

# Implementation of an Inductive Fuzzy Classification Interpreter

A thesis submitted in partial satisfaction of the  
requirements for the degree of

Master of Arts

in

Information Management

at the

UNIVERSITY OF FRIBOURG

SWITZERLAND

by

Philippe Mayer

Supervised by:

Professor Andreas Meier

Michael Kaufmann

# Abstract

An interpreter for the Inductive Fuzzy Classification Language is presented. This thesis explores the advantages offered by fuzzy set theory over crisp sets for modelling sets without sharp cutoffs. How fuzzy logic can be used to model uncertainty is equally analysed. This thesis investigates a practical application of this in the form of fuzzy classification of customer data.

The Inductive Fuzzy Classification Process is presented. The induction step is based on deriving fuzzy restrictions from data whose membership functions are inferred from normalised likelihood ratios of target class membership.

In the context of this master thesis, an interpreter for the Inductive Fuzzy Classification was developed. This interpreter aims to support the inductive fuzzy classification process in its different stages.

**Keywords:** Inductive Fuzzy Classification, Fuzzy Logic, Fuzzy Sets



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Aim . . . . .	2
1.3	Report Structure . . . . .	3
<b>2</b>	<b>Fuzzy Sets</b>	<b>5</b>
2.1	Motivation for Fuzzy Sets . . . . .	5
2.2	Membership Functions . . . . .	6
2.3	Operations on Fuzzy Sets . . . . .	9
2.3.1	Classical Fuzzy Set Operators . . . . .	10
2.3.2	Algebraic Operators . . . . .	11
2.3.3	A Compensatory Fuzzy Set Connector . . . . .	13
2.4	Chapter Summary . . . . .	14
<b>3</b>	<b>Fuzzy Logic</b>	<b>17</b>
3.1	Linguistic Variables . . . . .	17
3.2	Fuzzy Propositions . . . . .	19
3.3	Approximate Reasoning . . . . .	20
3.4	Applying fuzzy logic to Contingency Valuation . . . . .	22
3.5	Chapter Summary . . . . .	23
<b>4</b>	<b>Fuzzy Classification</b>	<b>25</b>
4.1	Extending the relational Model . . . . .	26
4.2	Aggregation Operator . . . . .	28
4.3	Hierarchical Fuzzy Classification . . . . .	30
4.4	The Fuzzy Classification and Query Language . . . . .	32
4.4.1	fCQL syntax . . . . .	32
4.4.2	Examples of fCQL Queries . . . . .	33
4.5	The fCQL toolkit . . . . .	34
4.6	Summary of Findings . . . . .	36

<b>5</b>	<b>Inductive Fuzzy Classification</b>	<b>39</b>
5.1	Theory of Inductive Fuzzy Classification . . . . .	40
5.1.1	Data Mining and Machine Learning . . . . .	40
5.1.2	Inductive Fuzzy Classification . . . . .	44
5.2	The Inductive Fuzzy Classification Process . . . . .	45
5.2.1	Data Preparation . . . . .	46
5.2.2	Attribute Selection . . . . .	47
5.2.3	Induction of Membership Functions . . . . .	47
5.2.4	Univariate Fuzzy Classification . . . . .	49
5.2.5	Multivariate Fuzzy Classification . . . . .	50
5.3	Summary of Findings . . . . .	51
<b>6</b>	<b>Inductive Fuzzy Classification Language</b>	<b>53</b>
6.1	Motivation and Objectives for iFCQL . . . . .	53
6.2	iFCQL Syntax . . . . .	54
6.2.1	Attribute selection . . . . .	55
6.2.2	Membership Function Induction . . . . .	55
6.2.3	Univariate Fuzzy Classification . . . . .	56
6.2.4	Multivariate Fuzzy Classification . . . . .	56
6.2.5	Model evaluation . . . . .	56
6.3	The iFCQL Interpreter . . . . .	57
6.3.1	Design Requirements . . . . .	57
6.3.2	Client Architecture . . . . .	58
6.4	Client Implementation . . . . .	60
6.4.1	Data Preparation . . . . .	61
6.4.2	Membership function induction . . . . .	61
6.4.3	Univariate Classification . . . . .	65
6.4.4	Multivariate Classification . . . . .	66
6.5	Summary of Finding . . . . .	67
<b>7</b>	<b>Conclusion</b>	<b>69</b>
7.1	Summary of Findings . . . . .	69
7.1.1	Fuzzy Sets . . . . .	69
7.1.2	Fuzzy Classification . . . . .	70
7.1.3	Inductive Fuzzy Classification Process . . . . .	70
7.1.4	Inductive Fuzzy Classification Language . . . . .	70
<b>A</b>	<b>iFCQL Interpreter Instruction Manual</b>	<b>71</b>
A.1	System requirements . . . . .	71
A.2	Installation Instruction . . . . .	71

# List of Figures

2.1	A graphical representation of the crisp set <i>Teenager</i> . . . . .	8
2.2	A graphical representation of the fuzzy set <i>Teenager</i> . . . . .	9
3.1	Terms of the linguistic variable age. Taken from [1] . . . . .	18
3.2	Truth value of the proposition <i>the glass is full</i> , defined by the membership function of a fuzzy set . . . . .	20
4.1	Classification space determined by turnover and behaviour	27
4.2	Hierarchy of credit worthiness taken from [33] . . . . .	31
4.3	Architecture of the <i>fcQL</i> toolkit. Taken from [23] . . . . .	35
4.4	The <i>fcQL</i> toolkit graphical user interface . . . . .	36
5.1	the inductive fuzzy classification process . . . . .	46
6.1	Parts of the IFP that have been implemented . . . . .	57
6.2	Overview of the <i>fcql</i> Client Server Architecture . . . . .	58
6.3	The <i>ifCQL</i> Client Architecture . . . . .	59



# Chapter 1

## Introduction

### 1.1 Motivation

Fuzzy set theory was first proposed by Lofti Zadeh as a precise mathematical tool for dealing with classes of objects without precisely defined criteria of membership [31]. Since then fuzzy set theory and fuzzy logic has been applied to a wide range of domains including fuzzy control theory and fuzzy classification. As part of a PhD thesis [28], the Information Systems group of the University of Fribourg specified the Fuzzy Classification and Query Language (FCQL) as well as an accompanying toolkit. This toolkit has been used successfully, particularly for analysis of online customer profiles [23][16].

The FCQL and the accompanying toolkit can be further developed in a number of way. One possible further development is in the way that fuzzy sets, for instances customer classes, are defined. Currently the membership function for elements to sets must be defined manually using mEdit, a membership function editor integrated into the FCQL toolkit [19]. It would

be interesting to automate this procedure via a mechanism that predicts the membership elements to the different sets.

In order to apply predictive fuzzy classification, an inductive fuzzy classification, in which the membership value of elements to a target set are induced using machine-learning techniques has been proposed[14]. This approach has been tested successfully by Postfinance for an online marketing campaign.

The next step is to automate this procedure. To this effect, an extension of FCQL, known as the inductive Fuzzy Classification Language (iFCQL) is being proposed.

## 1.2 Aim

The aim of this project was to develop an interpreter for iFCQL, which would support the Inductive fuzzy classification process in all its stages. These stages are as follows [14]:

1. Preparing the data using SQL
2. At this stage, the training set is created and target variables are defined.
3. Feature selection
4. Inducing membership functions
5. Single dimensional fuzzy classification
6. Multi dimensional fuzzy classification
7. Evaluation

The focus was on the implementation of steps four, five and six the core of the interpreter. The result should be a prototype that can be tested for

## 1.3 Report Structure

Chapter 2 briefly discusses fuzzy set theory that was first introduced by Lofti Zadeh. First, the motivation for fuzzy sets and their advantages over crisp sets is presented. Next, membership functions that allow elements to have varying degrees of membership to different sets are shown. Finally, some operations that can be performed on fuzzy sets are given.

Chapter 3 presents fuzzy logic, introduced by Lofti Zadeh, which is based based on fuzzy sets. First, linguistic variables, that can be used to describe concepts in linguistic rather than numeric terms are introduced. Next, fuzzy propositions, which allow for varying degrees of truth are shown. Approximate reasoning, or reasoning with fuzzy propositions and fuzzy variables, is then presented. Finally, fuzzy logic applied to contingency valuation is shown as an example of real-world problems that can be addressed by fuzzy logic.

Chapter 4 presents fuzzy classification. First, fuzzy classes, based on fuzzy set theory will be introduced. Next, a two dimensional fuzzy classification, made by combining two dimensions via an aggregation is shown. After this, a hierarchical fuzzy classification that combines the precision of a multidimensional classification with the legibility of a two dimensional classification is shown. Finally, an overview of the fuzzy Classification and Query Language and the accompanying toolkit that can be used to query fuzzy classification schemas is given.

Chapter 5 describes the inductive fuzzy classification process proposed by Kaufmann [14]. First an overview of data mining and machine learning techniques is given. Next how these techniques can be used for inductive fuzzy classification is explained. The aim of such a classification is to induce the membership functions of elements in a target class. Finally a systematic approach for achieving such a classification is described.

Chapter 6 describes the fuzzy classification language as well an interpreter for this language that was developed in the context of this master thesis. First, the iFCQL syntax is given, then an overview of the interpreter is presented. The presentation of the interpreter consists of an an architectural overview followed by a detailed description of how each iFCQL is translated into SQL and executed on a database.



# Chapter 2

## Fuzzy Sets

As a first step towards designing and implementing an inductive fuzzy classification interpreter, we will briefly discuss fuzzy set theory. First, the motivation for fuzzy sets and their advantages over crisp sets is presented. Next, membership functions that allow elements to have varying degrees of membership to different sets are shown. Finally, some operations that can be performed on fuzzy sets are given. The most important of these is the fuzzy set connector, which will be used extensively in the interpreter for multivariant fuzzy classification.

### 2.1 Motivation for Fuzzy Sets

Fuzzy set theory was first proposed by Lofti Zadeh as a precise mathematical tool for dealing with classes of objects without precisely defined criteria of membership [31]. The motivation for Zadeh's work was the observed gap between natural human reasoning and classical set theory.

Many classes encountered in the real world cannot be defined in terms of crisp sets. Human beings tend to see the world in terms of concepts such as a *young* person, a *tall* man or *warm* weather, which cannot be defined in terms of classic sets. Two features of crisp sets and classic logic that hinder us from expressing the ideas in terms of sets are:

1. The fact that sets have a sharp cutoff [1]
2. The law of the excluded middle which states that an element is either a member of a set or it is not.[32]

The first of these properties implies that for each set a sharp limit must be defined. In the case of the notion of *young* persons, an age must be specified that separates *young* people from *old* people. Defining such an age can be problematic as every human being has a different conception of the term *young*. Insurance companies for instance consider anybody under 25 is to be *young* whereas pension funds place the limit at 65.

The second property means that all members of a set are members to the same degree. In our example a 24 year old person and a 14 year old person are both *young* even though intuitively humans would consider one to be younger than the other.

Fuzzy set theory allows for a more intuitive representation of reality by replacing sharp cutoffs with gradual membership functions. Because of this, fuzzy sets can be seen as being a natural extension of classical sets [28]. This gradual degree of membership is expressed by *membership functions*, presented in the next section.

## 2.2 Membership Functions

In classical set theory, the membership of elements in a set is defined in binary terms: an element either belongs or does not belong to the set. A classic set  $A$  is defined by an *indicator function*  $\mu_A(x)$  defined by [1]:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \quad (2.1)$$

If an element  $x$  has an indicator value of 1 in set  $A$ , this means that it is part of this set. If the associated indicator value is 0, the element  $x$  is not in the set  $A$ . The law of the excluded middle applies, any element is either in set  $A$  or it is not.

Fuzzy sets, on the other hand, to allow elements to have varying degrees of membership, expressed by a *membership function* over a continuous domain between 0 and 1 [31, 26]. A fuzzy subset  $A$  of the universal set  $U$  (containing all elements) is defined as a set of ordered pairs

$$\{(x_i, \mu_A(x_i))\} \quad (2.2)$$

where  $x_i \in U$ ,  $\mu_A : U \rightarrow [0, 1]$  is the *membership function* of  $A$  and  $\mu_A(x) \in [0, 1]$  is the degree of membership of  $x$  in  $A$  [28].

All elements are therefore members of all sets to a degree varying between 0 and 1. This allows for a simple definition of one element being member to a greater degree to a given set than another element. If we consider the fuzzy set *old*, with membership depending on a persons age for instance; several people can belong to this group, with some being more *old*, older, than others.

Consider the example of a company marketing products to teenagers. If customer classification were done using a sharp classification, the ages that constitute sharp upper and lower boundaries of our age group would have to be defined. Intuitively, we would consider 13 and 19 as being suitable lower and upper boundaries for this set because, mathematically, these two ages form the limits of the –teen years.

Formally, we would write the indicator function of an age  $x$  as

$$\mu_{teenager}(age) = \begin{cases} 1 & \text{if } 13 \leq age \leq 19 \\ 0 & \text{if } x < 13 \text{ or } age > 19 \end{cases}$$

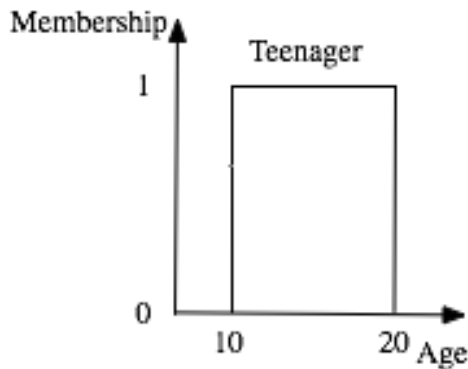


Figure 2.1: A graphical representation of the crisp set *Teenager*

An indicator value  $\mu_{teenager}(age)$  of 1 indicates that for this value of *age*, a person can be considered to be a teenager and for a value of 0, a person is not a teenager. Graphically this is represented in Figure 2.1. As can be seen, on his or her 13<sup>th</sup> birthday, a person suddenly becomes a teenager and on the day of his or her 20<sup>th</sup> birthday, he or she then immediately leaves this set.

While such a definition has the virtue of being mathematically precise, it does have some practical limitations. A company preparing an advertising campaign aimed at teenagers, would be ill advised to ignore anyone aged 12 or 20. Similarly, it would not be wise to treat all potential customers in the same way, regardless of their age.

A more natural way of representing the set of teenager would be to allow a person to gradually enter the set and then again gradually leave it. In this case the company could decide that anybody under 10 and over 20 cannot be a teenager, so these people would have a membership degree of 0. Anybody between these ages would have a membership value that gradually increases as they approach the center of the set, defined for instance, as being the age of 15.

Formally, we would write the membership function of an age  $x$  as

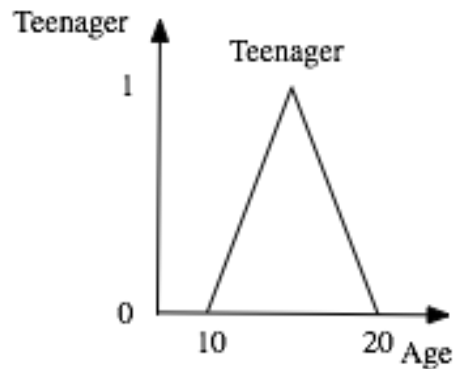


Figure 2.2: A graphical representation of the fuzzy set *Teenager*

$$\mu_{teenager}(age) = \begin{cases} \frac{x}{5} - 2 & \text{if } 10 \leq age \leq 15 \\ -\frac{x}{5} + 4 & \text{if } 15 < age \leq 20 \\ 0 & \text{if } age < 10 \text{ or } age > 20 \end{cases}$$

With this definition of the set *teenager*, a person under the age of 10 or over the age of 20 will not be a member. Anybody between these ages is a member to a varying degree with those aged 15 member to a degree of 1. Graphically this set is shown in Figure 2.2.

## 2.3 Operations on Fuzzy Sets

Having introduced the concept of fuzzy sets, some of the operations that can be performed on these sets will now be shown. The *set complement*, *set intersect* and *set union operators* from classical set theory can be generalised and applied to fuzzy sets. While different definitions of this operator exist [28], we will base ourselves on the definition proposed in Zadeh's original paper [31] and on the compensating operator proposed by Zimmermann [33].

### 2.3.1 Classical Fuzzy Set Operators

#### Set Complement

The *complement* of a fuzzy set is defined as being 1 minus the membership degree of all elements in the univers to this set. Formally this is written as follows:

$$\neg A = 1 - \mu_A(x), x \in U \quad (2.3)$$

#### Set Union and Set Intersection

For the *set intersect* and *set union operators*, Zadeh suggested the use of the *min* and the *max* operator respectively. The advantage of using these operators is that they are easy to understand and fast to compute [28].

The *intersection* of two fuzzy sets A and B is defined as follows:

$$A \cap B = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)), x \in U \quad (2.4)$$

The *union* of two fuzzy sets A and B is defined as follows

$$A \cup B = \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x)), x \in U \quad (2.5)$$

To understand how these operators work, consider the following example: Given the Univers U and the fuzzy sets A and B defined as follows:

$$\begin{aligned} U &= \{a, b, c, d\} \\ A &= \{(a, 0), (b, 0.2), (c, 0.6), (d, 1)\} \\ B &= \{(a, 0.1), (b, 0.3), (c, 0.4), (d, 0.9)\} \end{aligned}$$

The the set complement of A is

$$\neg A = \{(a, 1), (b, 0.8), (c, 0.4), (d, 0)\}$$

The intersection of A and B is

$$A \cap B = \{(a, 0), (b, 0.2), (c, 0.4), (d, 0.9)\}$$

The union of A and B is

$$A \cup B = \{(a, 0.1), (b, 0.3), (c, 0.6), (d, 1)\}$$

### 2.3.2 Algebraic Operators

A number of algebraic operators have been defined for fuzzy sets. In this section, we will restrict ourselves to the algebraic product and the algebraic sum which can be used to model the set intersection and the set union respectively.

#### Algebraic Product

The algebraic product two sets is defined as follows [33]

$$A \cdot B = \{x, \mu_A(x) \cdot \mu_B(x)\} \tag{2.6}$$

### Algebraic Product Set Intersection

The algebraic product can be used to model fuzzy set intersections. The intersection of two fuzzy sets  $A$  and  $B$  is simply defined as being the product of the two sets. Formally it is defined as [33]

$$A \cap B = A \cdot B = \{x, \mu_A(x) \cdot \mu_B(x)\} \quad (2.7)$$

The algebraic product set intersection can be generalised for  $n$  sets. The membership degree of an element  $x$  to the intersection of  $n$  fuzzy sets  $A_i$  is defined by the product of the individual membership degrees  $\mu_{A_i}(x)$  of  $x$  to each set  $A_i$ .

$$\mu_{\cap_{i=1}^n A_i}(x) = \prod_{i=1}^n \mu_{A_i}(x) \quad (2.8)$$

### Algebraic Sum

The algebraic sum of two sets  $A$  and  $B$  is defined as [33]

$$A + B = \{x, \mu_{A+B}(x)\} \quad (2.9)$$

where

$$\mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

### Algebraic Sum Union

Just as the algebraic product can be used to model fuzzy set intersection, the algebraic sum can be used to model the fuzzy set union. The union of

two fuzzy sets A and B is simply defined as the product of the two sets. Formally it is defined as [33]

$$A \cup B = A + B = \{x, \mu_{A+B}(x)\} \quad (2.10)$$

The algebraic sum set union can also be generalised for  $n$  sets. The membership degree of an element  $x$  to the union of  $n$  fuzzy sets  $A_i$  is defined by 1 minus the product of 1 minus the individual membership degrees  $\mu_{A_i}(x)$  of  $x$  to each set  $A_i$ .

$$\mu_{\cup_{i=1}^n} = 1 - \prod_{i=1}^n (1 - \mu_{A_i}(x)) \quad (2.11)$$

### 2.3.3 A Compensatory Fuzzy Set Connector

Zimmerman suggested the use of the compensatory and or *Gamma-operator* as a way of modelling how humans make complex decisions more accurately [33]. Zimmermann observed that, depending on the context, human beings combine the membership degrees of an element to different classes differently. Empirical studies conducted by Zimmermann suggest that the Gamma-operator provides a more realistic model for combining classes than the algebraic operators.

An example of such a complex decision is determining a person's credit-worthiness. The fuzzy set *credit worthiness* can for instance be an aggregation of the classes instance *financial credit worthiness* and *personal credit worthiness*. These sets can, depending on the operator used to combine them, vary in compensation. If the *min* operator is used to aggregate these sets, the smallest membership value is taken and no compensation occurs. If the *max* operator is used, the biggest value is taken and full compensation occurs. The first case would result in a wealthy person being refused a credit because in the past they failed to pay of a small loan and the second would result in a penniless client being offered a large loan because they

always paid off their, until now small, debts. Both situations are clearly absurd. The Gamma-operator is a better choice, because it allows for varying degrees of compensation depending on the context.

The Gamma-operator combines the algebraic product and the algebraic sum weighed by a parameter. The membership degree of an element  $x$  to the combination of  $n$  fuzzy sets  $A_i$  is defined as follows [33]:

$$\mu_{\Gamma_{\gamma=0}}^n = \left( \prod_{i=1}^n \mu_{A_i}(x) \right)^{(1-\gamma)} \left( \prod_{i=1}^n (1 - \mu_{A_i}(x)) \right)^\gamma \quad (2.12)$$

By varying the  $\gamma$  parameter, this operator incorporates more of the algebraic product, defined in Equation 2.8, or more of the algebraic sum, defined in Equation 2.11. For  $\gamma = 0$  this operator is equal to the algebraic product, and for  $\gamma = 1$  this operator is equal to the algebraic sum.

## 2.4 Chapter Summary

This chapter briefly introduced fuzzy set theory. Fuzzy set gives us a precise mathematical tool for dealing with many classes encountered in the real world which don't sharply defined cut-offs. For example, the cut-off between tall and short people. This is achieved by allowing elements to have varying degrees of membership, expressed by a membership function over a continuous domain between 0 and 1 to different sets. This means that an element can be a member of different sets at the same time and can that some elements can be member to a greater degree than others of a set.

We presented a practical application for this as being a market study of teenager. Rather than excluding teenage customers over the age of 19 and under the age of 12, we can create a model of a customer group where membership is a function of the different members' age. Customers then gradually enter and leave this group.

Finally, a variety of mathematical operations on fuzzy sets was presented. The most important of these is the compensatory, or Gamma operator which allows us to model complex human decision making. This allows us to create new sets by combining membership degrees to different sets. An example given was the set of credit worthy people. This set is a combination of the sets financial credit worthiness and personal credit worthiness. This example will be further discussed in Chapter 4.



# Chapter 3

## Fuzzy Logic

The following chapter presents fuzzy logic, introduced by Lofti Zadeh, which is based based on fuzzy sets. First, linguistic variables, that can be used to describe concepts in linguistic rather than numeric terms are introduced. Next, fuzzy propositions, which allow for varying degrees of truth are shown. Approximate reasoning, or reasoning with fuzzy propositions and fuzzy variables, is then presented. Finally, fuzzy logic applied to contingency valuation is shown as an example of real-world problems that can be addressed by fuzzy logic.

### 3.1 Linguistic Variables

Fuzzy logic, unlike probability theory, allows fuzzy quantifiers and fuzzy probabilities [30]. This allows users to work at a semantic level using linguistic variables [16], thereby allowing ergonomic data modelling. A customer can be described a being *loyal* or *likely* to pay on time. Such fuzzy

quantifiers are especially important in management decisions dealing with uncertainty and with values that can not be quantified [7, 1]

*Linguistic variables* are variables whose value is expressed in natural language [1]. Certain variables such as for instance the *loyalty* of a customer can not be assigned a numerical value. An example from everyday life, described by Bojadiev, is that of a person's *age* [1]. While it is possible to give a numerical answer to this question, people frequently describe themselves as young, old or middle aged. This is further complicated by the fact that one person may be considered to be young, yet someone of exactly the same age old.

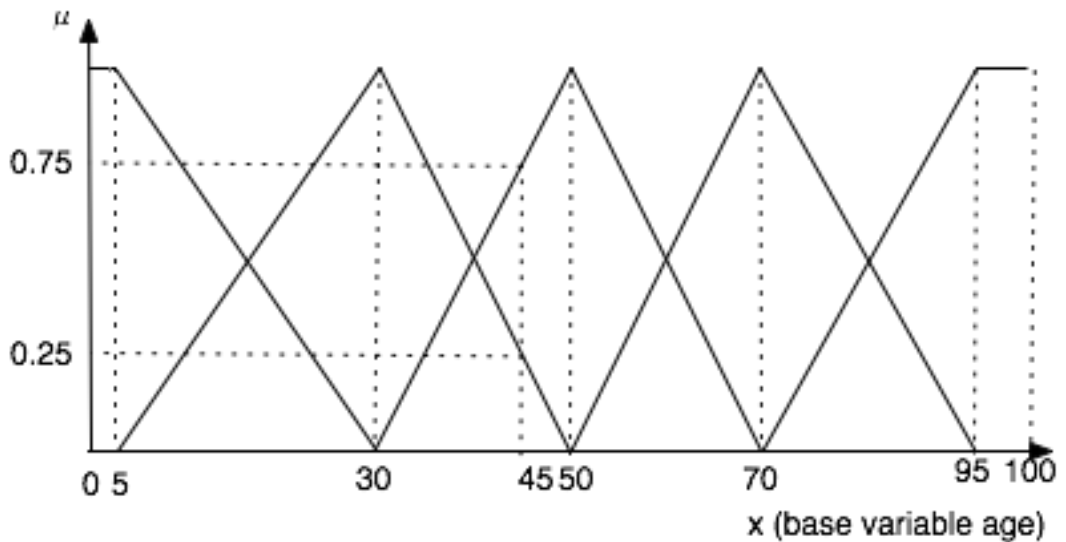


Figure 3.1: Terms of the linguistic variable age. Taken from [1]

Bojadiev [1] for instance, describes the linguistic variable *age* by the linguistic terms, very young, young, middle aged, old and very old. This is shown in Figure 3.1.

The membership functions for the terms are:

$$\mu_{very\ young}(x) = \begin{cases} 1 & \text{if } 1 \leq x \leq 5 \\ \frac{30-x}{25} & \text{if } 5 \leq x \leq 30 \end{cases}$$

$$\mu_{young}(x) = \begin{cases} \frac{x-5}{25} & \text{if } 5 \leq x \leq 30 \\ \frac{50-x}{20} & \text{if } 30 \leq x \leq 50 \end{cases}$$

$$\mu_{middle\ aged}(x) = \begin{cases} \frac{x-30}{20} & \text{if } 30 \leq x \leq 50 \\ \frac{70-x}{20} & \text{if } 50 \leq x \leq 70 \end{cases}$$

$$\mu_{old}(x) = \begin{cases} \frac{x-50}{20} & \text{if } 50 \leq x \leq 70 \\ \frac{95-x}{25} & \text{if } 70 \leq x \leq 95 \end{cases}$$

$$\mu_{middle\ aged}(x) = \begin{cases} \frac{x-70}{25} & \text{if } 70 \leq x \leq 95 \\ 1 & \text{if } 95 \leq x \leq 100 \end{cases}$$

If we take a person aged 45 years old, for instance, we see that this person is young to the degree of 0.25 and middle aged to the degree of 0.75. This particular person can therefore be described as young (degree 0.25) and middle aged (degree 0.75) at the same time.

## 3.2 Fuzzy Propositions

One of the theorems of classical propositional logic is the *law of the excluded middle* which states that a proposition is either true or false [32]. Statements are therefore assigned a truth value in the set  $\{0, 1\}$ . Just like fuzzy sets are an extension of classical sets that allow elements to have varying degrees of membership, fuzzy propositional logic is an extension of classical propositional logic that allows statements to be assigned varying degrees of truth. More precisely, a proposition  $p$  is assigned a truth value  $T(p)$  in the range  $[0, 1]$ . If  $T(p)$  is 0, then  $p$  is false; if  $T(p)$  is 1, then  $p$  is true; for all values of  $0 < T(p) < 1$ ,  $p$  is true to a degree expressed by  $T(p)$ . Formally Zadeh defined a fuzzy proposition as [29]:

$$p \equiv xisP \tag{3.1}$$

where  $x \in X$  is an element of a universe of discourse  $X$ ,  $P$  is a linguistic

term modeled by a fuzzy set, and a membership function  $\mu P(x)$  denotes the degree of membership of  $x$  in the fuzzy subset  $P$ .

Both degrees of truth and probability range between 0 and 1 and therefore appear to be similar in nature. Conceptually, however, they are fundamentally different. Truth represents membership of a fuzzy set, not likelihood of some event or condition as in probability theory.

As an example, consider a 100-ml glass of water and the fuzzy proposition

$p_1 \equiv$  *the glass is full*

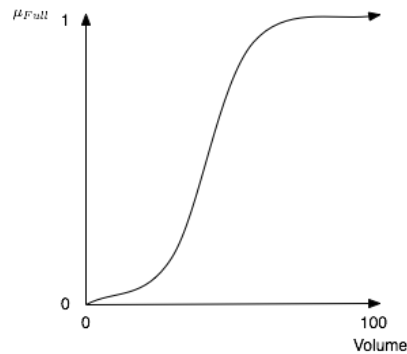


Figure 3.2: Truth value of the proposition *the glass is full* , defined by the membership function of a fuzzy set

In this case, the concept *full* is a linguistic term, whose membership function is defined over the contents of the glass, varying between 0 and 100. The truth value of the fuzzy proposition *the glass is full* is defined by the fuzzy set *full* whose membership function is shown in Figure 3.2.

### 3.3 Approximate Reasoning

Zadeh characterised fuzzy logic as being a logic of fuzzy or approximate reasoning with the following distinctive characteristics [29]:

1. Fuzzy truth values expressed in linguistic terms such as very true, true, not very true, false.
2. Imprecise truth tables
3. Rules of inference whose validity is approximate rather than exact

An example of approximate reasoning is the following:

$A_1$ : *Most men are vain*  
 $A_2$ : *Socrates is a man*  
 $\Rightarrow A_3$ : *It is likely that Socrates is vain*  
 OR  
 $A'_3$ : *It is very likely that Socrates is vain*

In this example, both  $A_3$  and  $A'_3$  are admissible approximate consequences of  $A_1$  and  $A_2$ . The exact degree of approximation depends on the definition of the fuzzy terms *most*, *very* and *likely* as fuzzy subsets of their respective universes of discourse.

These fuzzy propositions can be combined using the fuzzy operator *not*, and *and* [29]. Consider the propositions  $p$  and  $q$ , and  $v(p)$  and  $v(q)$  their respective truth values.

Then the *negation* of  $p$  is:

$$v(\neg p) \equiv 1 - v(p) \quad (3.2)$$

The *disjunction* of  $p$  and  $q$  by union of the corresponding sets is:

$$v(p \vee q) \equiv \mu_{P \cup Q}(x) \quad (3.3)$$

The *conjunction* of  $p$  and  $q$  by intersection of the corresponding sets is:

$$v(p \wedge q) \equiv \mu_{P \cap Q}(x) \quad (3.4)$$

## 3.4 Applying fuzzy logic to Contingency Valuation

Fuzzy logic can be particularly useful for decision making when probabilities and utilities are poorly defined [30]. One example of such a situation is the valuation of non-market resources, such as environmental preservation. While people derive a utility from these, assigning them a value is difficult because nobody pays for them. Consider for example a beautiful mountain. Everyone agrees that this is something agreeable to look at, yet nobody is willing to pay just for the privilege of seeing it. Contingency valuation (CV) is one method that was developed to assign a value to such resources.

The CV uses surveys to determine people's preferences for public goods by finding out their willingness to pay for them. In the absence of markets for these goods, it creates a hypothetical market for them in which people can bid for them. An interview conducted for such a survey generally consists of three parts: [2]

1. A description of the goods being valued and the hypothetical circumstances under which they can be purchased
2. Questions to determine respondents' willingness to pay
3. Questions about respondents' characteristics, for example their age and income, and questions to their preferences relevant to the goods being valued.

Some believe that CV is deeply flawed because respondents can give protest answers or ignore income constraints [9]. CV assumes that respondents can state with absolute certainty how much they are willing to pay for a certain resource. In reality, respondents may be uncertain about their preferences for a number of reasons [24].

1. Respondents may not be sufficiently well acquainted with the alternatives they are asked to value.

2. Respondents may be genuinely uncertain about their preferences because they have never had to make such a decision before.
3. Respondents might never fully know their own preferences.

One improvement is to model the underlying vagueness of preferences and the imprecision of what is being valued using fuzzy logic [24]. In this new model, it is assumed that for each good, respondents know with absolute certainty an upper maximum, and a lower minimum that they are willing to bid. In between these two values, respondents' preferences are assumed to be vague.

This method was used for evaluating the results of a CV of Swedish residents conducted in 1992 [24]. The survey asked respondents whether they would be willing to pay a given amount to “to continue to visit, use and experience the forest environment as they usually do”. The bid values were one of the following amounts: 50, 100, 200, 400, 700, 1000, 2000, 4000, 8000 and 16000 SEK. After answering yes or no to a certain bid, respondents were asked a number of follow-up questions to determine their confidence in their response.

### 3.5 Chapter Summary

This chapter introduced fuzzy logic, which is based on fuzzy set theory. Analogous to how fuzzy set theory extends crisp set theory via continuous membership functions, fuzzy logic extends binary logic via fuzzy truth values. This allows for linguistic variables and fuzzy propositions. Linguistic variables are variables whose value is expressed in natural language. An example of such a variable is the concept of young people. Rather than defining people as being under or over a certain age, a natural variable can be used.

Fuzzy propositions are propositions that have a degree of truth between 0 and 1. The example of given is the statement the glass is full. The degree of truth of this statement is a function of the amount of water in the glass.

Linguistic variable and fuzzy propositions can then be used for approximated reasoning. This allows us to defined stamen that can for instance be very true, occasionally true or occasionally false rather than simply being true or false. An application of fuzzy logic is contingency valuation. Contingency valuation (CV) is a method that was developed to assign a value to non-market resources. These are resources from which people derive benefits without being able to assign a value to them. The CV uses surveys to determine people's preferences for public goods by finding out their willingness to pay for them. In the absence of markets for these goods, it creates a hypothetical market for them in which people can bid for them. An example being the environment. We presented a case study conducted in Sweden were CV was used to evaluate peoples' attachment to the forest environment.

# Chapter 4

## Fuzzy Classification

This chapter briefly summarises the key findings of the Information Systems Research Group at the University of Fribourg (Switzerland) in the domain of fuzzy classification. First, fuzzy classes, based on fuzzy set theory will be introduced. Next, a two dimensional fuzzy classification, made by combining two dimensions via an aggregation will be shown. After this, a hierarchical fuzzy classification that combines the precision of a multidimensional classification with the legibility of a two dimensional classification will be shown. Finally, an overview of the fuzzy Classification and Query Language and the accompanying toolkit that can be used to query fuzzy classification schemas will be given. Throughout this chapter a simple example of fuzzy customer classification will be used to illustrate the presented theory.

## 4.1 Extending the relational Model

In 1970 E.F. Codd introduced the relational model for databases [4]. He proposed a model based on predicate logic and set theory that isolates a database user from how data is physically stored. Data is organised into two-dimensional tables called relations. The columns of this table hold the attributes of a stored object. Each row, called a tuple, represents a single stored object's attributes. In order to reduce data redundancy and possible data anomalies, a number of normal forms were developed [13]. Respecting these does however entail a number of restrictions [33]

1. Queries return sharp results. This comes from the fact that relational algebra is based on classical predicate calculus. The selection operator  $\sigma_F(R)$  for instance will return all tuples  $R$  for which  $F$  is true and no other tuple.
2. The data stored in the database is precise. This is a result of the first normal form which requires all attributes to be atomic. This excludes attributes composed of a set and a membership value.

These restrictions mean that storing fuzzy or ambiguous data in a relational database and performing fuzzy queries is impossible. To overcome these limitations, a *context model* was proposed by Chen [3] and extended by Meier et al. [25].

In this model every attribute  $A_j$ , defined over a domain  $D(A_j)$ , is assigned a context  $K(A_j)$ . This context  $K(A_j)$  is a partitioning of the domain  $D(A_j)$  into equivalence classes. The set of attributes  $A = (A_1, \dots, A_n)$  with their respective contexts  $C = (C_1(A_1), \dots, C_n(A_n))$  forms a relational database schema with contexts  $R(A, C)$  [22].

To illustrate this, an example that will be used throughout this chapter will be shown. Consider the case of an online shop that wishes to classify its customers on the basis of turnover generated by individual customers and customers' willingness to pay within a reasonable timeframe. In our example, we will define two contexts

1. *turnover*, discreet numerical values between 0 and 1000 representing the amount of money spent by the customer and

	excellent    good	sufficient    bad	D(behaviour)
1000 500	C1	C2	
499 0	C3	C4	
			D(turnover)

Figure 4.1: Classification space determined by turnover and behaviour

2. *behaviour*, linguistic variables in the set  $\{bad, sufficient, good, excellent\}$ .

The result is the set of attributes  $A = \{customer, turnover, behaviour\}$  and the set of contexts  $C = \{C(customer), C(turnover), C(behaviour)\}$ . To facilitate understanding of the resulting classification, the domain over the attribute  $D(turnover)$  is divided into the equivalence classes  $[0, 499]$  representing poor turnover and  $[500, 1000]$  representing high turnover. The domain of the attribute  $D(behaviour)$  is split into the equivalence classes  $\{bad, sufficient\}$  representing poor behaviour and  $\{good, excellent\}$  representing good behaviour.

Under such a classification consider the following customers:

customer	turnover	behaviour
Huber	560	bad
Mueller	500	good
Sieber	450	sufficient
Suter	900	excellent

The resulting classification is divided into the equivalence classes C1 to C4 as shown in Figure 4.1. The class C1 represents premium customers who spend a lot and pay on time. C2 represents customers who generate a high turnover but have problems paying on time, C3 represents those customers

who pay on time but spend little and finally C4 represents customers who generate little turnover and fail to pay on time. In our example, Mueller and Suter belong to class C1, Huber belongs to class C2 and Sieber belongs to class C4.

If this classification is used as a basis for preferential customer treatment, the advantage of using a fuzzy classification immediately becomes obvious. Under a sharp classification, Suter would receive identical treatment to Mueller even though he generates a higher turnover and has a better payment behaviour. Mueller and Sieber would receive vastly different treatments even though they generate similar turnover and have similar payment behaviour. A fuzzy classification on the other hand allows for a gradual membership degree to each set, with customers assigned to multiple sets. This means that a customer's value to the company can be calculated precisely and they can be rewarded accordingly [15].

## 4.2 Aggregation Operator

The raw membership of an object  $O_i$  to a class  $C_k$ , written as  $M_{raw}(O_i|C_k)$  can be evaluated by aggregating the linguistic variables that describe the class. In our example, we would calculate the membership of customers to the classes C1 to C4 by aggregating the  $\mu_{turnover}$  and the  $\mu_{behaviour}$  membership functions for each set.

As described in Section 2.3 a variety of operators exist for aggregating sets. As previously stated, the *min* value, by taking the smallest membership value to one of the aggregated sets, offers no compensation. This means that in our example customers will only be judged by their worst behaviour. Similarly the *max* operator, provides full compensation and as a result customers are only judged by their best behaviour. The Gamma operator on the other hand offers a much more flexible aggregation via varying degrees of compensation.

Continuing our example, let us define  $\mu_{high\_turnover}$  proportional to turnover and  $\mu_{attractive\_behaviour}$  stepwise, 1 for excellent, 0.6 for good, 0.3 for sufficient

and 0 for poor. We will likewise define  $\mu_{low\_turnover}$  and  $\mu_{unattractive\_behaviour}$  as the inverse of these.

This gives the following membership degrees:

Customer	$\mu_{high\_turnover}$	$\mu_{attractive\_behaviour}$
Huber	0.56	0
Mueller	0.5	0.6
Sieber	0.45	0.3
Suter	0.9	1

We can now calculate the value of each member to each set using a  $\gamma$  value of 0.5. This gives us:

$$\begin{aligned}
 M_{raw}(Huber|C1) &= (\mu_{high\_turnover}(560) * \mu_{attractive\_behaviour}(bad))^{0.5} \\
 &\quad * (1 - ((1 - \mu_{high\_turnover}(560)) * (1 - \mu_{attractive\_behaviour}(bad))))^{0.5} \\
 M_{raw}(Huber|C1) &= (0.56 * 0)^{0.5} * (1 - ((1 - 0.56) * (1 - 0)))^{0.5} \\
 M_{raw}(Huber|C1) &= 0
 \end{aligned}$$

$$\begin{aligned}
 M_{raw}(Huber|C2) &= (\mu_{high\_turnover}(560) * \mu_{unattractive\_behaviour}(bad))^{0.5} * \\
 &\quad (1 - ((1 - \mu_{high\_turnover}(560)) * (1 - \mu_{unattractive\_behaviour}(bad))))^{0.5} \\
 M_{raw}(Huber|C2) &= (0.56 * 1)^{0.5} * (1 - ((1 - 0.56) * (1 - 1)))^{0.5} \\
 M_{raw}(Huber|C2) &= 0.75
 \end{aligned}$$

$$\begin{aligned}
 M_{raw}(Huber|C3) &= (\mu_{low\_turnover}(560) * \mu_{attractive\_behaviour}(bad))^{0.5} \\
 &\quad * (1 - ((1 - \mu_{low\_turnover}(560)) * (1 - \mu_{attractive\_behaviour}(bad))))^{0.5} \\
 M_{raw}(Huber|C3) &= (0.44 * 0)^{0.5} * (1 - ((1 - 0.44) * (1 - 0)))^{0.5} \\
 M_{raw}(Huber|C3) &= 0.66
 \end{aligned}$$

$$\begin{aligned}
 M_{raw}(Huber|C4) &= (\mu_{low\_turnover}(560) * \mu_{unattractive\_behaviour}(bad))^{0.5} \\
 &\quad * (1 - ((1 - \mu_{low\_turnover}(560)) * (1 - \mu_{unattractive\_behaviour}(bad))))^{0.5} \\
 M_{raw}(Huber|C4) &= (0.44 * 1)^{0.5} * (1 - ((1 - 0.44) * (1 - 1)))^{0.5} \\
 M_{raw}(Huber|C4) &= 0.66
 \end{aligned}$$

The raw membership degree of an element to a class is only meaningful if we consider the membership of elements to a single set. If, as is the

case in our example, the membership of elements to multiple classes is considered, the normalised membership degree must be calculated [28]. This is done by dividing the membership degree of an element to each set by the total of the membership degrees of this particular element to all sets. This gives a value between 0 and 1.

In our case the normalised membership degree of Huber to the classes C1 to C4 is:

$$\begin{aligned}
 M_{final}(Huber|C1) &= 0/2.07 \\
 &0 \\
 M_{final}(Huber|C2) &= 0.75/2.07 \\
 &0.36 \\
 M_{final}(Huber|C3) &= 0.66/2.07 \\
 &0.32 \\
 M_{final}(Huber|C4) &= 0.66/2.07 \\
 &0.32
 \end{aligned}$$

The normalised membership degree can be used to evaluate customer treatment. One example would be rewarding customers with a personalised discount [17]. To achieve this, each class of customers will be assigned a rebate, and every customer is given an individualised discount in accordance with their membership degree to each class. If, for instance, we decide to assign a 10% discount to class C1 and a 5% discount to classes C2 and C3, customer Huber would receive a personalised discount of:

$$Discount(Huber)=10\%*0+5\%*0.32+5\%*0.32=3.2\%$$

### 4.3 Hierarchical Fuzzy Classification

Until now only two dimensional fuzzy classification has been considered. Such a classification has the virtue of being intuitive but lacks information. In our example, for instance, it can immediately be seen that Huber has a very unattractive payment behaviour but it is impossible to determine

why. Under a multidimensional classification additional information can be added, but the result will be more difficult to interpret. A solution is to create a hierarchal classification. Such classifications have for instance been done to analyse customer behaviour [28] or credit rating [33]. In our

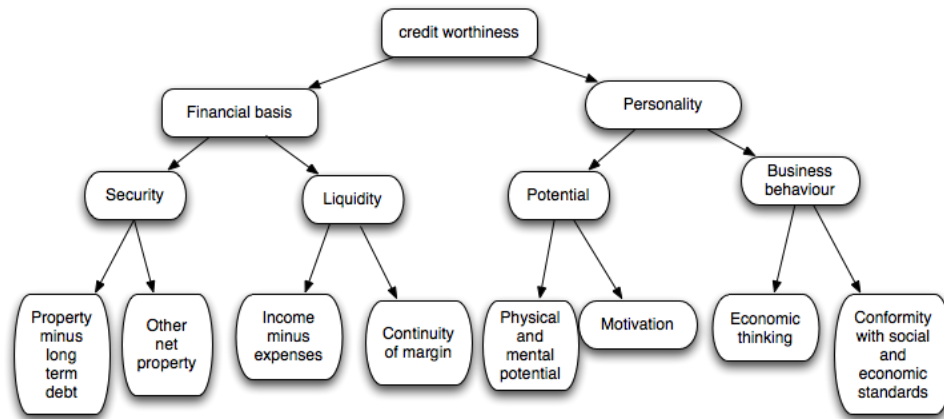


Figure 4.2: Hierarchy of credit worthiness taken from [33]

example Section 2.3.3 we determined a person's *credit worthiness* in terms of *personality* and *financial credit worthiness*. Zimmermann [33] defined *credit worthiness* using similar concepts and then broke these down further as shown in Figure 4.2.

- The concept of *credit worthiness* is broken down into the sub classifications *financial credit worthiness* and *personality*.
- The sub classification *financial credit worthiness* is then broken down into the concepts *security* and *liquidity*.
- The sub classification *personality* is then broken down into the concepts *potential* and *business behaviour*.
- The concept *security* is determined by the dimensions *property minus long term debts* and *other net property*.
- The concept *liquidity* is determined by the dimensions *income minus expenses* and *continuity of margin*.

- The concept *potential* is determined by the dimensions *physical and mental potential* and *motivation*.
- The concept of *business behaviour* is determined by the dimensions *economic thinking* and *conformity with social and economic standards*.

The hierarchical breakdown combines the simplicity and visual appeal of the two dimensional classification and the depth of information of the multidimensional classification. Analysis of a particular person's credit rating could for instance start at the top and then continue down through the different hierarchies while remaining easily understandable at all levels.

## 4.4 The Fuzzy Classification and Query Language

Once a fuzzy database schema has been defined, the next step is to query it. One way to do this is using the fuzzy Classification and Query Language (*fCQL*) and the accompanying toolkit, developed by the Information Systems research group at the University of Fribourg. *fCQL* allows users to perform fuzzy queries on fuzzy database schemas defined with linguistic variables [25][27][28].

### 4.4.1 fCQL syntax

*fCQL* is an extension of the Structured Query Language (SQL), the standard language for defining and query relational databases. The syntax, shown partially here is very similar to SQL.

<b>classify</b>	AttributeList
<b>from</b>	Relation
<b>where</b>	ClassificationCondition

The *classify* clause is similar in function and syntax to the SQL *select* clause. It is used to specify the list of attributes to be classified. The *from* is identical in name to its SQL equivalent and, just like in SQL, is used to specify the desired relation. The optional *where* clause is used, like the equivalent SQL *where* clause, to specify a predicate on which the classification is to be conditional.

#### 4.4.2 Examples of fCQL Queries

Having presented the fCQL, we will continue our previous example first shown in Section 4.1. Supposing we wish to query this database schema using fCQL, an initial query might be to display the names of all customers and their corresponding turnover and behaviour in the database:

```
classify customer, turnover, behaviour
from CustomerTable
```

This query is unconditional and the *where* clause is therefore omitted. The resulting table is the following:

ResultsTable		
customer	turnover	behaviour
Huber	560	bad
Mueller	500	good
Sieber	450	sufficient
Suter	900	excellent

A more advanced query might be to retrieve only those customers generating over 600 in revenue and having excellent behaviour. This query, expressed in fCQL would be as follows:

```
classify customer, turnover, behaviour
from CustomerTable
where turnover > 600
and behaviour is excellent
```

This results in the selection of only one customer as can be seen in the resulting table:

ResultsTable		
customer	turnover	behaviour
Suter	900	excellent

## 4.5 The fCQL toolkit

The previous section presented fCQL and a brief overview of the fCQL toolkit will now be given. A graphical overview of the architecture of the fCQL toolkit is shown in Figure 4.3. The whole toolkit was written as a standalone Java application. This means that the toolkit can run on most popular operating systems and can be used together with a wide variety of databases [28].

As mentioned in the previous section, fCQL is an extension of SQL. From an implementation point of view, this is achieved by adding meta-tables to the database catalogue. The meta-tables contain the definition of the membership functions and of the decomposition, if a hierarchical classification is used. This ensures that no data migration or transformation is required to perform fuzzy analysis on existing business data. This also means that the data can continue to be manipulated using SQL as happens in *case 1*.

At the heart of the fCQL toolkit is the fCQL interpreter. In *case 2*, users formulate fuzzy queries in fCQL. The fCQL interpreter then analyses these results and transforms them into SQL statements. To perform this translation, the interpreter accesses both the business data and the meta-tables stored in the database. The results are then returned to the user or to the application.

In addition to the interpreter, the fCQL toolkit also has a graphical user interface (GUI) (see *case 3*). This GUI provides wizards for creating fuzzy queries and defining new fuzzy classifications. It also allows query results to be displayed graphically to the user. A screenshot of the GUI is shown in Figure 4.4.

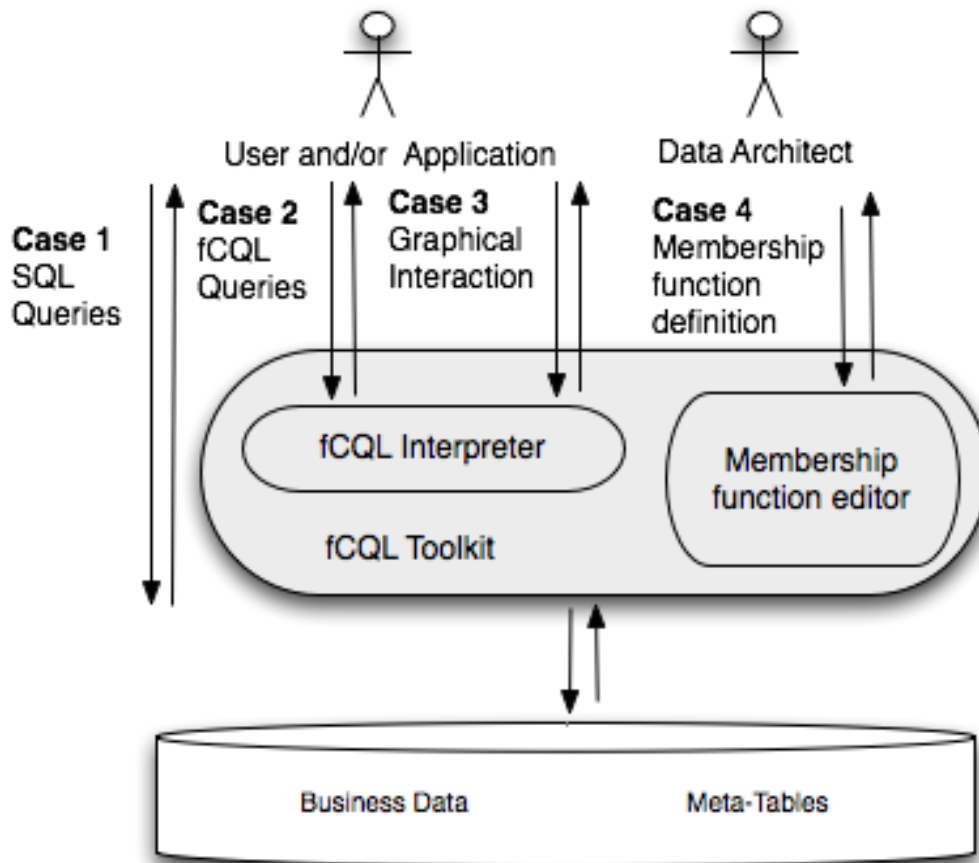


Figure 4.3: Architecture of the *fCQL* toolkit. Taken from [23]

One of the *fCQL* toolkit's key features is the membership function editor. It is a wizard that allows users to define and edit membership functions (*case 4*). It was developed to allow users to graphically create and manipulate both discrete and continuous membership functions without having to deal with complex mathematical functions [19].

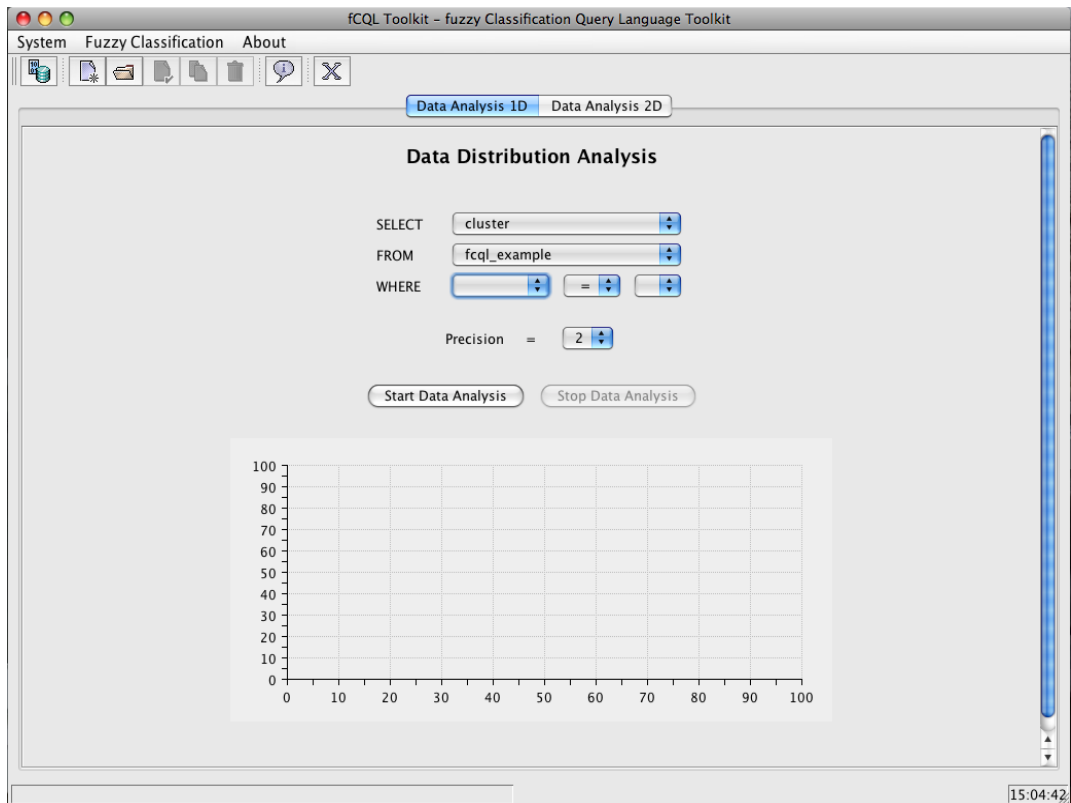


Figure 4.4: The *fCQL* toolkit graphical user interface

## 4.6 Summary of Findings

Modern databases are mostly based on the relational model proposed by Codd in 1970. This model is based on predicate logic and set theory. It also inherits their restrictions, namely queries return sharp results and all attributes must be atomic. To overcome these limitations, a context model was proposed by Chen and extended by the Information Systems Group at the University of Fribourg. In this model, each attribute is assigned a context.

A possible application for this is customer relationship management. Under a sharp customer classification, customers are assigned to a single set. Fuzzy classification on the other hand allows customers to have a gradual membership degree in each set, and customers can be assigned to multiple

sets. Membership to the different sets can be aggregated using the operators presented in Chapter 3. This means that an individual customer's value to the company can be precisely calculated.

To query such a fuzzy classification, the Fuzzy Classification and Query Language (FCQL) was developed. This language is an extension of the structured query language (SQL). The accompanying toolkit allows users to graphically edit membership functions, query the database and display results.



## Chapter 5

# Inductive Fuzzy Classification

This chapter describes the inductive fuzzy classification process proposed by Kaufmann [14]. First an overview of data mining and machine learning techniques is given. Next how these techniques can be used for inductive fuzzy classification is explained. The aim of such a classification is to induce the membership functions of elements in a target class. Finally a systematic approach for achieving such a classification is described.

## 5.1 Theory of Inductive Fuzzy Classification

### 5.1.1 Data Mining and Machine Learning

#### Purpose of Data Mining

The term data mining refers to a collection of techniques for analysing large sets of data. The purpose of such an analysis is to discover relations, to classify, to estimate and to predict [12].

- *Rule discovery* is the process of discovering what relations exist between data. An example of such a relationship might be that customers who buy a certain product are also likely to purchase another product.
- *Classification* is the process of grouping elements satisfying a certain predicate. The class  $C = \{e | P(e)\}$  is the class  $C$ , containing all the elements  $e$  satisfying the predicate  $P$ . An example of such a class might be customers willing to purchase a certain product.
- *Clustering* is similar to classification, but rather than determining membership of elements to a predefined class, classes are defined by grouping elements with similar attributes. To illustrate this, consider the difference between classifying and clustering students by grade. Whereas for classification a predicate such as *better than four* would have to be defined, whereas clustering would group students who have similar grades.
- *Prediction*: The techniques described above can not only be used to analyse historical data but to make predictions about the future. For instance once a relationship between elements in a customer's profile and products purchased has been discovered, this can be used to predict future purchasing behaviour.

## The Knowledge Discovery Process

While many sources treat data mining and knowledge as being synonymous, others view data mining as being a step in the knowledge discovery process [18]. Using

1. *Data cleaning* is the process of removing inconsistent and erroneous data
2. *Data integration* is the process of combining data from multiple sources
3. *Data selection* is the process of selecting data relevant to the upcoming analysis
4. *Data transformation* is the process of transforming data into a form suitable for analysis
5. *Data mining* is the process of applying intelligent methods to extract patterns from the data
6. *Pattern evaluation* is the process of evaluating the previously extracted patterns and retaining the most relevant ones.
7. *Knowledge presentation* is the process of visually presenting the results to the user.

Steps one to four together form the data preprocessing stage, where the data is prepared for the actual mining process. In step five the data is analysed using machine learning techniques, described next to identify interesting patterns that are then evaluated for correctness and presented to the user.

## Machine Learning

Machine learning is a subset of the field of artificial intelligence covering algorithms that allow a computer to *learn* [21]. In practice this means that given a set of data, these algorithms can be used to discover patterns and relations that can then be used to make predictions.

**Supervised versus unsupervised learning** Machine learning techniques can be grouped by the amount into two categories: supervised and unsupervised techniques.

Supervised learning techniques rely on example inputs and outputs, called a training set, to build a model that can then be used to analyse unlabeled data [21]. An example of a supervised learning technique is naive bayesian classification. Naive bayesian classification can for instance be applied to classify emails into spam and ham (not spam) [8]. Bayesian spam filters work as follows:

1. The training phase
  - (a) Two sets of messages, one labelled *spam*, the other *not spam* are created.
  - (b) The probabilities that each word will appear in a message, given that the message is spam or legitimate, is calculated.
2. The filtering phase
  - (a) The probability that a new message is spam, given that a certain term appears in this message is calculated.
  - (b) The probability that the message is spam is calculated, taking into consideration all of its terms. The algorithm is naive, meaning a strong independence between the probability of occurrence of the individual terms is assumed.

Unsupervised learning techniques do not require a labelled training set. An example of unsupervised learning is clustering, where similar elements are grouped based on a similarity measure [21]. One such clustering techniques is k-means clustering, as known as Lloyd's algorithm, after Stuart Lloyd who first proposed it [11]. Lloyd's algorithm operates as follows:

1. The input is partitioned into k initial sets, either randomly or using some heuristic.
2. The average point, or centroid, of each of the k-sets is calculated.

3. A new partition is created by associating each point with the closest centroid.
4. The centroid of each set is recalculated, and steps three and four are repeated until a convergence criteria is reached.

### **Fuzziness in Data Mining**

A number of advantages of using fuzzy methods in data mining have been identified. These include [10]:

- *Graduality*: Fuzzy sets can represent concepts with gradual, rather than crisp boundaries.
- *Interpretability*: Raw data, in numerical form, can be represented using linguistic variables, making it easier to interpret.
- *Robustness*: Because the boundaries between fuzzy sets are gradual, fuzzy methods are much less sensitive to *boundary effects*. When using crisp sets, a small shift of the boundary can lead to a dramatic change in classification if elements are located near the boundary.
- *Representation of uncertainty*: Machine learning is associated with uncertainty. This uncertainty is present in the data which can be imprecise, incoherent or incomplete. It is also present in the induced rules or classifications as more than one candidate theory could adequately explain an observation.

Fuzzy can be used both in supervised and unsupervised learning.

**Fuzzy methods in supervised leaning** Rule based fuzzy classifier can be adapted to used fuzzy rules [10]. These rules, induced form training sets are frequently of the type IF A THEN B. For example if a customer spends more than  $x$ , he is given  $y$  discount. Rather than using hard rules, soft rules can be used. This avoids threshold effects where a small variation in turnover  $x$  can lead to large difference in discout  $y$ .

**Fuzzy methods in unsupervised learning** In fuzzy cluster analysis, such as fuzzy k-means clustering [5], elements are grouped in fuzzy, as opposed to crisp, sets. This means that elements have different degrees of membership to a set. Elements can also be members of multiple sets.

## 5.1.2 Inductive Fuzzy Classification

### Inductive Classification

Inductive classification is a form of supervised learning in which the membership of an  $e$  element to a set  $y$  is induced. The classification is based on the attributes  $e$ .

As stated previously, supervised learning techniques require a training set. This training set  $d$  is an  $(n + 1) \times m$  matrix with  $n$  columns  $X_1, \dots, X_n$ , and a column  $Y$  indicating membership to the target class  $y$ . The columns  $X$ ,  $i \in 1, \dots, n$  are called dependent variables or attributes, and  $Y$  is called the target variable or class label. The rows of the matrix represent the elements  $e_j$ ,  $j \in 1, \dots, m$ .

$$\begin{pmatrix} x_{1j} & \dots & \dots & \dots & y_j \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & y_j \end{pmatrix}$$

The training set in matrix form

The matrix contains the attribute values  $x_{ij}$  representing the value of the  $i^{th}$  attribute for the  $j^{th}$  element, and the labels  $y$  representing the class membership for the  $j^{th}$  element. In case of a sharp classification, label  $y_j$  is equal to 1 if and only if the element  $e_j$  is in class  $y$ .

The training process induces a model  $M$  from the training set  $d$  based on the mapping of the dependant variables on the target variable. This model can

then be used to predict the class membership of data elements  $e'$  in a new dataset  $d'$  with unknown class memberships. For every element  $e' \in d'$ , a predicted membership  $Y'$  can be calculated by applying the model  $M$  to the elements attributes ( $Y' = M(x_{i1} \dots x_{in})$ ). Once  $Y'$  has been calculated, all the equivalence class  $y' = \{e_j | Y'_j = 1\}$  can be generated. This is an inductive process because the classification is based on a predicate induced from the learning set.

### **Inductive Fuzzy Classification**

Inductive fuzzy classification is an inductive classification where the target class is a fuzzy set. The difference to the afore-mentioned sharp classification is that the target variable  $Y$  is not a binary variable but a membership function. The induction performed therefore consists of inducing the membership functions of elements to a target set. This is done using the inductive fuzzy classification process outlined in the next section.

## **5.2 The Inductive Fuzzy Classification Process**

The following section outlines the fuzzy classification process (*IFCP*) proposed by Kaufmann [14]. This is a supervised learning process which aims to accurately predict the membership of data elements to a fuzzy class. For this, a fuzzy target class  $y'$  is created.  $y'$  is defined in such a way that the greater membership of an element  $e$  in  $y'$ , the greater the likelihood of  $e$  being in the crisp target set  $y$ . An example showing this distinction, would be the analysis of customer data to determine a customer's penchant for certain products. In this case membership to the crisp target set would indicate, in a binary way, if a customer has or will purchase this product, (given in the training set) and the membership in the fuzzy set would how likely a given customer is to purchase a product.

This multi-staged process is outline in Figure 5.1. It consists of the following steps:

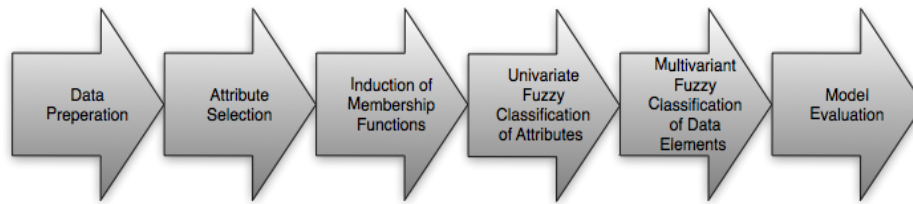


Figure 5.1: the inductive fuzzy classification process

1. *Data preparation*: Creation of the training set and definition of the target variables.
2. *Attribute selection*: Ranking of the elements in the training set according to their relevancy.
3. *Induction of membership functions*: Induction of the membership value of individual attributes to the target class.
4. *Univariate fuzzy classification of attribute values*: Transformation of the individual attributes into membership function.
5. *Multivariate fuzzy classification of data elements*: Aggregation of the individual membership functions.
6. *Model evaluation*: Evaluation of the classifier's predictive performance.

### 5.2.1 Data Preparation

The first step of the process is to create a training set. This is done by combining data from various sources into a single table that holds the various elements and for each element, a label indicating membership of the target set. This label is restricted to the values  $\{0, 1\}$ . A value of 1 indicates that this element is a member of the target set, and a value of 0 indicates that it is not.

## 5.2.2 Attribute Selection

Not all elements in the training set are equally relevant for predicting membership to the target set. When considering the likelihood of a person investing in a pension fund, a persons revenue and age are a much better indicator than their preferred language of correspondance. The relevancy of an element can be determined using for example the mutual information value.

**Mutual Information** The mutual information, of two random variables measures their mutual dependence [6].

Formally the mutual information of two discrete random variables  $X$  and  $Y$  is defined as [6]

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log\left(\frac{p(x, y)}{p_1(x)p_2(x)}\right) \quad (5.1)$$

Here  $p(x, y)$  is the joint probability distribution function of  $X$  and  $Y$ , and  $p_1(x)$  and  $p_2(y)$  are the marginal probability distribution functions of  $X$  and  $Y$  respectively.[6]

Intuitively, mutual information measures the information that  $X$  and  $Y$  share. For example, if  $X$  and  $Y$  are independent ( $p(x, y) = 0$ ), then knowing  $X$  does not give any information about  $Y$  and vice versa, so their mutual information is zero.

Once the relevancy of all elements has been determined, only the best  $n$  variables are retained.

## 5.2.3 Induction of Membership Functions

The most relevant elements selected in the previous step are assigned a membership degree in the fuzzy set  $y'$ . This is done by defining a fuzzy

restriction  $y_i$  for each element  $X_i$  on its domain. This fuzzy restriction represents the likelihood  $Y=1$ .

**Likelihood Function** The likelihood of an event A given B is equal to the conditional probability of B given A defined by Bayes theorem [14].

$$L(A|B) = p(B|A) = \frac{p(A|B)p(B)}{p(A)} \quad (5.2)$$

The likelihood is then transformed into a membership function as follows:

$$\mu_{y_i}(x) = \frac{p(X_i = x|Y = 1)}{p(X_i = x|Y = 1) + p(X_i = x|Y = 0)} \quad (5.3)$$

For each value of  $x \in \text{dom}(X_i)$ , the fuzzy restriction  $y_i$  is defined as the conditional probability of  $x$  given  $Y=1$ , divided by the sum of the probability of  $x$  give  $Y=1$  plus the conditional probability of  $x$  given  $Y=0$ . This corresponds to the normalised likelihood ratio of (NLR) of  $x$  given  $Y=1$  [14].

Next, these the membership functions for each variable must be calculated. For categorical variables, this is straight forward: the membership degree for each value is defined by the corresponding NLR. However, for continuous variables, the membership function could likewise be be continuous. In that case, a solution is to calculate the NLR or quantiles of  $\text{dom}(X_i)$ , generating a piecewise linear function which approximates a continuous function.

To illustrate how membership functions are induced, consider the following example. Suppose four customer groups A, B, C, and D and a target class  $y'$  whose customers purchase a given product  $a'$ . For those customers who have not yet purchased a product, the membership in  $y'$  will be induced. The training set is as follows:

Group	$Y = 1$	$Y = 0$	Total
A	11455	308406	319861
B	12666	54173	66839
C	5432	10843	16275
D	249	2917	3166
Total	29802	376339	406141

The column  $Y=1$  contains the number of customers in each group who purchased the product and column  $Y=0$  the number of customers who did not. The probability of a customer, given his group, purchasing a product can now be calculated using Bayes' formula [20]

$$p(A|B) = \frac{p(A \cap B)}{p(B)} \quad (5.4)$$

This gives the following results:

Group	$Y = 1$	$Y = 0$
A	0.384370176	0.819489875
B	0.425005033	0.143947345
C	0.182269646	0.028811789
D	0.008355144	0.00775099
Total	1	1

The membership degree of each customer group to the class  $y'$  can now be calculated. For group A, this is

$$\begin{aligned} \mu_{y'}(A) &= \frac{p(A|Y=1)}{p(A|Y=1)+p(A|Y=0)} \\ \Rightarrow \mu_{y'}(A) &= \frac{0.384370176}{0.384370176+0.819489875} \\ \Rightarrow \mu_{y'}(A) &= 0.319281445 \end{aligned}$$

The other membership degrees are:  $\mu_{y'}(B) = 0.746995793$   $\mu_{y'}(C) = 0.863503916$   
 $\mu_{y'}(D) = 0.518755384$

## 5.2.4 Univariate Fuzzy Classification

The next step consists of fuzzifying the different attributes. This is done by replacing each variable  $X_i$  by its corresponding membership function induced in the previous step. In our previous example, the following substitutions would be made:

$A \rightarrow 0.319281445$   
 $B \rightarrow 0.746995793$   
 $C \rightarrow 0.863503916$   
 $D \rightarrow 0.518755384$

## 5.2.5 Multivariant Fuzzy Classification

After the membership degrees of the individual elements to the target class  $y'$  have been induced, the membership degrees of the different elements  $e_j$  can be calculated. The membership of an element  $e_j$  to  $y'$  is an aggregation of the membership degrees of all the attributes  $x_{ij}$  of the element  $e_j$ .

A variety of fuzzy set aggregators were shown in Section 2.3. The most suitable of these is the gamma operator because it combines the algebraic sum and the algebraic product with a parameter  $\gamma \in [0, 1]$  which allows for varying degrees of compensation.

The result of this aggregation is a membership function for an element to the target class which can be used on unlabeled data for prediction.

### Model Evaluation

The final step in the inductive fuzzy classification process is evaluating the model's predictive performance. This is done by classifying a data set from which target class labels have been removed. By comparing the predicted membership values to the actual one, the classifier's performance can be quantified. A possible performance metric is the mutual information between the prediction  $Y'$  and the class label  $Y$ .

## 5.3 Summary of Findings

The inductive fuzzy classification process, proposed by Kaufmann, uses machine learning techniques and data mining create fuzzy classifications. It is a multistage process consisting of:

1. *Data preparation*: Creation of the training set and definition of the target variables.
2. *Attribute selection*: Ranking of the elements in the training set according to their relevancy.
3. *Induction of membership functions*: Induction of the membership value of individual attributes to the target class.
4. *Univariate fuzzy classification of attribute values*: Transformation of the individual attributes into membership function.
5. *Multivariate fuzzy classification of data elements*: Aggregation of the individual membership functions.
6. *Model evaluation*: Evaluation of the classifier's predictive performance.

The result is a model that can be used to predict membership degrees for different elements to a fuzzy target set.



# Chapter 6

## Inductive Fuzzy Classification

### Language

The following chapter describes the fuzzy classification language as well as an interpreter for this language that was developed in the context of this master thesis. First, the iFCQL syntax is given, then an overview of the interpreter is presented. The presentation of the interpreter consists of an architectural overview followed by a detailed description of how each iFCQL is translated into SQL and executed on a database.

#### 6.1 Motivation and Objectives for iFCQL

The objective of the iFCQL and the accompanying toolkit is to support the inductive fuzzy classification in all of its stages. To summarise the previous chapter, these stages are:

1. *Data preparation*: Creation of the training set and definition of the target variables.
2. *Attribute selection*: Ranking of the elements in the training set according to their relevancy.
3. *Induction of membership functions*: Induction of the membership value of individual attributes to the target class.
4. *Univariate fuzzy classification of attribute values*: Transformation of the individual attributes into membership function.
5. *Multivariate fuzzy classification of data elements*: Aggregation of the individual membership functions.
6. *Model evaluation*: Evaluation of the classifier's predictive performance.

## 6.2 iFCQL Syntax

iFCQL was designed to be an extension of FCQL. It provides statements that support the inductive fuzzy classification process in all of its stages. The following sections show the syntax of the different statements grouped by stage in the IFC process. Please note that the first stage, data preparation, is supported by SQL and iFCQL therefore does not provide a syntax to support it.

## 6.2.1 Attribute selection

### Audit

*< data – audit – statement >:= audit < dependent – variable >  
from < relation >  
targeting < target – variable >  
[partition by ntile < Number of tiles >]  
[compare to < univariate fuzzy class >]*

*< dependent – variable >:= < SQL column expression >  
< target – variable >:= < SQL column expression >  
< relation >:= < SQL table > | < SQL view >*

*< SQL column >:= String  
< SQL table >:= String  
< SQL view >:= String*

*< univariate fuzzy class >:= String*

### Attribute Ranking

*< attribute – ranking – statement >:= rank attributes  
from < relation >  
targeting < target – variable >  
[partition by ntile < Numberoftiles >]  
[into < tablename >]*

## 6.2.2 Membership Function Induction

*< membership – induction – statement >:= induce fuzzy membership  
of < dependent – variable > in < target – variable >  
from < relation >  
[partition by ntile < Number of tiles >]*

### 6.2.3 Univariate Fuzzy Classification

*Classify column* < column > as < univariate – fuzzy – class >  
With < SQL – expression > | (< membership – induction – statement >)

< SQL – expression >:= String

### 6.2.4 Multivariate Fuzzy Classification

*Classify table* < relation >  
As < multivariate – fuzzy – class >  
Select < columns >  
With < gamma – aggregation >  
[Into < new – table >]

< multivariate – fuzzy – class >:= String

< columns >:= < SQLcolumn > [, < SQLcolumn >]\*

< gamma – aggregation >:= gamma < gamma\_value >  
(< column > is < univariate – fuzzy – class >  
[, < column > is < univariate – fuzzy – class >]\*)

< new – table >:= String

### 6.2.5 Model evaluation

*Evaluate prediction* < prediction – variable > with < target – variable >  
From < relation >  
Where < SQL – conditions >

< prediction – variable >:= < SQL column >

## 6.3 The iFCQL Interpreter

The iFCQL interpreter currently supports the IFP in three of its core stages (shaded in Figure 6.1). These stages are:

- 3 *Induction of membership functions*
- 4 *Univariate fuzzy classification of attribute values*
- 5 *Multivariate fuzzy classification of data elements*

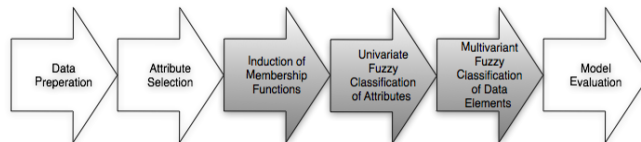


Figure 6.1: Parts of the IFP that have been implemented

### 6.3.1 Design Requirements

The architecture of the iFCQL, shown in Figure 6.2 has the following components:

- *Script based meta data*: The meta data that is required by the classification process, such as scripts and fuzzy classification definitions, is stored in external text files. This has the following advantages:
  - Once a fuzzy classification has been induced using the interactive client, it can be stored in a text file for easy reuse,
  - The data stored in the database does not need to be modified,
  - The user of the iFCQL client does not require permission to write to the database.

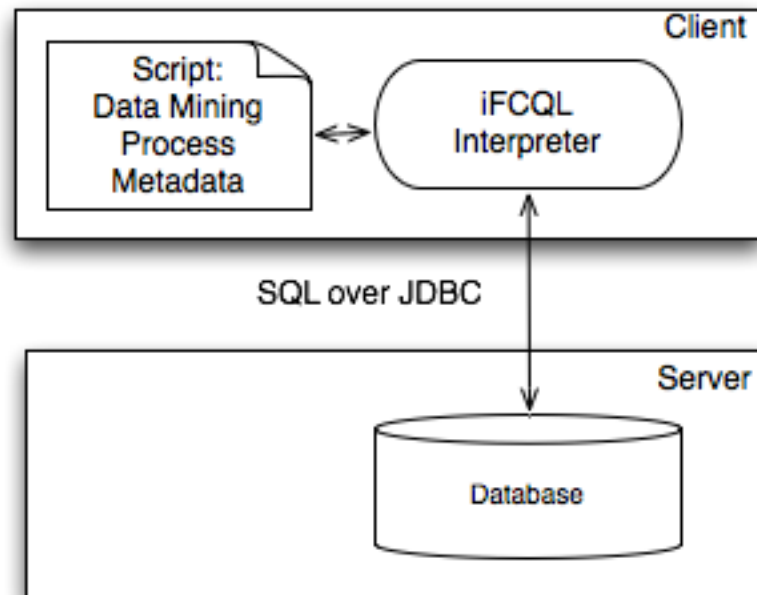


Figure 6.2: Overview of the fcql Client Server Architecture

- *Client Architecture:* The client is a self contained Java application and no server installation is required. This has the following advantage:
  - The iFCQL toolkit can be installed directly on the client computer and can then work with an existing database.
- *Generic JDBC interface:* Connections to the database are made using a standard connection. This has the following advantage:
  - The iFCQL toolkit is can work with a variety of databases without requiring any modification.

### 6.3.2 Client Architecture

A graphical overview of the architecture of the iFCQL client is given in Figure 6.3. The client is composed of largely independent modules which

can be decoupled. This allows the lexer, parser and the evaluator to be inserted into an existing data-mining tool.

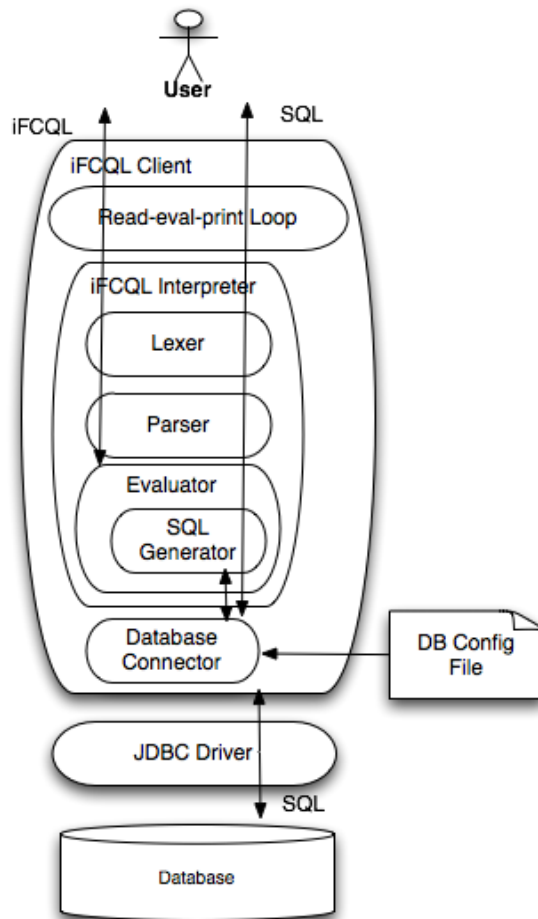


Figure 6.3: The *ifCQL* Client Architecture

- The *read-eval-print loop* accepts input in the form of **SQL** and *iFCQL* statements. **SQL** statements are based directly onto to the database connector and *iFCQL* statements are passed to the lexer. The query results are then displayed in the Console.
- The *lexer*, also called a tokeniser, separates the input into individual tokens and removes whitespace characters.

- The *parser* recognises the individual *iFCQL* commands and stores their parameters in a temporary structure before calling the corresponding function in the evaluator.
- The *code evaluator* and *SQL generator* then create the SQL code that corresponds to the *iFCQL* command.
- The *database connector* reads the database configuration files and establishes a connection to the database that is used to execute the generated SQL commands.
- The *JDBC driver* is an external jar file which enables the java client application to connect to the database.

## 6.4 Client Implementation

The *iFCQL* grammar was introduced in Section 6.2, and the architecture of an interpreter for this language was described in Section 6.3. Next a description of how the *iFCQL* interpreter translates the *iFCQL* statements into SQL expressions will be given. The *iFCQL* commands are grouped according the stage of the *FCQL* process that they support. As a reminder, these stages are:

1. the data preparation stage,
2. the membership induction stage,
3. the univariate classification stage,
4. the multivariate classification stage.

Throughout this chapter, an example will be used to illustrate how the SQL commands are generated. This example is a case study, conducted by Kaufmann for a personalised marketing campaign for a financial service provider [14]. The inductive fuzzy classification process was used to determine an individuals customer's affinity for a certain investment fund.

### 6.4.1 Data Preparation

The iFCQL client console allows users to enter SQL commands directly. These can be used to create the training sets. The query results are displayed directly in the console.

In our case study, the criteria retained where:

- Customer segment
- Number of products already purchased
- overall account balance
- Customer loyalty
- Customer group
- Private account balance
- Age

### 6.4.2 Membership function induction

This function calculates the membership function for an individual element to a specified set and generates the appropriate SQL code.

#### Input

This function takes the following input:

- *< dependent – variable >*
- *< target – variable >*

- *< table >*
- [*n\_tile < integer >*] (optional)

## Output

The output of this function is an SQL case statement which assigns a membership degree *< classification – column >* of *< dependent – variable >* in *< target – variable >* which varies according to the value of *< dependent – variable >*. More specifically:

- If the value of *< dependent – variable >* is missing, this element is assigned a *< classification – column >* value of 0.5.
- For discrete variables, a membership degree is induced for each value of *< dependent – variable >* using the normalised likelihood ratio.
- For continuous variables, a piecewise linear membership function is generated. This function is a linear interpolation of the NLR of each quintile.
- The default value for the number of quantiles is 10.

## The Resulting SQL code

When generating the SQL code, we have to distinguish two cases, generating the code for discrete variables and generating the code for continuous variables.

### Discrete Variables

In this case the input would be in the following format:

```
Induce fuzzy membership of <dependent-variable>
in <target-variable> from <table>
```

And the resulting SQL code is:

```
Select x, (n_x1/n__1) / ((n_x1/n__1)+(n_x0/n__0)) as nlr
from
  (Select <dependent-variable> as x, sum(<target-variable>) as n_x1,
  sum(1-<target-variable>) as n_x0
  from <table> group by <dependent-variable>) a
  cross join
  (select sum(<target-variable>) as n__1, sum(1-<target-variable>)
  as n__0 from <table>) b
```

In our case study, the customer segment, *KS*, is an example of a discrete variable. Customers are classified into groups according to their status: *Privatkunden Basis*, *Privatkunden Gold*, *Privatkunden KU* and *Privatkunden Premium*. The likelihood that a customer has invested in a particular fund, indicated by the target variable *hat\_fonds* based on their status, is calculated as follows, assuming all data is stored in the table *ads\_fonds\_pk5*:

The corresponding iFCQL command is:

```
Induce fuzzy membership of ks in hat_fonds from ads_fonds_pk5
```

The SQL code generator will then generate the following code:

```
Select x, (n_x1/n__1) / ((n_x1/n__1)+(n_x0/n__0)) as nlr
from
  (Select hat_fonds as x, sum(ks) as n_x1,
  sum(1-ks) as n_x0
  from ads_fonds_pk5 group by <dependent-variable>) a
  cross join
  (select sum(hat_fonds) as n__1, sum(1-hat_fonds)
  as n__0 from ads_fonds_pk5) b
```

When this code is evaluated, it gives the following result:

```

case
when KS is null then 0.5
when KS = 'Privatkunden Basis' then 0.319281444780263
when KS = 'Privatkunden Gold' then 0.746995793024603
when KS = 'Privatkunden KU' then 0.518755384475754
when KS = 'Privatkunden Premium' then 0.863503916037109
end

```

## Continuous Variables

In our case study, the customer age, *age*, is an example of a continuous variable, defined between zero and 100.

The likelihood that a customer has invested in a particular fund, indicated by the target variable *hat\_fonds* based on their age, discretised over 4 quantiles, is calculated as follows, assuming all data is stored in the table *ads\_fonds\_pk5*:

In this case the input would be as follows:

```

Induce fuzzy membership of age
in hat_fonds from ads_fonds_pk5
partitioned by 4

```

Induction of the membership function is done using the following steps:

1. Calculating the quantiles. This is done by:
  - (a) Sorting the *< table >* by *< dependent – variable >*
  - (b) Numbering the lines in *< table >*
  - (c) Calculating the quantiles.
2. Calculating the normalised likelihood ratio for each quantile
3. Calculating the piecewise linear function.

When this code is evaluated, it gives the following result:

```
case
when KS is null then 0.5
when KS = 'Privatkunden Basis' then 0.319281444780263
when KS = 'Privatkunden Gold' then 0.746995793024603
when KS = 'Privatkunden KU' then 0.518755384475754
when KS = 'Privatkunden Premium' then 0.863503916037109
end
```

### 6.4.3 Univariate Classification

The purpose of the univariate fuzzy classification is to define a fuzzy set using an SQL statement. The exact syntax of this command is:

*Classify column < column > as < univariate – fuzzy – class >  
With < SQL – expression > | (< membership – induction – statement >)  
< SQL – expression > := String*

#### Input

The classify column takes the following input:

1. *< column >* The column to be classified.
2. *< univariate – fuzzy – class >* The name of the fuzzy class that will be generated.
3. Either:

- (a) *< SQL – expression >* An SQL statement defining the membership degrees of the elements in *< column >* in the new set *< univariate – fuzzy – class >*. This SQL statement is generated in the previous step of the IFC process.
- (b) Or (*< membership – induction – statement >*) An iFCQL statement that will generate the membership degrees of elements in *< column >* in the new set *< univariate – fuzzy – class >*

#### 6.4.4 Multivariate Classification

The multivariate classification takes two or more univariate fuzzy classifications and aggregates them into a single fuzzy class using the gamma operator. The exact syntax of this function is as follows:

```
Classify table < relation >
As < multivariatefuzzyclass >
Select < columns >
With < gammaaggregation >
[Into < newtable >]
```

*< multivariatefuzzyclass >:= String*

*< columns >:= < SQLcolumn > [, < SQLcolumn >]\**

*< gammaaggregation >:= gamma < gamma\_value >  
( < column > is < univariatefuzzyclass >  
[, < column > is < univariate – fuzzy – class >]\* )*

*< new – table >:= String*

#### Input

The multivariate classification takes the following input:

1. *< multivariatefuzzyclass >* The table in the database containing the data that is to be classified.

2. *< columns >* The columns of *< multivariatefuzzyclass >* that are to be displayed
3. *< gamma-aggregation >* The gamma aggregation statement containing:
  - (a) *< gamma-aggregation >* The value of the gamma parameter.
  - (b) A list of *< column >* is *< univariate – fuzzy – class >*. Where *< column >* is a column stored in the vector *< univariatefuzzyclass >*.
4. Optionally [Into *< new-table >*] The result is inserted into a table named *< newtable >*

## Output

The result of a multivariate classification is either an SQL script that will generate the classification, or a table in the database containing the classification.

## 6.5 Summary of Finding

To support the Inductive Fuzzy Classification Process described in Chapter 4, the Inductive Fuzzy Classification Language (iFCQL) and an accompanying interpreter were developed. iFCQL supports the Inductive Fuzzy Classification in three of its six steps. The steps supported are:

1. the data preparation stage,
2. the membership induction stage,
3. the univariate classification stage,
4. the multivariate classification stage.

Two key features of the interpreter are firstly that it is a stand-alone Java application and secondly that it is database independent. Being a self contained application means that the database means that the raw data being classified does not have to be modified. Database independence is achieved by using a generic JDBC interface to connect to the database. This means that the data does not need to be migrated to a new database system before being analysed.

# Chapter 7

## Conclusion

### 7.1 Summary of Findings

The purpose of this Master thesis was to create an interpreter to support the inductive fuzzy classification process. The inductive fuzzy classification process creates a fuzzy classification, the theoretical basis for which is in fuzzy logic and fuzzy set theory.

#### 7.1.1 Fuzzy Sets

Fuzzy set theory gives us a mathematical tool for modelling classes of objects without clear cutoffs. Unlike sharp set theory where membership to sets is binary, membership of elements to fuzzy sets is defined by a membership function. Membership degrees can have a value between zero and one. This means that an element can be a member of different sets at the same time and can that some elements can be member to a greater degree than others of a set. This gives us a more accurate model for customer classes. Fuzzy Logic

Just like fuzzy sets are an extension of crisp sets, fuzzy logic is an extension of binary logic. This allows for fuzzy propositions and linguistic variables.

Fuzzy propositions have a truth value between 0 and 1. This allows the modeling of uncertainty, even when probabilities are unknown. Linguistic variables present a more ergonomic way of manipulating data.

### **7.1.2 Fuzzy Classification**

Fuzzy set theory was applied to extend the relational model to allow fuzzy classification in databases. Such a classification can for instance be used for customer relationship management. Under a sharp customer classification, customers are assigned to a single set. Fuzzy classification on the other hand allows customers to have a gradual membership degree in each set, and customers can be assigned to multiple sets. This allows a company to determine the precise value of each customer.

### **7.1.3 Inductive Fuzzy Classification Process**

The objective of the Inductive Fuzzy Classification process is to create a fuzzy classification. First a training set is created. Then data mining and machine learning techniques are used to determine the membership of different elements to a target set. These membership functions are then used to create a univariate and possibly a multivariate classification. This can then be used to predict the membership of new objects to the target set.

### **7.1.4 Inductive Fuzzy Classification Language**

The Inductive Fuzzy Classification Language was developed to support the Inductive Fuzzy Classification Process. This language is an extension of SQL. The accompanying interpreter was developed to allow users to easily create and apply each of the steps in the process. Importantly, this is a standalone application that is database independent. This means that the user does not have to modify the underlying data.

# Appendix A

## iFCQL Interpreter Instruction Manual

### A.1 System requirements

- Java 1.6 or later
- Postgres SQL 8.3

### A.2 Installation Instruction

1. Unpack the file *iFCQLtoolkit.zip*
2. Microsoft Windows users can run the file *start.bat*

### A.3 Running the iFCQL toolkit

After starting the toolkit, the first thing you must do is configure access to the database. This can be done by clicking on the options menu. It is important to specify url to the database as well as the login name and password that

you will be using. After this, you can enter iFCQL commands in the left input pane. Hitting the eval button will start the interpreter. The results will be shown in the right output pane.

# Bibliography

- [1] G. Bojadziev /& M. Bojadiev. *Fuzzy Logic for Business, Finance and Management*. World Scientific, 2nd edition, 2007.
- [2] R. Mitchell & R. T. Carson. *Using Surveys to Value Public Goods*. Resources for the Future, 1989.
- [3] G. Chen. *Fuzzy Logic in Data Modeling - Semantics, Constraints and Database Design*. Kluwer Academic Publishers, 1998.
- [4] E.F. Codd. A relational model for large shared data banks. *Communications of the ACM*, 13(6):377–387, June 1970.
- [5] J.J. deGrujter & A.B. McBratney. A modified fuzzy k means for predictive classification. *Classification and Related Methods of Data Analysis*, pages 97–104, 1988.
- [6] R. Togneri & J.S. deSilva. *Fundamental of Information Theory and Coding Design*. Chapman & Hall, 2002.
- [7] C. Cornelis et al. Preference uncertainty in non-market valuation: A fuzzy logic approach. *American Journal of Agricultural Economics*, 3(83):487–500, Aug 2001.
- [8] P. Graham. A plan for spam. In Paul Graham, editor, *Hackers & painters: big ideas from the computer age*. O'Reilly, 2004.
- [9] P. A. Diamond & J. A. Hausman. Contingent valuation: Is some number better than no number? *The Journal of Economic Perspectives*, 8(4):45–64, 1994.
- [10] E. Hüllermeier. Fuzzy methods in machine learning and data mining: Status and prospects. *Fuzzy Sets and Systems*, 156(3), 2005.

- [11] S. P. Lloyd. Last square quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [12] D. G. Luenberger. *Information Science*. Princeton University Press, 2006.
- [13] A. Meier. *Relationale und Postrelationale Datenbanken*. Springer-Verlag, 6th edition, 2007.
- [14] M. Kaufmann & A. Meier. An inductive fuzzy classification applied to individual marketing. Technical report, University of Fribourg (Switzerland), 2008.
- [15] N. Werro & H. Stormer & A. Meier. A hierarchical classification of online customers. In *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE)*, pages 256–263, Shanghai, China, oct 2006.
- [16] N. Werro & H. Stormer & A. Meier. A hierarchical fuzzy classification of online customers. In *IEEE International conference on e-Business Engineering*, pages 256–263, 2006.
- [17] N. Werro3 & H. Stormer & A. Meier. Personalized discount - a fuzzy logic approach. In *Proc. of the 5th IFIP International Conference on eBusiness, eCommerce and*, Poznan, Poland, 2005.
- [18] J. Han & M.Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2 edition, 2006.
- [19] Christian Nancoz. medit - membership function editor for fcql-based architecture. Master’s thesis, University of Fribourg, Switzerland, 2005.
- [20] J.A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, 2 edition, 1995.
- [21] T. Segaran. *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O’Reilly, 2007.
- [22] S. Sheno. Fuzzy sets, information clouding and database security. In P. Bosc & J. Kacprzyk, editor, *Fuziness in Database Management Systems*, volume 5 of *Studies in Fuzziness*, pages 207–228. Physica Publisher, 1995.

- [23] A. Meier & H. Stormer. Using a fuzzy classification query language for customer relationship management. In *Proceedings of the 31 VLDB Conference*, pages 1089–1096, loyalty programs have been derived from fuzzy customer Trondheim, Norway, 2005.
- [24] G. C. van Kooten & Emina Krcmar & E. H. Bulte. Preference uncertainty in non-market valuation: A fuzzy approach. *American Journal of Agricultural Economics*, 83(3):487–500, Aug 2001.
- [25] A. Meier & C. Savary & G. Schindler & Y. Veryha. Database schema with fuzzy classification and classification query language. In *Proc. of the International Congress on Computational Intelligence – Methods and Applications*, University of Wales, Bangor U.K., June 19-22, 2001. CIMA 2001.
- [26] H.T. Nguyen & E.A. Walker. *A First Course in Fuzzy Logic*. Chapman & Hall, 2006.
- [27] A. Meier & G. Schindler & N. Werro. Zur unscharfen classification von datenbanken mit fcql. In *Proceedings of the GI - Workshop LLWA - Lehren Wissen Adaptivität*, pages 151–158, Karlsruhe, Germany, October 2003.
- [28] N. Werro. *Fuzzy Classification of Online Customers*. PhD thesis, University of Fribourg, 2008.
- [29] L. Zadeh. Fuzzy logic and approximate reasoning. *Synthese*, 30(3-4):407–428, 1975.
- [30] L. Zadeh. Discussion: Probability theory and fuzzy logic are complementary rather than competitive. *Technometrics*, 37(3):271–276, 1995.
- [31] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [32] J. Zahnd. *Logique Elementaire*. Presses polytechniques et universitaires romande, 1998.
- [33] H. J. Zimmermann. *Fuzzy Set Theory and its Applications*. Kluwer Academic Publishers, 2 edition, 1993.