]. Detyniecki M. and Bouchon-Meunier B., Aggregating Truth and Falsity Values, *FUSION'2000*, Paris, France, 18-24, July 2000.

[22]. Detyniecki M. and Bouchon-Meunier B., Building an Aggregation Operator with a Balance, *IPMU'2000*, Madrid, Spain, 686-692, July 2000.

[23]. Detyniecki M., Seyrat C. and Yager R., Interacting with Web Video Objects, *Proceedings of 18th International Conference of the North American Fuzzy Information Society - Nafips'99*, pp.914-917, New York, USA, June 1999.

[24]. Detyniecki M., Bouchon-Meunier B., and Yager R., Balance Operator: A new vision on Aggregation Operators, *Proceedings of Eurofuse-Sic'99*, 241-246, Budapest, Hungary, May 1999.

[25]. Detyniecki M. and Bouchon-Meunier B., L'évaluation des nombres flous avec des α-poids, *Journées francophones sur la Logique Floue et ses Applications - LFA'99*, Valenciennes, France, October 1999.

[26]. Detyniecki M. and Yager R., Comparing Possibilities through Probabilities, *Technical Report MII-1920*, Iona College, New York, 1998.

[27]. Detyniecki M. and Yager R., A note on using alpha-weighted valuations, *Technical Report MII-1913*, Iona College, New York, 1998.

[28]. Detyniecki M. and Yager R., A study on the value of the t-norms on (1/2,1/2), *Technical Report MII-1912*, Iona College, New York, 1998.

[29]. Dombi, J., Basic concepts for the theory of evaluation : The aggregative operator, European Journal of Operational Research 10, 282-293, 1982.

[30]. Dubois, D. and Prade H., Triangular norms for fuzzy sets, *Proceedings of the 2nd International Symposium of Fuzzy Sets*, Linz, 39 - 68, 1981.

[31]. Dubois, D. and Prade, H., A review of fuzzy sets aggregation connectives, *Information Sciences 36*, 85 - 121, 1985.

[32]. Dubois, D. and Prade, H., Weighted minimum and maximum operations in fuzzy set theory, *The Reliability of Expert Systems*, Hollnagel, E. (ed.), Ellis Horwood Limited : Chichester, 64-118, 1986.

[33]. Dubois, D. and Prade, H., Handling uncertainty in expert systems : pitfalls, difficulties, remedies, *Information Sciences 39*, 205 - 210, 1986.

[34]. Dubois, D. and Prade, H., Representation and combination of uncertainty with belief functions and possibilities measures, *Computational Intelligence 4*, 244 - 264, 1988.

[35]. Dubois, D., Prade, H. and Bezdek, J., (Eds.) *Fuzzy sets in approximate reasoning and information systems*, Kluwer Academic Publisher, Boston, London, Dortrecht, 1999.

[36]. Dubois, D. and Prade, H., *Possibility theory - An approach to the computerized Processing of Uncertainty*, Plenum press, New York, 1988.

[37]. Dubois, D. and Prade, H., Default reasoning and possibility theory, *Artificial Intelligence 35*, 243-257, 1988.

[38]. Dubois, D. and Prade, H., On the combination of uncertain or imprecise pieces of information in ruled based systems, *International Journal of approximate Reasoning 2*, 65-87, 1988.

[39]. Dujmovic, J. J., Evaluation, Comparison and Optimization of hybrid computers using the theory of complex criteria, *Simulation of Systems*, Dekker, L. (Eds.), North Holland : Amsterdam, 553-566, 1976.

[40]. Dyckhoff, H. and Pedrycz, W., Generalized means as model of compensative connectivities, *Fuzzy Sets and Systems 14*, 143-154, 1984.

[41]. Elkan, C., The paradoxical success of fuzzy logic, IEEE Expert, 3-8, August 1994.

[42]. Fodor, J. and Roubens, M., Fuzzy Preference Modeling and Multicriteria Decision Support, Kluwer Academic Publisher : Dordrecht, 1994.

[43]. Fodor, J.C., Yager, R.R. and Rybalov, A., Structure of Uninorms, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 5*, 411-427, 1997.

[44]. Frank M.J., On the simultaneous associativity of F(x,y) and x+y-F(x,y), *Aequationes Math., 19* (1979), 194--226.

[45]. Grabisch, M., Fuzzy integrals in multicriteria decision making, *Fuzzy Sets and Systems 69*, 279-298, 1995.

[46]. Grabisch, M., k-additive fuzzy measures, *6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (*IPMU*), Granada, Spain, July 1996.

[47]. Grabisch, M., Nguyen, H.T. and Walker, E.A., Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference, *Kluwer Academics Publishers* : Dordrecht, 1995.

[48]. Hamacher, H., On Logical Connectives of Fuzzy Statements, *Proceedings of the 3rd. European Meeting Cybernetics and Systems*, Vienna, 1976.

[49]. Hibino, S. Rundenstein, E. A., A visual multimedia query language for temporal analysis of video data, *Multimedia Database Systems*, Kluwer Academic Publisher : Norwell, MA. 123-159, 1996.

[50]. Hjelsvold, R. and Midstraum, R., Modeling and querying video data, *Proceedings of the 20th VLDB Conference*, Santiago de Chile, 686-694, 1994.

[51]. Hohle, U., Probabilistic uniformization of fuzzy topologies, *Fuzzy Sets and Systems 1*, 1978.

[52]. Kasturi, R., Jain, R., Dynamic Vision, Computer Vision : Principles, Kasturi R., Jain, R., Editors, *IEEE Computer Society Press*, Washington, 1991.

[53]. Kelman, A. and Yager R.R., On the Application of a Class of MICA Operators, *International Journal of Uncertainty, Fuzziness and Knowledge based Systems, Vol. 3, No. 2*, 113-126, 1995.

[54]. Klement E. P., Mesiar R. and Pap, Triangular norms, *Trends in Logic, Studia Logica Library, Vol.8*, Kluwer Academic Publishers, 2000.

[55]. Klement, E. P., Mesiar, R. and Pap, E. On the relationship of associative compensatory operators to triangular norms and conorms. *International Journal of Uncertainty, Fuzziness and Knowledge based Systems, Vol. 4, No. 2*, 129-144, 1996.

[56]. Klement, E. P., A theory of fuzzy measures : a survey, *Fuzzy Information and Decision Processes*, edited by Gupta, M. and Sanchez, E., North Holland : Amsterdam, 59-66, 1982.

[57]. Klir, G. J. and Folger, T. A., Fuzzy Sets, Uncertainty and Information, *Prentice-Hall :* Englewood Cliffs, N.J., 1988.

[58]. Kolesarova A. and Komornikova M., Triangular norm-based iterative compensatory operators, *Fuzzy Sets and Systems 104,* 282-293, 1999.

[59]. Kolmogorov, A., Sur la notion de moyenne, *Atti delle Reale Accademia Nazionale dei Lincei Mem. Cl. Sci. Mat. Natur. Sez. 12,* 323-343, 1930.

[60]. Komornikova M., Generated aggregation operators. *Proceedings of EUSFLAT - ESTYLF'99, Palma de Mayorca*, 355-357,1999.

[61]. Ling C., Representation of associative function, *Publ. Math.,* Debrecen 12, (1965), 189-212.

[62]. Luce, R.D. and Krantz, D.H., Suppes, P., and Tversky, A., Foundations on measurements, *Academic Press,* New York, 1990.

[63]. Luhandjula, M. K., Compensatory operators in fuzzy linear programming with multiple objectives, *Fuzzy Sets and Systems 8,* 245-252, 1982.

[64]. Mackay, W.E., EVA : An experimental video annotator for symbolic analysis of video data, *SIGCHI Bulletin 21*, 68-71, 1989.

[65]. Mayor, G. and Trillas E., On the representation of some Aggregation functions, *Proceeding of ISMVL*, 111-114, 1986.

[66]. Menger, K., Statistical metrics, *Proc. Nat. Acad. Sci. U.S.A. 8*, 535-537, 1942.

[67]. Mesiar R. and Komorníková M., Aggregation Operators, *Proceeding of the XI Conference on applied Mathematics PRIM' 96*, Herceg D., Surla K. (eds.), Institute of Mathematics, Novi Sad, 193-211, 1997.

[68]. Mesiar, R. Compensatory operators based on triangular norms and conorms. *Proceeding of EUFIT'95*, Aachen, 131-135, 1995.

[69]. Nwosu, K. C., Thuraisingham, B. and Berra, P.B., *Multimedia Database Systems*, Kluwer Academic Publisher : Boston, 1996.

[70]. O'Hagan, M., Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic, *Proceedings of the 22nd Annual IEEE Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, Ca., 681-689, 1988.

[71]. Ovchinnikov, S., On Robust Aggregation Procedures, *Aggregation Operators for Fusion under Fuzziness.* Bouchon-Meunier B. (eds.), 3-10, 1998.

[72]. Prade H., Reasoning with fuzzy default values, *Proceedings of the 5th International Symposium on Multiple-Valued Logic*, Kingston, Ont.,

May, 191-197, 1985.

[73]. Ruiloba, R. and Joly P., Framework for Evaluation of Video-to-Shots Segmentation Algorithms : A Description Scheme for Editing Work, *Network Information Systems, Special Issue on Video*, 2000

[74]. Schweizer B. and Sklar A., Probabilistic metric spaces, *North Holland*, New York, 1983.

[75]. Schweizer, B. and Sklar, A., Statistical metric spaces, *Pacific J. Math. 10*, 313-334, 1960.

[76]. Seyrat C. and Detyniecki M. On Multimedia Indexing, *Second International Workshop on Knowledge Representation for Interactive Multimedia Systems - KRIMSII*, Trento - ITALY. June 1998.

[77]. Shahraray, B., Scene Change Detection and Content-Based Sampling of Video Sequences, *Digital Video Compression : algorithms and Technologies*, Rodriguez, Safranek, Delp, Eds., Proc. SPIE 2419, 2-13, Feb 1995.

[78]. Silvert, W., Symmetric summation : a class of operations on fuzzy sets, *IEEE Transactions on Systems, Man and Cybernetics 9,* 659-667, 1979.

[79]. Snodgrass, R., The temporal query language TQUEL, *ACM Transactions on Database Systems* 12, 247-298, 1987.

[80]. Sugeno, M., Theory of fuzzy integrals and its application, *Doctoral thesis*, Tokyo Institute of Technology, 1974.

[81]. Thiénot, C., Seyrat, C., Faudemay P. and Joly P., Status of the MPEG-7 parser, *MPEG Document M5767*, Mars 2000

[82]. Tsadiras, A., and Margaritis, K., The MYCIN certainty factor handling function as uninorm operator and its use as a threshold function in artificial neurons, *Fuzzy Sets and Systems, 93*, 263-274, 1998

[83]. Turksen, I. B., Interval-valued fuzzy sets and 'compensatory AND', *Fuzzy Sets and Systems 51*, 295-307, 1992.

[84]. Vitek, M., Fuzzy information and fuzzy time, *Proceeding of IFCA Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis,* Marseille, 159-162, 1983.

[85]. Weber S., A General Concept of Fuzzy Connectives, Negations and Implications *Fuzzy Sets and Systems 11*, 115-134, 1983.

[86]. Yager, R. R. and Kacprzyk J. (Eds.), *The Ordered Weighted Averaging Operators, Theory and Applications*, Kluwer Academic Publisher : Boston, Dordrecht, London, 1997.

[87]. Yager, R. R. and Rybalov, A., Full reinforcement operators in aggregation

techniques, *IEEE Transactions on Systems, Man and Cybernetics 28,* 757-769, 1998.

[88]. Yager, R. R., Fuzzy temporal methods for video multimedia information systems, *Journal of Advanced Computational Intelligence 1*, 37-45, 1997.

[89]. Yager, R. R., On a general class of fuzzy connectives, *Fuzzy Sets and Systems 4*, 235-242, 1980.

[90]. Yager, R. R., Aggregation operators and fuzzy systems modeling, *Fuzzy Sets and Systems 67,* 129-146, 1994.

[91]. Yager, R. R., and Rybalov, A., Uninorm Aggregation Operators, *Fuzzy Sets and Systems 80*, 111-120, 1996.

[92]. Yager, R. R., A new methodology for ordinal multiple aspect decisions based on fuzzy sets, *Decision Sciences 12,* 589-600, 1981.

[93]. Yager, R. R., On a class of weak triangular norm operators, *Information Science 96,* 47-78, 1997.

[94]. Yager, R. R., Non-monotonic set-theoretic operators, *Fuzzy Sets and Systems 42*, 173-190, 1996.

[95]. Yager, R. R., On ordered weighted averaging aggregation operators in multi-criteria decision making, *IEEE transactions on Systems, Man and Cybernetics 18*, 183-190, 1988.

[96]. Yager R., Detyniecki M. and Bouchon-Meunier B., A Context Dependent Method for Ordering Fuzzy Numbers Using Probabilities, *Technical Report MII-2012*, Iona College, New York, 2000.

[97]. Zabih, R., Miller, J., Mai, K., A Feature-based Algorithm for Detecting and Classifying Scenes Breaks, *Proceeding of ACM Multimedia 95,* 189-200, San Francisco, CA, November 1995.

[98]. Zadeh, L. A., A computational approach to fuzzy quantifiers in natural languages, *Computing and Mathematics with Applications 9,* 149-184, 1983.

[99]. Zadeh, L. A., Fuzzy Logic = Computing with Words, *IEEE Transactions on Fuzzy Systems* 59, 125-148, 1993.

[100]. Zhang, H.J., Kankanhalli, A., Smoiliar, S.W., Automatic Partitioning of Full-Motion Video, *Multimedia Systems Vol. 1 No1*, 10-28, 1993.

[101]. Zimmermann, H.-J. and Zysno, P., Latent connectives in human decision making, *Fuzzy Sets and Systems 4,* 37-51, 1980.