A Decision Aiding System based on a Decision Rule Apporach and Fuzzy Logic

Jarle Skjørestad Heldstab

June 15th, 2010

Contents

List of Figures

List of Tables

Abstract

The first problem discussed in this thesis studied the theory of using a software system for decision aiding in decision problems with multiple criteria. Two such problem types were considered, namely classification and sorting problems. A second problem in this thesis studied a method to allow a human preference model as a basis for a real-world decision aiding problem. A problem number three looked at the possibilites of improving decision making in the oil and gas industry.

Three experiments was performed to test the three problems in this thesis. For the first problem, an implementation of a decision aiding system utilizing a decision rule approach, namely classical rough set theory and dominance rough set approach, was done. The system was able to make decisions in classification and sorting problems. For the second problem, fuzzy logic was combined into the implementation. The results showed that combining a decision rule approach with fuzzy logic made it possible to use a human preference model as a basis for real-world decision aiding problems. For the third problems, it was found that in a real-world decision making problem from the oil and gas industry, utilization of an autonomous decision aiding system can improve the quality of the results.

Lastly, the implemented decision aiding system was compared to an artificial neural network, and the results showed that the system had some advantages over the neural network.

Keywords: Decision aiding, classical rough set theory, dominance rough set approach, fuzzy logic, decision rules, decision maker.

Acknowledgements

First of all, I would like to thank my supervisors during this process, Associate Professor Hein Meling at the University of Stavanger, and Nejm Sadallah at the International Research Institute of Stavanger.

I would also like to thank Ingrid Tjøstheim Eek, my dear girlfriend, and our one year old son, Sebastian. Thanks for putting up with me in this period. Without you, this thesis would not have been the same.

1 Introduction

In this study, two decision rule approaches, namely the classical rough set theory and dominance rough set approach, are utilized to build a decision aiding system. The system has the ability to make decisions in classification and sorting problems. The study also proposes to combine the decision rule approach techniques with fuzzy logic, and thus use the decision rule approach in the process of constructing a fuzzy logic controller, while taking advantage of the decision making capabilities of the fuzzy controller to further extend the system's area of application.

The rest of this chapter gives a brief overview of the work carried out in this thesis. Firstly, a short statement of the issue of the work is given. Then, the problem statement is pinpointed. Lastly, a short outline of the contents of the thesis is given.

1.1 The issue

The nature of the decision problems that humans in general face are of multiple criteria [1]. According to the autors of [2], when people make decisions they search for rules which provide good justification of their choices. The process of making such decisions can vary greatly in accordance to several factors. One such factor includes human preferences, or for example in the oil and gas industry, the quality of decision making may vary persuant to the experience of the drilling staff. The idea behind a decision aiding system is to automate the process of decision making, and thus improve the quality of the decision making results.

1.2 Problem statement

Three main problems are discussed in this thesis:

1. The first step with this work is to propose the design and implementation of a decision aiding system that has the ability to assist a decision maker by recommending decisions in classification and sorting problems. The basis of the recommendations that the system gives, stem from example decisions originating from historical data or from the preferences of the decision maker.

2. For the second step with this work, two assumptions are made:

- Firstly, it is assumed that people prefer to make qualitative examples of how they make their decisions, thus example decisions thart stem from people have a qualitative form
- Second, it is assumed that the nature of most of the multiple criteria decision problems in the real world is quantitative.

In the light of these assumptions, already existing decision aiding systems [ref] assumes that qualitative example decisions can only be used as a basis in decision problems with a qualitative characteristic, making them not suitable for using a qualitative human preference model as a basis for recommending decisions in quantitative real-world decision problems. The second step with this work is therefore to propose a solution to the problem of decision making in real-world quantitative problems on the basis of qualitative decision examples provided by people.

3. Step three of this work is to find out study if such a decision aiding system could be used to improve decision making problems in the oil and gas industry by employing the it in a real-world decision making problem.

These problems will be examined by using example case studies for each problem. In addition, for problems 1 and 2, the proposal of the design of a decision aiding system will be presented, and thus how the system can tackle the problems mentioned in this section.

1.3 Thesis overview

Chapter 2, **Background** will give an introduction to multiple criteria decision analysis, and describe concepts within the domain of this thesis.

Chapter 3, Theory will introduce the relevant theory used in the thesis.

Chapter 4, System setup will describe the high-level design and implementation of the decision aiding system. The chapter also incudes one section that describes the training process of the system, an the two last sections are dedicated to two decision making processes in the system.

Chapter 5, Results introduces three case studies that approaches and solves the problems discussed in this thesis. The chapter also reviews a case study comparison of artificial neural networks and the decision aiding system.

Chapter 6, Conclusions summarizes the thesis and points to possible future work

2 Background

This chapter will describe some well known concepts used in this thesis. . The first section briefly introduces multiple criteria decision analysis. Then the second section describes the purpose of a decision aiding system. The third section introduces information matrices. The explanation of decision rule approach is introduced fifth, before the related systems section is discussed last.

2.1 Multiple criteria decision analysis

Many complex real-world problems are characterized as decision making with multiple, conflicting and non commensurate objectives. The nature of decision problems that a decision maker, and humans in general, usually faces are based on multiple attributes [2]. When making decisions in such problems, it is necessary to take into consideration several points of view, for example human preferences. The points of view can be represented in an information matrix that can be used as a basis for making a decision. The characteristic of the points of view in such information matrices may be either quantitative or qualitative, and the corresponding value set describing a point of view might have a nominal or an ordinal scale. Multiple criteria decision analysis [1] is a research field that provides logical and well structured theories and techniques for dealing with such complex decision problems. The basis of the problems are simple however. They consist of one finite or infinite set of alternatives, and at least two criteria, and at least one decision maker is involved.

Two such complex problems are looked at in this thesis, namely classification and sorting problems [3]. Those problems deal with making decisions by the assignment of objects to one class of a set of predefined decision classes, on the basis of points of view regarding the decision.

2.1.1 Classification problems

Classification problems [3] are multiple attribute decision problems, meaning that objects in classification problems are described by a set of regular attributes, and the value set of the regular attributes constitutes a description of the object. The final aim of classification problems is to assign the objects to exactly one predefined decision class based on the object's description. Classification problems are also known as nominal classification problems.

2.1.2 Sorting problems

Sorting problems [3] are closely related to classification problems, however, in distinction, the final aim of sorting problems is to sort objects from best to worse, or vice versa. This implies that a preference relationship among the decision classes must be considered. For this reason, objects in sorting problems are described by a set of criteria, not regular attributes, and the value on the object's criteria constitutes the description of the object. Criteria are special attributes where preference relationship is taken into consideration. Due to the preference ordering on the criteria, improvement on the values that describe the object should not worsen the sorting rank of the object. Sorting problems are also known as ordinal classification problems.

2.2 Decision aiding system

Decision aiding can be defined as being the activity of the person who, through the use of explicit but not necessarily completely formalized models, helps obtain elements of responses to the questions posed by a stakeholder in a decision process. These elements work towards clarifying a decision and usually towards recommending, or simply favoring, a behavior that will increase the consistency between the evolution of the process and this stakeholder's objectives and value system [2].

Based on the description of an object, decision aiding systems thereby can be used in classification problems to recommend the assignment of an object to exactly one decision class from a set of predefined decision classes, where the basis of the recommendation is based on already existing examples. In a sorting problem, the system has the ability to recommend the sorting of an object to exactly one preference ordered decision class from a set of predefined preference ordered decision classes, where the basis of the sorting is based on already existing examples.

2.3 Real-world decision aiding in accordance to human preferences

Real-world decision aiding in accordance to a human preference model means that the basis of the assignment of an object to a decision class, made in for example a real-world decision making problem, stem from a human preference model. This indicates that a human preference model must be provided beforehand of the decision aiding process. One way of understanding the human preference model is to request a set of examples of how they prefer to make their decisions. The examples provided can be analyzed, thus resulting in a knowledge base that can be used as the basis of the recommendation of decisions in real world decision making problems. Further, an assumption is made regarding the characteristic of the examples provided: It is assumed that people prefer to provide qualitative examples of how they make their decisions. This assumption corresponds well to the fact that preference models are formal representations of comparison of objects established through the use of a formal and abstract language [4]. Another assumption is made with respect to real-world decision making problems: It is assumed that real-world classification problems usually have a numerical character. This assumption makes real-world decision aiding in accordance to a human preference model a challenge in a decision aiding system, because linguistic examples have to be transformed into numerical numbers.

Example This example shows in Table 1 a human preference model of two electrical cars. The cars are described on two criteria, namely *Range* and *Top speed*, and thus based on the value on the criteria, assigned to a decision class stating the *Price* of the two cars.

Car	Range	Top speed	Price
1	good	high	high
2	low	low	low

Table 1: Electrical cars described by a human preference model

Table 2 presents the a real-world version of the the two electrical cars. They are described by numerical value sets on the condition attributes, and the decision attribute *Price* is also numerical.

Car	Range	Top speed	Price
1	160 km	120 km/h	300.000
2	40 km	80 km/h	120.000

Table 2: Electrical cars described by a real-world model

Real-world decision aiding in accordance to human preferences means taking a real-world object described by numerical attributes as in Table 2 as input, and then assign the object to a predefined numerical decision class, based on the information in a human preference model as in Table 1.

2.4 Information matrix

Problems within the multiple criteria decision analysis domain are usually structured within information matrices. The separate rows of the information matrix refer to distinct objects, where every object store some associated information. Simply stated, the information matrix is an $i \times j$ matrix, where the rows corresponds to objects, and columns corresponds to attributes. More formally as presented in [5], an information matrix can be defined as a 4-tuple $S =< U, Q, V, f >$, where each tuple has the meaning:

- U is the finite set of objects, alternatives or actions, also called Universe, of interest.
- $Q = \{q1, q2, ..., q_i\}$ is a finite set of i attributes. The set Q is further divided into two disjoint classes, C and D, called condition and decision attributes.. Bot sets C and D are not empty, $C \neq \emptyset$, $D \neq \emptyset$, and both sets are unique, $C \cap D{=}\emptyset$, hence $C \cup D = Q$. Furthermore, condition attributes are those used to describe the characteristics of the objects. The decision attributes define a partition of the objects into groups according to the condition attributes. The distinction of the sets is made with the aim of explaining the evaluations on D using the evaluations on C.
- V_q is the domain of the attribute $q \in Q$ and $V = \cup_{q \in Q} V_q$.
- The function $f : U \times Q \to V$ is such that $f(x, q) \in V_q$, where $q \in Q$ and $x \in U$. The function f is called informal function.

2.5 Decision rule approach

A technique within multiple criteria decision analysis for dealing with classification and sorting problems, is the decision rule approach [5]. A decision rule approach analyzes existing exemplary decisions and computes a set of logical decision rules on the form *if-then*. The left hand side of a decision rule is called the condition part, and the right hand side of the rule is called the decision part. The left hand side of the rule may have several conditions. The conditions of a decision rule are defined as $f(x)\, relation\, to\, constant$, where $relation$to is a relational operator from the set $\{=, \leq, \geq\}$, and the constant being a value of attribute $f(x)$. An example decision rule with two conditions can be as follows:
$If\ A\ =\ 2\, and\, B\, is\ \geq\ 2\, then\, Product\ \geq\ 4$.

An induction of decision rules from a universe of decision examples can be compared to artificial intelligence, defined in [6] as being the study of intelligent behavior. Thereby, the resulting set of decision rules constructs a knowledge base that can be utilized by a decision aiding system to make intelligent decisions. The decision rules induced from examples covers the whole set of objects in the example set, and are able to assign all of the example objects, and never before seen objects, to their decision class based on only the description of the object. This is done by matching the condition of a rule to the description of an object, thus if it is a match, the decision part of the rule holds for the object. The process of matching rules to objects is described more thorough in chapter 4.

2.6 Related systems

Two related decision aiding systems implementing a decision rule approach to classification and sorting problems is ROSE [7, 8] and jMAF [9]. Both systems takes information matrices as input and use it as a basis for decision making problems. Firstly, ROSE is a software written in C++ implementing basic elements of the classical rough set theory and rule discovery techniques. The system contains several tools for rough set based knowledge discovery. Among these are the ability to induce sets of decision rules from rough approximations of decision classes, and use the sets of decision rules as classifiers.

jMAF is a multiple-criteria and attribute analysis framework written in the Java language. The system implements methods of analysis provided by the dominance rough set approach. The system has the ability to resolve multiple criteria sorting problems.

None of these software systems are open source material, hence it was necessary to make own implementations of the techniques that these systems offer to be able to perform the necessary experiments.

3 Theory

This chapter describes the fundamental theories used in this thesis, namely classical rough set theory, dominance rough set approach, and fuzzy logic.

3.1 Classical rough set theory

The classical rough set theory (CRST) [10, 11, 12] was developed by Zdzislaw Pawlak in 1982. The theory deals with describing the dependencies between attributes, the significance of attributes, as well as inconsistent data. The theory was chosen in this thesis because it has the ability to support nominal classification problems.

3.1.1 Indiscernibility relation

The indiscernibility relation is a mathematical basis concept of the rough set theory. Given an information matrix $S =< U, Q, V, f >$, two objects $x,\ y \in U$ are said to be indiscernible (similar) if and only if they are described by the same information, hence they represent redundant data. More formally, the function $f(x, q) = f(y, q)$ for every $q \in P \subseteq Q$. Any subset P of Q determines a binary relation I_p on U. This relation is called an indiscernibility relation and is defined as $(x,\ y) \in I_p$. I_p is an equivalence relation for any P.

Any set of all indiscernible objects is called an elementary set, and it constitutes a basic granule of knowledge about the data in the universe. Equivalence classes of the relation I_p are referred to as P-elementary sets in S, and $Ip(x)$ denotes the P-elementary set containing object $x \in U$.

3.1.2 Lower and upper approximation of decision classes

The principle of rough approximation of decision classes in classical rough set theory is allowing to take inconsistency into the data analysis process by using the introduced indiscernibility relation. For each decision class, two rough approximations, namely the lower approximation and upper approximation, are calculated. The aim is to include in the lower approximation only those objects which are consistent, meaning that they certainly belongs to the decision class, and in the upper approximations objects that possibly belong to the decision class. The difference between the lower and upper approximations of decision classes defines a region of objects that cannot be certainly classified into one decision class.

More formally, if $P \subseteq Q$ and $Y \subseteq U$, then the P-lower approximation and P-upper approximation of Y can be defined as:

- $\underline{P}Y = \{x \in Y \ : \ I_p(x) \in Y\}$
- $\overline{P}Y = \bigcup_{x \in Y} I_p(x)$

The P-boundary, which means the doubtful region of Y, is defined as follows

- $Bn_p(Y) = \overline{P}Y - \underline{P}Y$

The accuracy of a rough set Y , denoted $\alpha_y(Y)$, can be estimated by calculating the ratio of the number of objects belonging to the lower approximation to the number of objects belonging to the upper approximation:

- $\alpha_y(Y) = \frac{|\underline{P}Y|}{|\overline{P}Y|}$

The subsequent steps of the analysis of the approximation of rough sets involve the development of a set of rules for the classification of the alternatives into the groups that they actually belong.

3.1.3 Decision rules

The lower and upper approximation of decision classes are sets that can be used in decision rule algorithms. Certain decision rules are induced from the lower approximations, and possible rules are induced from the upper approximations. One strategy to generating the decision rules, is to generate the minimal set of decision rules that satisfy the correct classification of example objects from an information matrix. Minimal means that no rule covers a subset of objects of another rule using weaker or the same strength on conditions, given that they both cover the same approximation. For extracting the decision rules from the information matrix, an algorithm called Modlem [13] is used. The procedure of the algorithm is shown in Algorithm 3.

3.2 Dominance rough set approach

Dominance rough set approach (DRSA) was proposed by the authors of [5] as an extension of classical rough set theory. The theory has the ability to deal with preference order in the value sets that describe objects, in comparison to classical rough set theory that cannot. From this, dominance rough set theory was chosen in this thesis to support ordinal classification problems.

3.2.1 Decision class unions

Given an information matrix $S =< U, Q, V, f >$, decision attributes D makes a partition of U into a finite number of classes $Cl = \{Cl_t, t \in T\}$, with $T = \{1, ..., n\}$ and $Cl_t = \{x \in U : f(x, d) = t\}$, with $x \in U$ belonging to one and only one class $Cl_t \in Cl$. The classes from Cl are preference ordered according to increasing order of class indices, i.e. for all $r, s \in T$, such that $r > s$, each object from Cl_r are preferred to the objects from Cl_s. Given this definition, two sets used in dominance rough set approach for approximation of the unions $Cl_t^{\geq}$ and $Cl_t^{\leq}$can be defined:

- Upward unions of classes, defined $Cl_t^{\geq} = \cup_{s \geq t} Cl_s$.
  An object $x \in Cl_t^{\geq}$means that x belongs to class Cl_t or better.

- Downward unions of classes, defined $Cl_t^{\leq} = \cup_{s \leq t} Cl_s$.
  An object $x \in Cl_t^{\leq}$means that x belongs to class Cl_t or worse.

3.2.2 Dominance principle

Using the definitions from the previous section with respect to criteria from set C, sets of objects dominating or dominated by a particular object can be defined. This definition is the dominance principle. It is said that object x P-dominates object y, if and only if $x \succeq_q y$ for all $q \in P$ (denotation $xD_py \Leftrightarrow x \succeq_q y, \forall q \in P$), where $P \subseteq C$, then object x should have a comprehensive description at least as good as object y.

- P-dominating set:
  $D_p^+(x) = \{y \in U : yD_px\}$
  Representing the set of objects that outrank x.

- P-dominated set:
  $D_p^-(x) = \{y \in U : xD_py\}$
  Representing the set of objects that x outranks.

The dominance principle hence requires that an object x dominating object y on all attributes $(x \succeq_q y)$, also dominate the decision of object y. These objects are called consistent, and those objects not satisfying the dominance principle are called inconsistent. Because there might be inconsistent objects in an information matrix, the concept of *rough approximations* is defined.

3.2.3 Rough Approximations

The concept of rough approximations in DRSA deals with inconsistencies with respect to the dominance principle. The formal expression of rough approximations is stated:

- P-lower approximation of Upward union:
  $\underline{P}(Cl_t^{\geq}) = \{x \in U \ : \ D_p^{+}(x) \subseteq Cl_t^{\geq}\}$.
  This definition means that an object x certainly belongs to Cl_t or better, if there is no object belonging to Cl_{t-1} that P-dominate x.

- P-upper approximation of Upward union:
  $\overline{P}(Cl_t^{\geq}) = \{x \in U \ : \ D_p^{-}(x) \cap Cl_t^{\geq} \neq \emptyset\}$.
  This definition means that an object x possibly belongs to Cl_t or better, if there exist an object that belongs to Cl_{t-1} or better, and that x P-dominates.

- P-lower approximation of Downward union:
  $\underline{P}(Cl_t^{\geq}) = \{x \in U \ : \ D_p^{-}(x) \subseteq Cl_t^{\leq}\}$.
  This definition means that an object x certainly belongs to Cl_t or worse, if all the objects that x P-dominates also belong to Cl_t or worse.

- P-upper approximation of Downward union:
  $\overline{P}(Cl_t^{\geq}) = \{x \in U \ : \ D_p^{+}(x) \cap Cl_t^{\leq} \neq \emptyset\}$.
  This definition means that an object x possibly belongs to Cl_t or worse, if there exist an object that belongs to Cl_{t-1} or worse, that P-dominates x.

In the case of inconsistencies, the boundaries between the upper and lower approximations $Bn_p(Cl_t^{\geq})$ and $Bn_p(Cl_t^{\leq})$ are defined. Inconsistency means that the examples cannot be certainly classified (also called doubtful regions). The denotion of the boundaries are:

- P-boundary of Upward union:
  $Bn_p(Cl_t^{\geq}) = \overline{P}(Cl_t^{\geq}) - \underline{P}(Cl_t^{\geq})$.

- P-boundary of Downward union:
  $Bn_p(Cl_t^{\leq}) = \overline{P}(Cl_t^{\leq}) - \underline{P}(Cl_t^{\leq})$.

3.2.4 Decision rules

The lower and upper approximations of decision classes are sets that can be used to extract knowledge in terms of decision rules. Certain decision rules are induced from the lower approximations, and possible rules are induced from the upper approximations. In this thesis, the minimal set of decision rules that satisfy the correct classification of example objects from an information matrix are extracted using a popular algorithm called Domlem [14]. The general scheme of the algorithm is presented in Algorithm 1. Generally, the main procedure of the algorithm is repeated for a rough approximation set, generating a minimal set of decision rules.

3.3 Fuzzy Logic

Reasoning in fuzzy logic [15, 16**?**] is a a matter of generalizing the familiar two-valued logic statement that is either true or false, but not both. However, in fuzzy logic, a proposition may be either true or false, or have an intermediate truth-value, such as maybe true. Consider the question: Is Friday a weekend day? If the number 1 is a numerical value for yes, and 0 is for no, using fuzzy logic it is possible to answer the question by a value of for example 0.8, meaning that Friday is a weekend day for the most part, but not completely. From this, fuzzy logic is a method appropriate to make decisions where the boundaries of the basis of the decisions are not clearly identified. These properties of fuzzy logic makes it possible to use linguistic terms as the basis of a numerical decision, and thus the main reason why the fuzzy logic theory was chosen in this thesis. Also, since the fuzzy logic controller processes user-defined decision rules for making decisions, combining it with a decision rule approaches such as classical rough set theory or dominance rough set theory seems natural and straight forward.

3.3.1 Fuzzy membership functions

Fuzzy membership functions [16] are used to generalize the value of the degree of truth in fuzzy logic. The function itself can be an arbitrary curve whose shape can be define as a function that suits us from the point of view of simplicity, convenience, speed, and efficiency. The simplest membership functions are formed using straight lines, and the only condition a membership function must really satisfy is that it must vary between 0 and 1. The

degree of truth in fuzzy logic represents membership in fuzzy sets. A fuzzy set is an extension of a classical set. If X is the universe of discourse and its elements are denoted by x, then a fuzzy set A in X is defined as a set of ordered pairs. More formally:
$A = \{x, \mu_A(x) \,|\, x\, X\}$,
where $\mu_A(x)$ is called the membership function of x in A. The membership function maps each element of X to a membership value between 0 and 1. Consider again the question from the previous chapter; Is Friday a weekend day? Fuzzy membership functions are used to model to which degree Friday is a weekend day.

There are many ways to assign membership functions to fuzzy variables [13]. This thesis relies on human intuition, which is simply derived from the capacity of humans to develop membership functions through their own innate intelligence and understanding. Intuition involves contextual and semantic knowledge about an issue, these curves are then a function of context and the analyst developing them. For example, considering a temperature scale, if the temperatures are referred to the range of human comfort, one set of curves is present, and if they are referred to the range of safe operating temperatures for a steam turbine, another set will be present. However, the important character of these curves for purposes of use in fuzzy operations is the fact that they overlap.

3.3.2 Fuzzy rules

Decisions in fuzzy logic are based on matching the decription of objects to every rule in the fuzzy knowledge base. The knowledge base in fuzzy logic is a set of fuzzy rules [16] that assumes the form *If x is A then y is B*, or possibly with multiple inputs as follows: *If x is A and z is C then y is B.* On the left hand side of the rule, A and B are fuzzy sets included in the condition part of the rule, while x and y are both numerical inputs to the rule. On the right hand side of the rule, C is the fuzzy set of the decision part of the rule, while z is the overall conclusion of the rule. Fuzzy membership functions are used to determine if an input x belongs to the fuzzy set A, and thus the conclusion part z of belongs to the fuzzy set C by the same degree.

4 System setup and how it works

The first step with this work is to build a decision aiding system that has the ability to make decisions in classification and sorting problems. The second step with this work is to build the decision aiding system such that is has the ability to approach the problem of decision making in real-world problems on the basis of human preferences.

This first part of this chapter presents the technologies that is used in the system. The second part describes the logical design and implementation details, and then the third part of the chapter focus on how the system is trained, and the forth on how is it used. The last part of the chapter presents how the system approaches the problem of real-world decision aiding in accordance to a human preference model.

4.1 Technologies and Logical design

The decision aiding system uses two decision-rule approach techniques, namely the classical rough set theory and dominance rough set approach for creating a knowledge base. Then, to approach the problem introduced in section 2.3, the thesis proposes that the system takes advantage of combining the decision rule approach with fuzzy logic.

The implementation of the decision aiding system is meant to be used as a proof of concept, thus, the emphasize on the graphical user interface is on the functional and informative side, rather than being a well thought-through human machine interface. Dominance rough set approach functionality, the fuzzy logic controller, and the decision rule matcher have been implemented in Microsoft language C#. The fuzzy logic implementation used is from an open source library called s [17]. The classical rough set theory functionality used in the system stem from a software called Rose2 [7, 8]. The logical design of the system can be seen in Figure 1.

4.2 Training process

First of all, the decision aiding system needs training in order to learn how to make decisions. During the training process, the system first takes an information matrix as input and performs a decision rule analysis that results in the knowledge base of the system in terms of decision rules on the form *if-then*. Figure 2 presents a graphical presentation of the process.

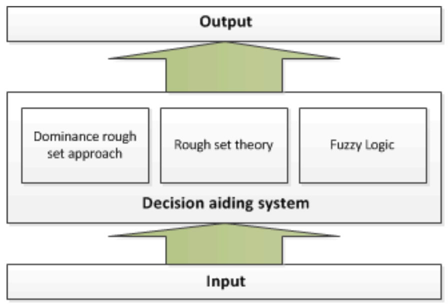


Figure 1: Logical design of decision aiding system

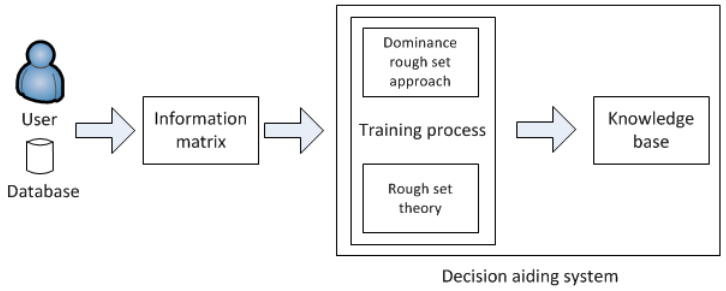


Figure 2: Training process

The knowledge base is the basis of the decision making capabilities of the system, and it represents the minimal set of rules possible to cover the set of exemplary objects in the information matrix. According to [17], minimal sets of decision rules represent the most concise and non-redundant knowledge representations.

From the role of the user of the system, minimal effort is required and thus simplified to present to the system the information matrix with exemplary decisions. By minimal effort, it is meant that the user does not have to get familiar with with theory basis of used analysis model in order to present the information matrix.

4.2.1 Algorithm for induction of decision rules

Two algorithms for induction of decision rules are used. The main procedure of the two algorithms is iteratively repeated over sets of lower or upper approximation of decision classes. For each loop, a best condition, or possibly several conditions, that cover only objects from the input set is found, which becomes the condition part of a decision rule.

Algorithm 1 [14] demonstrates the procedure used to induce decision rules in sorting problems. In the algorithm, $P \subseteq C$ and E denotes a complex (conjunction of elementary conditions e) being a candidate for a condition part of the rule. Moreover, $[E]$ denotes a set of objects matching the complex E. Complex E is accepted as a condition part of the rule if and only if $\emptyset \neq [E] = \cap_{e \in E}[e] \subseteq B$, where B is the considered approximation.

Algorithm 1 Rule induction procedure sotring problems

```
1: Procedure DOMLEM
2: (input : $L_{upp}$ - a family of lower approximations of upward unions of
   decision classes : $\{\underline{P}(Cl_t^{\geq}, \underline{P}(Cl_{t-1}^{\geq}, ..., \underline{P}(Cl_2^{\geq})\}$ ; output : $R_{\geq}$ set of - a
   set of $D_{\geq}$-decision rules) ;
3: begin
4:    $R_{\geq} := \emptyset$ ;
5:    for each decision rule $B \in L_{upp}$ do
6:    begin
7:       E :=find_rules(B) ;
8:       for each rule $E \in$ E do
9:       if $E$ is a minimal rule then $R_{\geq} := R_{\geq} \cup E$ ;
10:   end
11: end.
```

Algorithm 2 Function find rules as a part of rule induction procedure

```
Function find_rules
(input : a set $B$; output : a set of rules $\mathbf{E}$ covering set $B$
begin
  $G := B$; {a set of objects from the given approximation}
  $\mathbf{E} := \emptyset$;
  while $G \neq \emptyset$ do
  begin
    $E := \emptyset$; {starting complex}
    $S := G$; {set of objects currently covered by $E$}
    while $E \neq \emptyset$ or not $(E \subseteq B)$ do
    begin
      $best := \emptyset$; {best candidate for elementary condition}
      for each criterion $q_i \in P$ do begin
        $Cond := \{(f(x, q_i) \geq r_{q_i}) : \exists x \in S\ (f(x, q_i) = r_{q_i})\}$;
        {for each positive object from $S$ create an elementary condition}
        for each $elem \in Cond$ do
      if $evaluate(\{elem\} \cup E)$ is_better_than $evaluate(\{best\} \cup E)$
      then $best := elem$;
      end; {for}
      $E := E \cup \{best\}$; {add the best condition to the complex}
      $S := S \cap [best]$;
    end; {while not $(E \subseteq B)$}
    for each elementary condition $e \in E$ do
      if $[E - \{e\}] \subseteq B$ then $E := E - \{e\}$;
    create a rule on the basis of $\mathbf{E}$;
    $\mathbf{E} := \mathbf{E} \cup \{E\}$; {add the induced rule}
    $G := B - \cup_{E \in E}[E]$; {remove examples covered by the rule}
  end; {while $G \neq \emptyset$}
end; {function}
```

In function $evaluate(E)$, the complex E with the highest ratio $\mid [E] \cap G \mid / \mid [E] \mid$ is chosen. The complex E with the highest value of $\mid [E] \cap G \mid$ is chosen in case of a tie.

Algorithm 3 [13] demonstrates the procedure for induction of rules in classification problems.

Algorithm 3 Rule induction procedure classification problems

```
1: Procedure MODLEM
2: (input : $B$ - a family of lower or upper approximations; output : $\mathbf{P}$ - single local covering of $B$)
3: begin
4:   $G := B$; {examples not covered by conjunction from $\mathbf{P}$}
5:   P:= $\emptyset$;
6:   while $G \neq \emptyset$ do
7:   begin
8:     $P := \emptyset$; {starting complex} {candidate for condition part of the rule}
9:     $S := U$; {set of objects currently covered by $P$}
10:    while $P = \emptyset$ or not $([P] \subseteq B)$ do
11:    begin
12:      $best := \emptyset$; {candidate for elementary condition}
13:      for each attribute $a \in C$ do begin
14:        $new\_p := Find\_best\_condition(a, S)$;
15:        if $Better(new\_p, best, criterion)$ then $best := new\_p$;
16:        {evaluate if new condition $new\_p$ is better than previous $best$}
17:      end;
18:      $P := P \cup \{best\}$; {add the best condition to the condition part}
19:      $S := S \cap [best]$;
20:    end; {while not $(P \subseteq B)$}
21:    for each elementary condition $best \in P$ do
22:      if $[E - \{best\}] \subseteq B$ then $P := P - \{best\}$; {test minimality of the rule}
23:      $\mathbf{P}$:=$\mathbf{P}\cup\{P\}$; {add P to the local covering}
24:      $G := B - \cup_{P \in P}[P]$; {remove examples covered by the rule}
25:   end; {while $G \neq \emptyset$}
26:   for each $P \in \mathbf{P}$ do
27:   if $\cup_{P' \in P-P}[P'] = B$; then $\mathbf{P} := \mathbf{P}$ - $P$
28: end; {procedure}
```

4.3 Decision making process

The decision making process is the subsequent step after the training process. From the user of the system's perspective, an object is sent as input to the system, resulting in the system autonomously assigning it to exactly one predefined decision class. From the internal workings of the system's perspective, an input object is matched to each of the decision rules in the knowledge base in order to assign it to exactly one decision class. Algorithm 4 is used to match the input object to the decision rule set. The algorithm loops through each decision rule, and compares the description of the in object to the condition part of the rules. If there is a match, the rule supports the object, and the decision part of the rule states the recommending of the decision class of the object.

Algorithm 4 Matching decision rules to object description

```
1: Function match_object_to_rules
2: (input : $z$ - an object, $D$ - finite set of decision rules; output : $R$ - a
   set of decision rules that matches the description of object $z$)
3: begin
4:    $R := \emptyset$;
5:    for each decision rule $Rule \in D$ do
6:       for each condition of $Rule$ do
7:       if (all conditions match description of object $z$) ;
8:       then $R := R \cup Rule$;
9:       end ; {for}
10:   end ; {for}
11: end ; {function}
```

From the procedure of Algorithm 1 and 2, Let Cov_z be the set of decision rules covering a given object z. Three situations can occur when matching the description of object z to the set of decision rules:

1. no rule covers object z, $Cov_z = \emptyset$
2. one rule covers object z, $Cov_z = 1$
3. several rules cover object z, $Cov_z > 1$

The first situation means that the decision aiding system is not able to assign the object to a predefined decision class, because there are no decision rule to justify the decision. The object therefore may be assigned to any decision class. For the second and third situation, classification and sorting problems tackle them differently:

1. For classification problems;

   (a) The second situation for classification problems is straight forward. The object is assigned to the decision class that is recommended by the one rule that covers the object.

   (b) The third situation where several rules, indicating different classifications, matches an object, the rule with the highest strength is considered as the conclusive decision. The strength of the rule is a quotient of the support to all objects in the training set.

2. For sorting problems, situation 2 and 3 are tackled by using a method proposed in [18] that takes into account the strength of the rules suggesting an assignment to a class Cl_t as arguments in favor of Cl_t , and all other covering rules as arguments against Cl_t as following:

   (a) For the second situation, a score value, $Score_r(Cl_t, z)$, is calculated for each decision class to determine the most certain decision class Cl_t from Cl according to the followinig formula:
   $Score_p(Cl_t, z) = \frac{|Cond_p \cap Cl_t|^2}{|Cond_p||Cl_t|}$.
   From the formula, $Cond_r$ denotes the set of objects that the rule r supports, and $|Cond_r|$, $|Cl_t|$ and $|Cond_r \cap Cl_t|$ denote cardinalities of the corresponding sets: the set of objects verifying $Cond_r$, the set of objects belonging to class Cl_t and the set of objects verifying $Cond_r$ and belonging to class Cl_t. From this the definition of $Score_r(Cl_t, z)$ can be interpreted as a product of credibility CR_r and relative strength RS_r of rule r:
   $CR_r = \frac{|Cond_r \cap Cl_t|^2}{|Cond_p|}$,
   $RS_r = \frac{|Cond_r \cap Cl_t|}{|Cl_t|}$.
   A new object will be assigned to the class Cl_t for which the value of $Score_r(Cl_t, z)$ is the greatest.

   (b) The third situation is tackled similarly to situation two. A score value is calculated for each decision class to determine the most certain Cl_t from Cl.
   $Score_R(Cl_t, z) = Score_R^+(Cl_t, z) - Score_R^-(Cl_t, z)$.
   $Score_R^+(Cl_t, z)$ includes the decision rules that agrees with the assignment of the new object to class Cl_t. The following formula is defined:

$Score^+_R(Cl_t, z) = \frac{|(Cond_{pl} \cap Cl_t) \cup ... \cup (Cond_{pk} \cap Cl_t)|^2}{|Cond_{pl} \cup ... \cup Cond_{pk}||Cl_t|}$,
where $Cond_{pl}, ..., Cond_{pk}$ are the objects that the given rules support, and is analogus to $Score_r(Cl_t, z)$ for situation 2. Forthermore, the following formula is defined:

$$Score^-_R(Cl_t, z) =$$
$$\frac{|(Cond_{pk+1} \cap Cl^{\geq}_{pk+1}) \cup ... \cup (Cond_{pl} \cap Cl^{\geq}_{pl}) \cup (Cond_{pl+1} \cap Cl^{\leq}_{pl+1}) \cup ... \cup (Cond_{pk} \cap Cl^{\leq}_{pk})|^2}{|Cond_{pk+1} \cup ... \cup Cond_{pl} \cup Cond_{pl+1} \cup ... \cup Cond_{pk}||Cl^{\geq}_{pk+1} \cup ... \cup Cl^{\geq}_{pl} \cup Cl^{\leq}_{pl+1} \cup ... \cup Cl^{\leq}_{pk}|}$$

,
where $Cl^{\geq}_{pk+1}, ..., Cl^{\geq}_{pl}$ and $Cl^{\leq}_{pl+1} \cup ... \cup Cl^{\leq}_{pk}$ are all the upward and downward unions of decision classes that the rules that do not support the new object has suggested for assignment. Simply stated, $Score^-_R(Cl_t, z)$ is a product of credibility and relative strength of the rules that suggest that the decision class should not be Cl_t.
Similarly to situation 2, the greatest score value will determine the final decision class of the new object.

The decision aiding process can be used to measure how well the decision aiding system can perform. By presenting it to all the objects from the training set, and then compare the original decision class of the object to the decision class of the object recommended by the system.

4.4 Decision making process in real-world decision problems in accordance to human preferences

The decision aiding system approaches the type of problems explained in section 2.3 by taking advantage of the decision making capabilities of a fuzzy logic controller. The fuzzy logic controller has the ability to make numerial decisions from a knowledge base with a linguistic character.

Figure 3 shows a graphical presentation of the flow of two decision making processes. From top to bottom, the first process is discussed in this section, while the other process was discussed in the previous section.

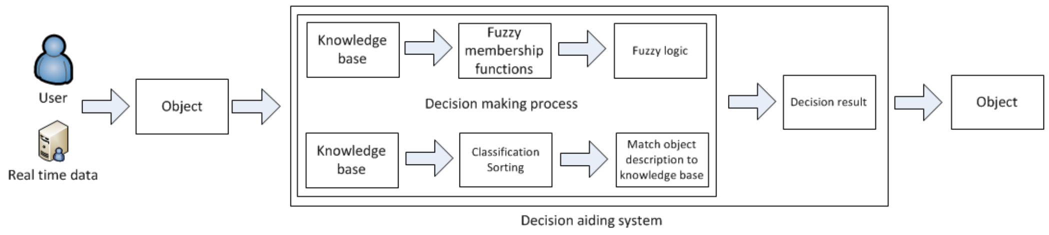


Figure 3: Decision making process

4.4.1 Fuzzy knowledge base

The first step in the set up process of the fuzzy logic controller for the system is to deal with induction of a knowledge base required by the fuzzy controller [15]. The main disadvantage of fuzzy logic systems is the possible difficulty in preparing the knowledge base for the system [16]. The knowledge base in a fuzzy logic controller usually consists of a set of human determined fuzzy rules, which can be complex to determine in case of many rules. However, for the fuzzy controller used in the decision aiding system in this thesis, the knowledge base is formed by taking advantage of the decision rule approach used during training of the system. That means using the set of linguistic decision rules resulting from the training process as the knowledge base in the fuzzy controller.

4.4.2 Fuzzy membership functions

The next step in the setup of the fuzzy controller after defining the knowledge base is to include fuzzy membership functions. This is a role that the user of the decision aiding system must act upon. In addition to providing a human preference model as a basis of the decision making process, the fuzzy membership functions required by the fuzzy logic controller must be provided. Figure 4 shows how the fuzzy membership functions presented to the system during the training process.

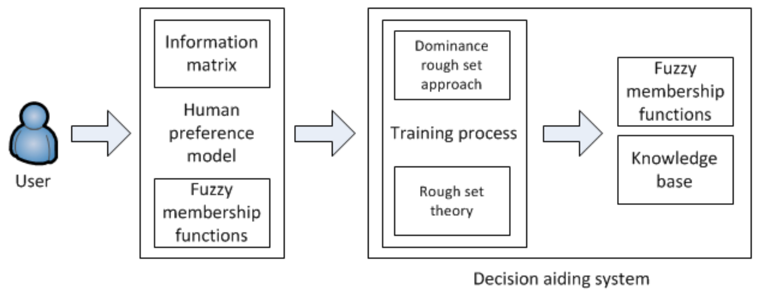


Figure 4: Presenting the fuzzy membership functions to the system

This means that the user constructs the membership functions for the linguistic terms used in the preference model accoring to his human intuition. The membership functions are then used in the fuzzy logic controller.

Consider two linguistic terms, A and B, used in the example fuzzy rule: *If A is High then $B \in$ Medium.*
A typical form of for the membership function for the proposition *A is High* is presented in Figure 5. The same figure also shows the forms for the propositions *A is Medium* and *A is Bad*.

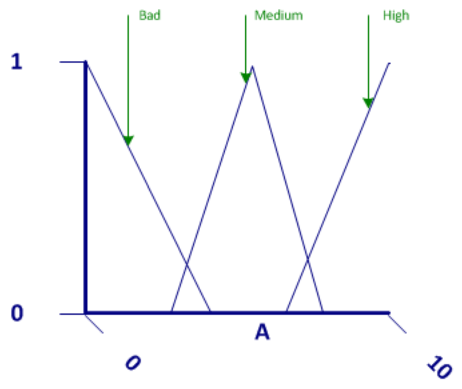


Figure 5: Membership functions for A

From the membership funtions of A in Figure 5, the x-axis shows that A is high to some degree if the numerical value of A corresponds to a value in the range of approximately $[6, 10]$. The membership function of the term B

used on the right hand side of the example decision rule is shown in Figure 6.

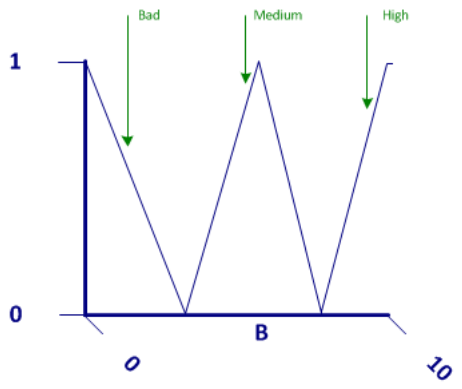


Figure 6: Membership functions for B

4.4.3 Fuzzy inference process

The decision making process in a fuzzy logic controller requires four steps of the fuzzy controller, called *the fuzzy inference process* [16]. The input to the fuzzy inference process is an object that has been given a numerical description. The result of the process is a crisp output number that corresponds to the decision class of the object. The basis of the knowledge used in the inference process is a linguistic human preference model.

Step 1: Fuzzification

Firstly, the numerical values that describe the input object are fuzzified. The fuzzifying process transforms the values into a number that represents to which degree they belong to a corresponding linguistic set, resulting in a number between 0 and 1.
Consider again the fuzzy rule *If A is High then B is High*. If an object was described by the linguistic term A with a value corresponding to the numerical value of 8, then the membership function in Figure 7 indicates that the proposition on the left hand side of the rule is partly true in accrodance to the description of the object. This can be seen in Figure 5, which presents the result of the fuzzification process, corresponding to a value of 0.7. The value indicates that proposition of the rule supports the object to a degree of 0.7.

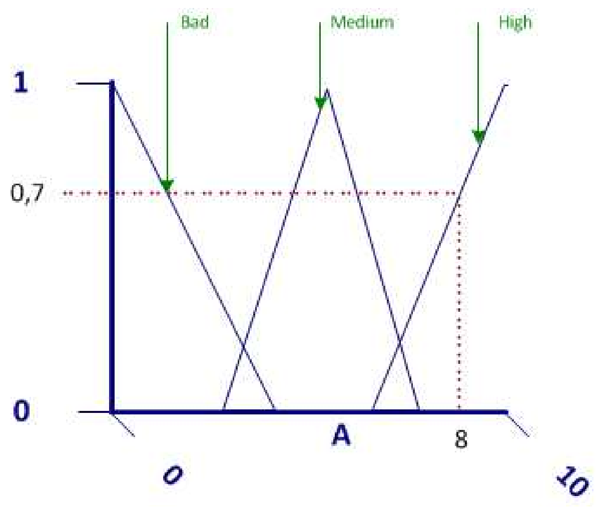


Figure 7: Fuzzification

If a rule has several conditions, such as *If A is High and B is Bad then C is Medium*, the fuzzification process applies a logical operator corresponding to the logical AND-operand, and as a result, the condition that has the lowest degree of support is used.

Step 2: Implication method

The implication method is performed for each rule in the knowledge base. This means reflecting the results from the fuzzification step on the output for each rule. The implication method used in the decision aiding system is a method that truncates the output fuzzy set.
Consider again the fuzzy rule used in the fuzzification step, in which supported an object to a degree of 0.7. The shaded area of Figure 8 demonstrates how the implication method truncates the output fuzzy set B in accordance to the fuzzification result.

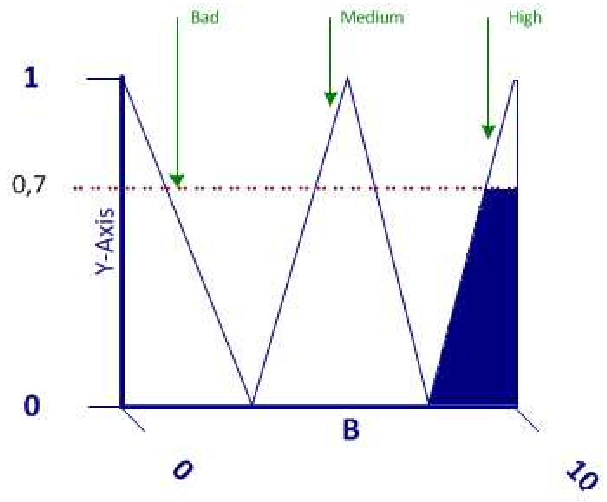


Figure 8: Implication

Step 3: Aggregation

The aggrecation step is used to derive an overall conclusion regarding the membership of an object into the fuzzy set based on the description of the object. According to the description of an object, several rules may support the object, resulting in several fuzzy sets as a result of the implication method. Simply stated, the aggregation step combines all these sets into one single fuzzy set by joining the maximum of each set.

Step 4: Defuzzification

Lastly, the defuzzification step transforms the single fuzzy set from the aggregation step into a single crisp value. This is done by applying a method called the center of gravity method. There are several defuzzification methods, however the center of gravity method is the most prevalent defuzzification method [16]. The center of gravity can be found using the following formula:
$Center\,of\,gravity = \frac{\int \mu_B(z) \times z dz}{\int \mu_B(z) \times dz}$,
where $\int \mu_B(z)$ denotes the integral of every resulting fuzzy set, and z is the correrspoonding x-axis value. Simply stated, if an area of a plate is considered as equal density, then the centre of gravity is the point along the x axis about which this shape would balance. Figure 9 shows

a red cricle representing the ceter of gravity of the shaded area. The shaded area is the result of the aggregation step in Figure 6.

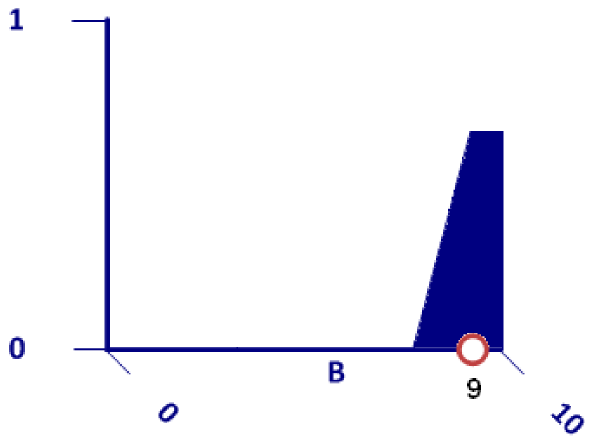


Figure 9: Aggregation and defuzzification

From Figure 9, the center of gravity indicates that the result of the fuzzy inference process is a numerical value of 9. Thereby, using a linguistic preference model in terms of the decision rules as a knowledge base in a fuzzy logic controller, makes the system able to make numerical decisions.

5 Results

This chapter present the results of the case studies performed in this thesis. The first part of the chapter presents a case study about making decisions in accordance to human preferences. The next part presents a case study where a human preference model is used as a basis of a real-world problem. Then, a case study from the oil and gas industry is presented. Lastly, the a case study on related systems are described.

5.1 Example of decision making based on human preferences

This example introduces a case study where the value of a residential property is to be estimated without knowing the exact numerical details of the property. The role of the user in this case study is therefore to provide a preference model describling example values that constitutes the basis of the problem. It is assumed that a specialist in the domain of the real estate market has provided his preference model of the value of 10 exemplary residential properties described by multiple criteria. The preference model can be seen in Table 3.

	Location	Size	Standard	Build year	Value
1	urban	big	good	recently	high
2	urban	small	good	old	medium
3	urban	medium	bad	old	medium
4	suburban	medium	excellent	recently	high
5	suburban	big	bad	recently	medium
6	urban	big	bad	new	high
7	urban	big	excellent	new	high
8	suburban	small	good	old	low
9	countryside	small	good	old	low
10	suburban	medium	Bad	old	medium

Table 3: Human preference model describing residential properties

More formally, the preference model in represented in an information matrix inculding a finite set of 10 objects U, described by the set of criteria Q:

- $Q = \{Location, Size, Standard, View, Value\}$

The set Q is further divided into a set C of condition criteria, and a set D of decision criteria:

- $C = \{Location, Size, Standard, View\}$
- $D = \{Value\}$.

Every object in U is assigned to a preference ordered and predefined decision class belonging to the domain of set D, according to the evaluation on the condition classes in set C. The domain of the classes in condition set C and the decision class in set D is as follows:

- $V_{Location} = \{countryside,\ suburban,\ urban\}$
- $V_{Size} = \{small,\ medium,\ big\}$
- $V_{Standard} = \{bad,\ good,\ excellent\}$
- $V_{Age} = \{old,\ average,\ new\}$
- $V_{Value} = \{low,\ medium,\ high\}$

There are monotonic relationships between the criteria meaning that for example a residential property in an urban area should have at least the same or a higher value than that of a residential property in a subruban area, hence it follows that preference order is from left to right.

Training process The first step that is performed during training of the system is empolying a decision rule approach to analyse the information matrix, or more precisely dominance rough set approach. The analysis process denotes knowledge discovery by using the decision attribute $Value = \{Low, Medium, High\}$. From this set, three classes can be identified: $Cl_1 = \{Low\}$, $Cl_2 = \{Medium\}$ and $Cl_3 = \{High\}$. Furthermore, four unions of classes denoted $Cl_1^{\leq}$, $Cl_2^{\leq}$, $Cl_2^{\geq}$ and $Cl_3^{\geq}$ are introduced. From the basis of the four unions of classes, the lower approximations of every union can be found:

- $\underline{P}(Cl_1^{\leq}) = \{2, 8, 9\}$
- $\underline{P}(Cl_2^{\leq}) = \{2, 3, 5, 8, 9, 10\}$

- $\underline{P}(Cl_2^{\geq}) = \{1, 3, 4, 5, 6, 7, 10\}$
- $\underline{P}(Cl_3^{\geq}) = \{1, 4, 6, 7\}$

The lower approxiamtions of classes are input to the algorithm from (x), resulting in a minimal set of decision rules. The decision rules can be seen in the following list, with the objects number that the corresponding rule supports stated in parenthesis:

If Build year $\leq$ Old then Value $\leq$ Medium
If Location $\leq$ Suburban and Standard $\leq$ Bad then Value $\leq$ Medium
If Size $\geq$ Medium then Value $\geq$ Medium
If Location $\geq$ Urban then Value $\geq$ Medium
If Size $\leq$ Small and Location $\leq$ Suburban then Value $\leq$ Low
If Standard $\geq$ Excellent then Value $\geq$ High
If Built $\geq$ New then Value $\geq$ High
If Location $\geq$ Urban and Size $\geq$ Big then Value $\geq$ High

As already stated, such rules corresponds to how people try to justify their decisions, hence they can be used to make decisions according to human preferences.

Using the system After training of the decision support system, resulting in the set of induced decision rules, it is possible to infer the value of a new residential properties based on the linguistic description of the property by using the decision rules. Following, a new residential property as can be seen in Table 4 is presented to the decision aiding system. It is of interest to know the value of the property according to the specialist in the real estate marked's preferences.

	Location	Size	Standard	Build year	Price
1	suburban	medium	good	recently	

TABLE 4: DESCRIPTION OF A RESIDENTIAL PROPERTY WITHOUT VALUE

The object is input to the decision aiding system. The description of the object is then matched to the set of decision rules, resulting in one rule that matches the object's description:

1. If Size $\geq$ medium then Value $\geq$ medium

The right hand side of the decision rule claims that the object should have a value of at least *medium*. From this claim, it is also possible to infer that the object could have a value of *high*. A classification scheme is applied to examine which value is the most suitable for the object. The $Score_p(Cl_t, z)$ is calculated for is calculated for each of the decision class low, medium and high:
$Score_p(Cl_t, z) = \frac{|Cond_p \cap Cl_t|^2}{|Cond_p||Cl_t|}$.
$Score_p(medium, z) = \frac{|1,3,4,5,6,7,10 \cap 2,3,5,10|^2}{|1,3,4,5,6,7,10||2,3,5,10|} = \frac{|3,5,10|^2}{|7||4|} = 0.32$.
$Score_p(high, z) = \frac{|1,3,4,5,6,7,10 \cap 1,4,6,7|^2}{|1,3,4,5,6,7,10||1,4,6,7|} = \frac{|1,4,6,7|^2}{|7||4|} = 0.57$
The method compares a ratio between objects verifying and objects not verifying the left hand side of the decision rule. The result of the classification scheme is interpretted as a degree of certainty of the assignment of the object to the value [Ref] indicating that the value of the new residential property is *medium* by a certainty of 0.32, and *high* by a certainty of 0.57. The value of the new property is therefore chosen to be *high*, as can be seen in Table 5.

	Location	Size	Standard	Build year	Value
1	suburban	medium	good	recently	high

TABLE 5: DESCRIPTION OF A RESIDENTIAL PROPERTY WITH A VALUE

5.2 Example of real-world decision aiding in accordance to human preferences

This case study is a continuance of the example in the previous section. The decision system from the previous example uses a human preference model to assign residential properties a vaule of either low, medium or high. The study in this section will use the same human preference model as a basis to give a residential property a value, when a numerical description of the property is known. The role of the user is simplified to preparation of the preference model, in addition to providing fuzzy membership functions.

Training The training process follows the same scheme as in with the previous example. This is because the same preference model is used as a basis in this example, and thus resulting in the same 8 decision rules, which constitutes the fuzzy knowledge base:

If Build year $\leq$ Old then Value $\leq$ Medium
If Location $\leq$ Suburban and Standard $\leq$ Bad then Value $\leq$ Medium
If Size $\geq$ Medium then Value $\geq$ Medium
If Location $\geq$ Urban then Value $\geq$ Medium
If Size $\leq$ Small and Location $\leq$ Suburban then Value $\leq$ Low
If Standard $\geq$ Excellent then Value $\geq$ High
If Built $\geq$ New then Value $\geq$ High
If Location $\geq$ Urban and Size $\geq$ Big then Value $\geq$ High

Using the system After training the system, the object in Table 6 is the input to the fuzzy inference process. The object is a numerical description of a residential property that is located 7 km from the city centre, has a size of $90m^2$, has a standard of 7/10, and is built in 2010. On the basis of the linguistinc human preferences in Table 3 given by the expert, the decision aiding system is to find the numerical value of the property based on its numerical description.

	Location	Size	Standard	Build year	Value
1	7 km	$90m^2$	7	2010	

Table 6: Numerical description of a residential property without a value

The fuzzy membership functions are used in the process to fuzzify the numerical values that describe the object. Figures 10 to 13 shows the membership functions for each criteria in Table 6, as well as an annotation describing to which degree an input number corrresponds to a fuzzy set. In the Figures, the linguistic traslation of the inequatilies $\geq$ and $\leq$ is *At least* and *At most*.

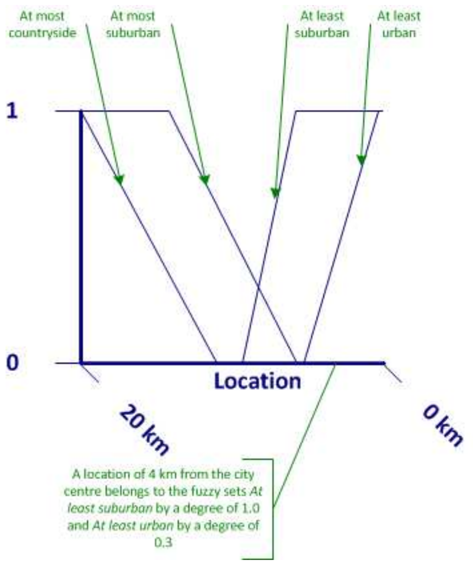

At most countryside
At most suburban
At least suburban
At least urban
1
0
Location
20 km
0 km
A location of 4 km from the city centre belongs to the fuzzy sets *At least suburban* by a degree of 1.0 and *At least urban* by a degree of 0.3

Figure 10: Membership functions for Location

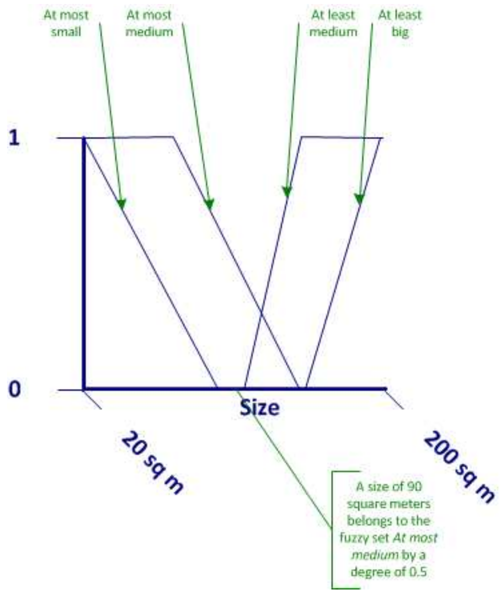

At most small
At most medium
At least medium
At least big
1
0
Size
20 sq m
200 sq m
A size of 90 square meters belongs to the fuzzy set *At most medium* by a degree of 0.5

Figure 11: Membership functions for Size

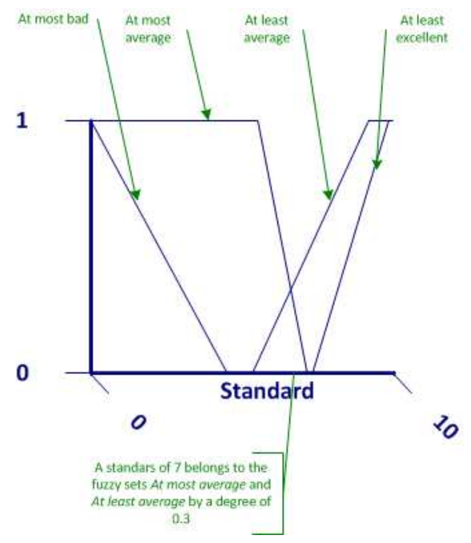


Figure 12: Membership functions for Standard

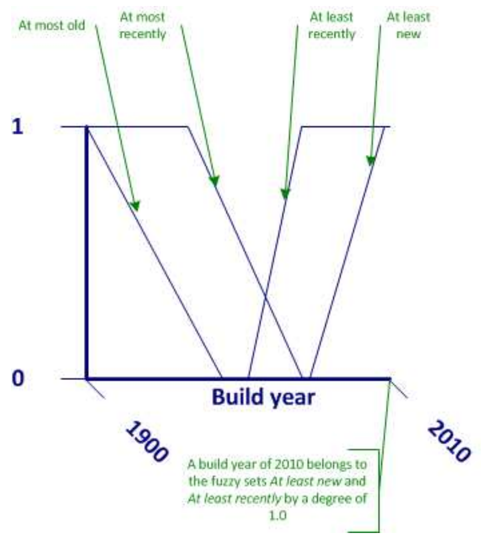


Figure 13: Membership function for Build year

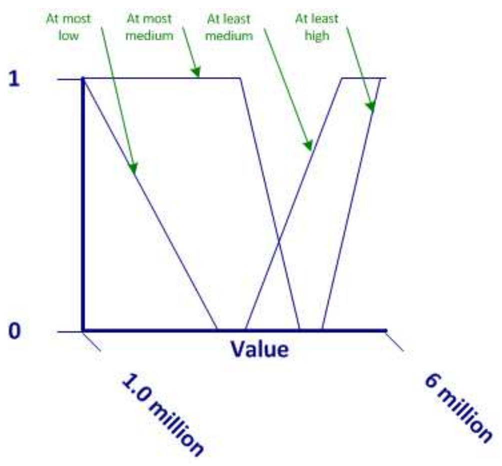


Figure 14: Membership function for Value

The fuzzy knowledge base is then used to analyze which decision rule that supports the object. The left hand side of the decision rules implies that the object belongs to a fuzzy set included in the rule by some degree. The following rules supports the new object:

1. If Build year $\geq$ New then Value $\geq$ High. The object belongs to the fuzzy set *At least New* by a degree of 1.0. This means that the rule supports the object by a degree of 1.0.

2. If Location $\geq$ Urban then Value $\geq$ Medium. The object belongs to the fuzzy set *At least Urban* by a degree of 0.3. This means that the rule supports the object by a degree of 0.3.

3. If Location $\geq$ Urban and Size $\geq$ Big then Value $\geq$ High. The object belongs to the fuzzy set *At least Urban* by a degree of 0.3, and the fuzzy set *At least Big* by a degree of 0.0. The implication method combines the two conditions in this rule, meaning that this rule supports the object by a degree of 0.0.

Two rules, rule number 1 and 2, affect the outcome of the valued of the object. Rule number one states that the value is *At least High* by a degree of 1.0. Rule number two states that the value of the object is *At least Medium* by a degree of 0.3. These statements are aggregated. The result is projected on the output value as can be seen in the shaded area of Figure 15

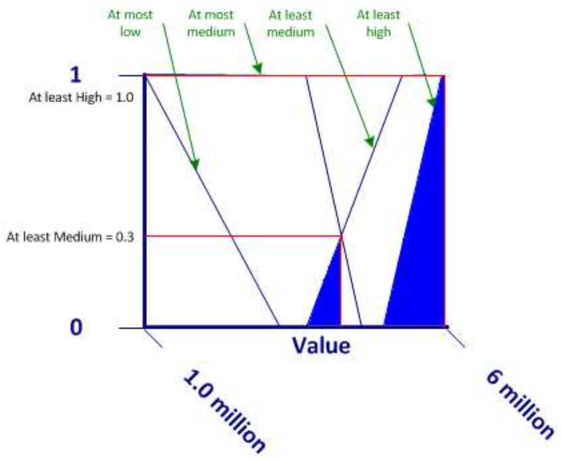


Figure 15: Graphical representation of the participation of each input reflected on the output

The center of gravity method is then applied to the output, finding the point where the center of the mass resides marked by a red circle. This point corresponds to 4.9 million, as can be seen in Figure 16.

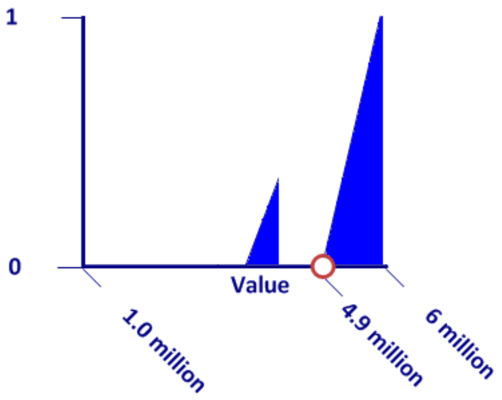


Figure 16: Defuzzification applied on the resulting output

5.3 Example of decision making in the oil and gas industry

With this case study, a decision making problem from the oil and gas industry is studied [19]. The problem occurs during drilling opertion. Abnormal surface torque and hook load values are symptoms of downhole drilling condition deterioration which can result in unexpected situations. Usually, friction tests are performed at regular intervals and rig personnel uses these measurements to monitor trend variations in order to detect possible risk of poor hole cleaning or increased borehole tortuosity, and thereby decect any problems. The quality of the detection can vary greatly in function of the work load and experience of the drilling staff. The availability of real-time measurements through data servers makes it possible to automate and systemize the monitoring process, and therefore trigger alarms before drilling problems really occurs. The task a decision aiding system has in this case is to indicate unexpected measurements several hours before a pack-off problem occur and therefore help the drilling staff in detecting the worsening of downhole drilling conditions.

Training process To demonstrates the data that is used for training the decision aiding system, Table 7 contains a selection of 5 examples out of 1130 examples recorded from former drilling operations. Of these 1130 examples, 912 are used to train the system, while 218 examples are left for measuring the performance of the system after the training process. As can be seen from Table 7, every object is described by 7 regular attributes with a numerical domain; BD = Bit Dept, FR = Flow rate, HL = Hook load, SPP = Stand pipe preassure, ST = Surface torque, ToS V = ToS Velocity, and S RPM = Surface RPM, and then assigned to one of two predefined decision classes, A or NA according to its description values. An example assigned to class A means that its description values indicates an acceptable friction level, while an example assigned to class NA means that its description values indicates a friction level that is not satisfactory. The 912 decision examples are presented to the decision aiding system for training. During the training phase, the 912 example decisions are analysed by utililizing classical rough set approach. The result of this process is the following set of 21 decision rules that supports every of the 912 objects in the example decisions. The rules can be seen in Appendix B.

	BD	FR	HL	SPP	ST	ToS V	S RPM	F
a	2684.3	0.0036	91543.6	676841.6	31.7	99.98	0.001	A
b	2970.1	0.0048	106162.2	923353.5	24.1	100.06	0.001	NA
c	3342.1	0.0496	84165.2	19412569.3	48.4	99.97	0.002	A
d	5314.9	0.0534	103088.5	24973968.6	53.1	100.02	0.002	NA
e	3007.8	0.0308	102439.1	8946768.6	24.1	99.98	0.003	NA

Table 7: Samples adapted from actual drilling operations

Using the system The last step is a performance test for measuring how well the system is performing. This is done by first presenting it to the same 912 objects already used during training, however the decision class is not present. Then, 218 objects never before seen by the decision aiding system is presented.

The first 912 objects were all classified according to their original decision classes, hence the decision rules covers all the objects in the training set. Further, 3 out of 218 never before seen examples could not be classified using the rough set approach, analogus to 98,68% correct classification. One of those three could not be classified because it was not covered by any rule. The other two was recommended the assigment of both decision classes, meaning that several rules stating different decisions classified the examples.

The result is a system that makes over 98% correct decsisions according to presented data.

5.4 Artificial neural networks

This section is intended to present work that that is related to and carries out the same decision aiding functions as the system presented in this thesis. An Artificial neural network (ANN), or simply neural network (NN), was assumed to possibly match the range of application. To test this assumption, two problems presented in the next sections will be solved using an ANN. The results will then be compared to the results from solving the same problems using the classical rough set theory as used in the decision aiding system. To be able to accurately compare the two results, the case studies are identical with respect to data for both theories. The ANN was created using a Neural Network Toolbox provided with the Matlab software [20].

According to [21], ANN are proficient classifiers and are particularly well suited for non-linear classification problems. They consist of units called neurons, connected together forming a network. The network is layered, accommodating at least one input layer and one output layer. Also, there might be one or two occurences of a layer called the hidden layer. The number of hidden layers are determined by the complexity of the problem the network comes across. One of the most common ANN architectures are the feedforward backpropagation neural network. This architecture is very popular because it can be applied to many different tasks. Also, a study in [22] showed that such an ANN provided the best classification performance compared to other neural networks. In the feedforward artificial neural network, every neuron is connected forward to the next layer. The input layer is connected to the hidden layer, which then connects to the output layer. Backpropagation is the way the network is trained, which is a way of supervised training. Supervised training means using both a sample output data and anticipated output data that together measure the performance of the network during training. The anticipated output are compared to the actual output, and the backpropagation training algorithm then takes a calculated error and adjusts the the interconnection of the various layers backwards from the output layer to the input layer [Ref]. The artificial neural network is trained as long as the calculated error is decreasing, meaning its performance is improving. After training the network, real world data can be presented and computed.

5.4.1 Expectations

Since ANN is such a vast used technology [20] and a common research topic [Ref], the expectations of it performing well is high. Before starting the study of the artificial neural network model for the purpose of this section, the impression of a neural network was that it works like kind of a black box, where it is presented to some input data, performing its algorithm, and then presenting the outcome without showing how it was reasoned. Also, since the subject is very extensive, the construction process was expected to take some time in order to get the network working.

5.4.2 Case Study

The case study in this section examines two problems that will be solved by using a feedforward backpropagation ANN. The network consist of one input layer with one neuron for each input attribute, connected to one hidden layer.

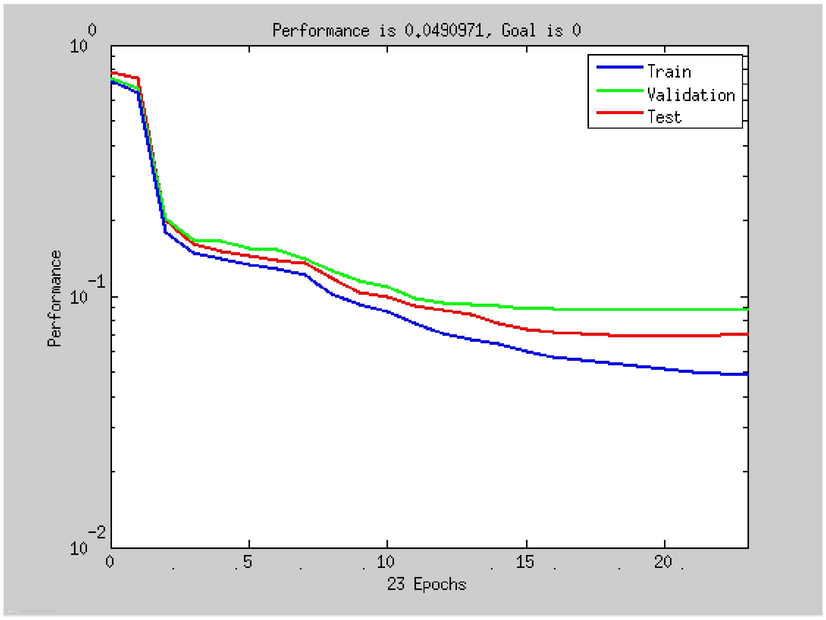


Figure 17: Artificial neural network training performance

The hidden layer will have 20 neurons, but since the number of neurons used in the hidden layer is somehow arbitrary [Ref], the network performance will be tested by using considerable numbers of neurons in this layer. The hidden layer is then connected to one output layer with one neuron, corresponding to the classification result.

Moreover, the data presented to the artificial neural network, is divided further into three sets: *Training*, *validation* and *testing*. *Training* takes up 60% of the data, *Validation* takes up 20%, while the *Testing* set is the reminding 20% of the data. The ANN is then trained using both the *Training* set and the *Validation* set. The *Validation* set is constantly used during training to determine how error prone the network is, and to determine if the performance is improving. The training of the network continues as long as the network's error on validation is decreasing, meaning the performance is improving. The training of the network with 20 neurons in the hidden layer is presented by Matlab as a graph shown in Figure 17. The green line levels out when the network is performing at its best, thus training terminates. After training, the network is ready and can be tested using the *Testing* data samples. The test will show how well the network performs on data from the real world.

Case study 1: For case study 1, the example from the case study in section 5.3 is adopted, thus the results from that study are used in this section.

Case study 2: This problem is adopted from [21]. The problem studies using an ANN to identify the sex of crabs from physical measurements of the crab. Six physical characteristics of a crab are considered: *Species*, *Frontallip, Rearwidth*, *Length*, W*idth* and *Depth*. The problem on hand is to identify the *Gender* of a crab given the observed values for each of these six physical characeristics. Table 8 includes two crabs for demonstation purpose.

Crab	Species	Frontallip	Rearwidth	Length	Width	Weight	Gender
1	0	20.6	14.4	42.8	46.5	19.6	male
2	1	19.9	16.6	39.4	43.9	17.9	female

Table 8: Crab data example

For the sake of simplicity, a set of 160 out of the 200 examples will be picked and used for the training purpose, whereas the remaining 40 examples then will be used as a key book to measure the correctness of the classifications.

5.4.3 Results

The results gathered from the study is summarized in Table 9 and 10.

Theory	Test samples	Train samples	Classification
Artificial neural network	228	912	95,18%
Rough set approach	228	912	98,68%

Table 9: Comparison margin of error case 1

From case study 1, 3 out of 228 testing examples could not be classified using a classical rough set approach, analogus to 98,68% correct classification. One of those three could not be classified because it was not covered by any rule. The other two was classified as both classes, meaning that decision rules stating different decisions classified the examples. For the artifical neural network, 11 out of 228 testing examples was not classified correctly.

Theory	Test samples	Train samples	Classification
Artificial neural network	40	160	95%*
Rough set approach	40	160	100%

Table 10: Comparison margin of error case 2

From case study 2, all testing examples was correctly classified using the classical rough set theory, analogus to 100% correct classification. With the ANN,, 2 out of 40 testing examples were not classified correctly. From these results, it tured out that classifying the data using classical rough set theory proved to be more efficient than that of an ANN. The deviation of the results are not gigantesque, however it seemed to reveal a drawback with neural network models. Using decision rules as with classical rough set theory, one can easily make something out of the cause of the classification error, because decision rules are understandable for humans. Either the classification error is because no rule matches, or several rules matches and indicated different classification. With ANN models, the reason of classification error does not seem so obvious. For example, adding an error example to the training data could lead the network to classify other examples differently. Also, the only really possibility to improve performance with the ANN was to tweek the number of neurons in the hidden layer, or even change the number of hidden layers. Thus, none of these methods resulted in better performance from the artifical neural network, as can be seen in section 5.4.5.

5.4.4 Comments on expectations

From early on, it was clear that the neural network would perform well, as expected. Since using Matlab for constructing the ANN, some time was spent on getting familiar with the software, but the Neural Network Toolbox had a very good documentation, making it pleasant to work with, and less time consuming than expected. Lastly, the expectation of the black box feel to the ANN was present, as errors on the output was detected, but not very easily corrected.

5.4.5 Number of neurons in the hidden layer

The performance of an ANN is said to vary according to the number of neurons in the hidden layer. In [23], an empirical approach to find the most suitable number of neurons in the hidden layer is suggested. A test was done during training of the ANN for case study 1 to compare performance of the network measured using from 1 to 20 neurons in the hidden layer. A strict selection of the result is shown in Table 11. From Table 11, 12 neurons in the hidden layer resulted in the best performance.

Hidden neurons	Performance (Correct classification)
10	90,35%
12	95,18%
14	92,10%
20	93,42%
22	90,35%
24	94,29%
26	89,91%
28	93,42%

Table 11: Hidden layer neurons

The same test was applied to within case study 2, however with no performance improvement within 10 to 28 neurons.

6 Conclusions

6.1 Summary

To summarise the main points of the work in this thesis, the three problems from the problem statement section is brought back:

1. The first step with this work is to propose the design of a decision aiding system that has the ability to assist a decision maker by recommending decisions in classification and sorting problems. The basis of the recommendations stem from example decisions originating from historical data or from the preferences of the decision maker.

2. The second step with this work is to propose a solution to the problem of decision making in real-world quantitative problems, on the basis of human preferences.

3. Step three of this work is to find out how such a decision aiding system could be used to improve decision making problems in the oil and gas industry, by employing the it in a real-world decision making problem.

With this work, a decision aiding system utilizing a decision rule approach was implemented. The system has the ability to assist a decision maker by recommending decisions in classification and sorting problems. The basis of the recommendations stem from example decisions originating from historical data or from the preferences of the decision maker.

The work in this thesis also proposed a method combining a decision rule approach with fuzzy logic. The proposed method had two advantages: First, by constituting a fuzzy logic knowledge base from a decision rule approach, it was possible to solve the problem of decision making in real-world quantitative problems on the basis of qualitative decision examples. Second, the design of a fuzzy knowledge base is a difficult step in the creation of a fuzzy logic system [15], and a decision rule approach may be a good tool to use also for creating the knowledge base for standalone fuzzy logic systems. An experimantal case study was demonstrated in section 5.2, showing that the proposed method can find an exact price of a house from the basis of a linguistic human preference model.

A real-world example case study was used in this work to demonstrate that a decision aiding system using a decision rule approach can improve the decision making process in the oil and gas industry. This example was also solved by using an ANN. The results were compared and discussed. Firstly, the decision rule approach had a better margin or error than the ANN. Second, errors in systems that use a decision rule approach seems to be easier for people to understand, because they make their decisions using rule syntax, which is understandable for humans. Lastly, training and constructing the ANN seemed to be more strenous than using the classical rough set theory, because it required application of an empirical method, which is time consuming. However, since the area of neural networks is extensive and complex, spending more time with it, making a more comprehensive study could improve the overall performance.

Comments on problem statement 2 For a validation of the proposed method for problem statement 2, an actual decision problem from the real-world was required. The presentation in Appendix A was given at the International Research Institute of Stavanger, with the purpose being a real-world case study suitable for applying of the method. As a result of the presentation, the problem provided in section 5.3 was proposed. This problem corresponds to the assumptions made about real-world decision problems, namely that most such problems have a quantitative charateristic. However, there was no preference order considered within the data presented in that problem, meaning that it was impossible to use a human preference model as a basis of decisions in that problem. Also, representing the data in the presented problem linguistically would be too complex for people. This lead to a conclusion that real-world case studies for problems as described in section 2.3 needs more studying. In theory, such problems seems easy to come by, however in real-world such problems seemes insignificant.

Traditionally, problems exist before its soultion, however in this case, the soulution to a problem was proposed before the problem existed, and a real-world problem that could validate the method was not found.

6.2 Future work

Problem statement 2 The following steps after this work is at first, to do more research on problems with characteristics as mentioned in section 2.3. The work in this thesis proposes a method on how to approach such problems. However, a real-world case study that discovers a quantitative decision problem and thus making decisions in such problems according to human preferences, needs to be present in further work. Thus, the three requirements for such problems must be fulfilled:

- A human preference model must be present.
- A corresponding real-world decision problem must exist.
- Preference relationship must be taken into consideration.

The results of such further work can be used for validation of the proposed method. Thereby, a benchmark study on the combination of a decision rule approach and fuzzy logic could be performed to verify the correctness of the numerical decisions that the system makes from regarding linguistic information.

Problem statement 3 As a natural course of the result presented in section 5.3, further work could involve deploying the decision aiding system in a real-world environment. Thereby, a more thorough study could be done showing the impact that the system may have over a longer period of time.

References

[1] M. Ehrgott, J. Figueira, and S. Greco, *Multiple Criteria Decision Analysis: State of the Art Surveys.* Springer-Verlag, Berlin, 2005.

[2] J. Figueira, S. Greco, and M. Ehrgott, "Paradigm and challenges," *Chapter 1 in Figueira J., Greco S., Ehrgott M. (eds.), Multiple Criteria Decision Analysis: State of the Art Surveys*, 2005.

[3] S. Greco, "Dominance-based rough set approach for decision analysis - a tutorial," *Rough Sets and Knowledge Technology*, pp. 23–24, 2008.

[4] M. Ozturk, A. Tsoukias, and P. Vincke, "Preference modelling," *Chapter 4 in Figueira J., Greco S., Ehrgott M. (eds.), Multiple Criteria Decision Analysis: State of the Art Surveys*, pp. 27–72, 2005.

[5] M. Ehrgott, J. Figueira, and S. Greco, "Multiple criteria decision analysis: State of the art surveys," *Chapter 13 in Figueira J., Greco S., Ehrgott M. (eds.), Multiple Criteria Decision Analysis: State of the Art Surveys*, 2005.

[6] A. Gernham, *Artificial Intelligence: An introduction.* Routledge, 1988.

[7] B. Predki, R. Slowinski, J. Stefanowski, R. Susmaga, and S. Wilk, "Rose - software implementation of the rough set theory," *Polkowski, Skowron (eds.), Rough Sets and Current Trends in Computing, Lecture Notes in Artificial Intelligence.*, vol. 1424, pp. 605–608, 1998.

[8] B. Predki and S. Wilk, "Rough set based data exploration using rose system." *Z.W.Ras and A.Skowron, eds. Foundations of Intelligent Systems, Lecture Notes in Artificial Intelligence*, vol. 1609, pp. 172–180, 1999.

[9] J. Blaszczynski, S. Greco, B. Matarazzo, R. Slowinski, and M. Szelag, "jmaf," 2009. [Online]. Available: http://www.cs.put.poznan.pl/jblaszczynski/Site/jRS.html

[10] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers.

[11] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron, "Rough sets: A tutorial," *Rough-Fuzzy Hybridization: A New Trend in Decision Making*, pp. 3–98, 1998.

[12] Z. Pawlak, "Rough sets, international journal of computer and information sciences," no. 11, pp. 341–356, 1982.

[13] J. Stefanowski, "The rough set based rule induction technique for classification problems." *Proceedings of 6th European Conference on Intelligent Techniques and Soft Computing EUFIT98*.

[14] S. Greco, B. Matarazzo, R. Slowinski, and J. Stefanowski, "An algorithm for induction of decision rules consistent with the dominance principle," pp. 507–562, 2001.

[15] S. Kaehler., "Fuzzy logic tutorial - an introduction," 1998. [Online]. Available: http://www.seattlerobotics.org/encoder/Mar98/fuz/index.html

[16] T. J. Ross, *Fuzzy Logic with Engineering Applications.* Wiley, 2004, vol. Second edition.

[17] "Fuzzy logic library for microsoft .net." [Online]. Available: Availablefromhttp://sourceforge.net/projects/fuzzynet

[18] J. Baszczynski, S. Greco, and R. Slowinski, "Multi-criteria classification - a new scheme for application of dominance-based decision rules," vol. 181, no. 3, pp. 1030–1044, 2007.

[19] E. Cayeux and B. Daireaux, "Early detection of drilling conditions deterioration using real-time calibration of computer models: Field example from north sea drilling operations," 2009.

[20] "Neural network toolbox - ver 6.0.4," 2010. [Online]. Available: http://www.mathworks.com/products/neuralnet

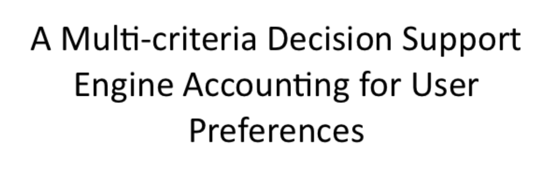
[21] Mathworks, "Crab classification," 2010. [Online]. Available: http://www.mathworks.com/products/neuralnet/demos.html?file=/products/demos/shipping/nnet/crabclassify.html

[22] T. F. Burks, S. A. Shearer, J. Heath, and K. Donohue, "Evaluation of neural-network classifiers for weed species discrimination," vol. 91, no. 3, pp. 293–304, 2005.

[23] K. Priddy and P. Keller, *Artificial Neural Networks: An Introduction.* SPIE Society of Photo-Optical Instrumentation Engineering, 2005.

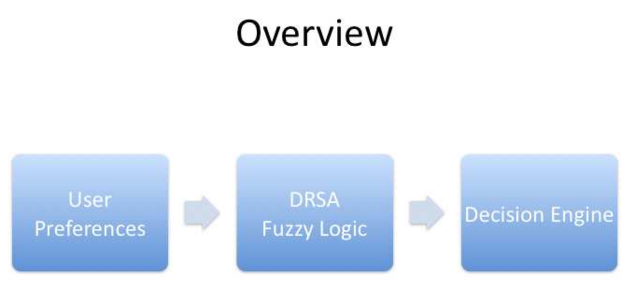
Appendix A - Presentation given at IRIS

This presentation was given at the International Research Institute of Stavanger beforhand of the work with this thesis.



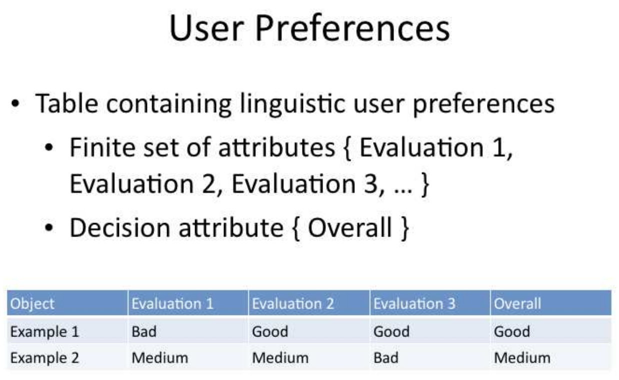
A Multi-criteria Decision Support Engine Accounting for User Preferences

Figure 18: IRIS presentation slide 1



Overview

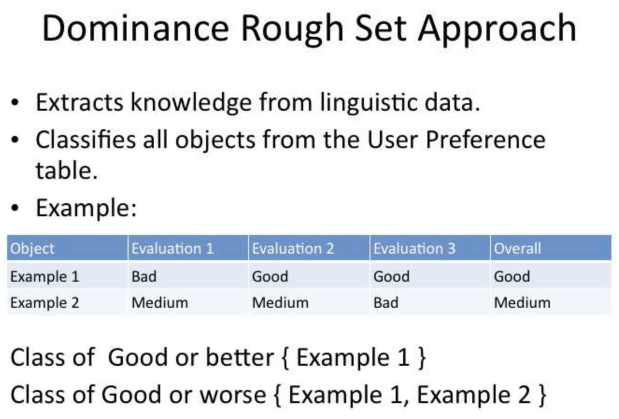
Figure 19: IRIS presentation slide 2


User Preferences

- Table containing linguistic user preferences
  - Finite set of attributes { Evaluation 1, Evaluation 2, Evaluation 3, … }
  - Decision attribute { Overall }

Object	Evaluation 1	Evaluation 2	Evaluation 3	Overall
Example 1	Bad	Good	Good	Good
Example 2	Medium	Medium	Bad	Medium

Figure 20: IRIS presentation slide 3

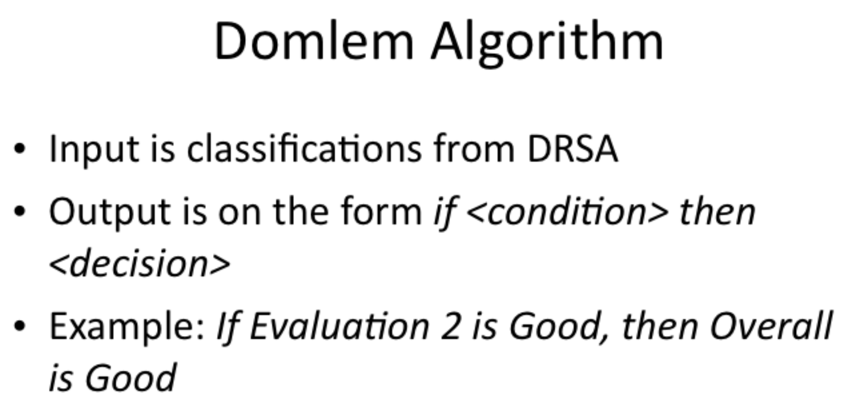

Dominance Rough Set Approach

- Extracts knowledge from linguistic data.
- Classifies all objects from the User Preference table.
- Example:

Object	Evaluation 1	Evaluation 2	Evaluation 3	Overall
Example 1	Bad	Good	Good	Good
Example 2	Medium	Medium	Bad	Medium

Class of Good or better { Example 1 }
Class of Good or worse { Example 1, Example 2 }

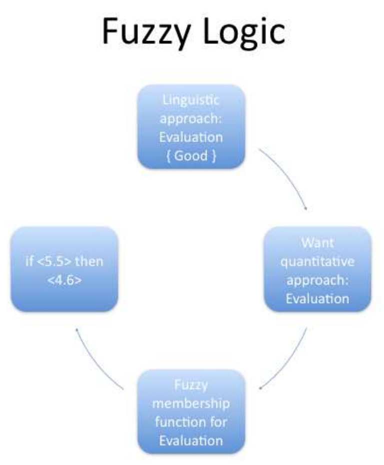
Figure 21: IRIS presentation slide 4



Domlem Algorithm

- Input is classifications from DRSA
- Output is on the form *if <condition> then <decision>*
- Example: *If Evaluation 2 is Good, then Overall is Good*

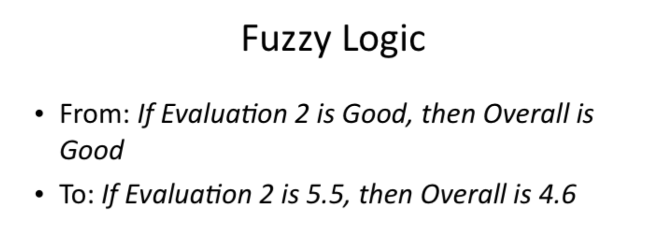
Figure 22: IRIS presentation slide 5



Fuzzy Logic

Linguistic approach: Evaluation { Good }
Want quantitative approach: Evaluation
Fuzzy membership function for Evaluation
if <5.5> then <4.6>

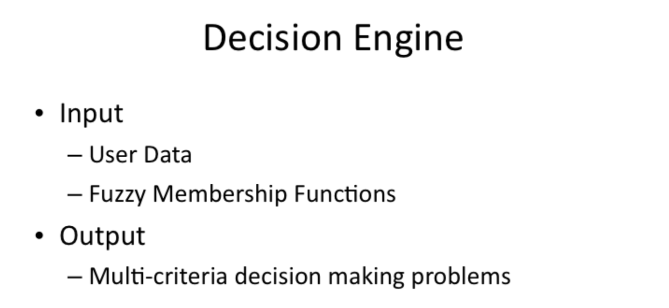
Figure 23: IRIS presentation slide 6



Fuzzy Logic

- From: *If Evaluation 2 is Good, then Overall is Good*
- To: *If Evaluation 2 is 5.5, then Overall is 4.6*

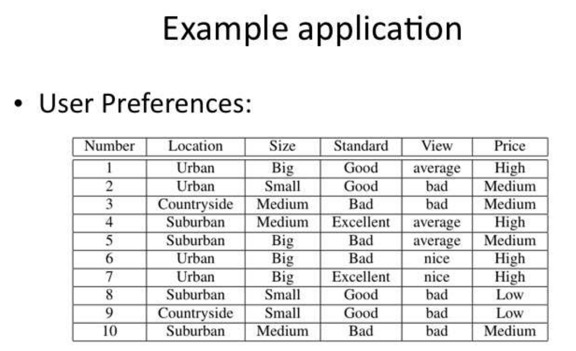
Figure 24: IRIS presentation slide 7



Decision Engine

- Input
  - User Data
  - Fuzzy Membership Functions
- Output
  - Multi-criteria decision making problems

Figure 25: IRIS presentation slide 8

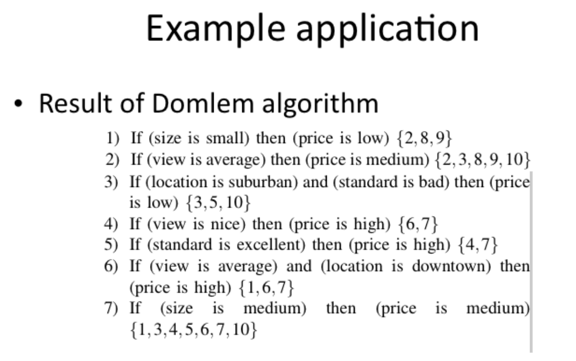


Example application

- User Preferences:

Number	Location	Size	Standard	View	Price
1	Urban	Big	Good	average	High
2	Urban	Small	Good	bad	Medium
3	Countryside	Medium	Bad	bad	Medium
4	Suburban	Medium	Excellent	average	High
5	Suburban	Big	Bad	average	Medium
6	Urban	Big	Bad	nice	High
7	Urban	Big	Excellent	nice	High
8	Suburban	Small	Good	bad	Low
9	Countryside	Small	Good	bad	Low
10	Suburban	Medium	Bad	bad	Medium

Figure 26: IRIS presentation slide 9



Example application

- Result of Domlem algorithm

1) If (size is small) then (price is low) $\{2,8,9\}$
2) If (view is average) then (price is medium) $\{2,3,8,9,10\}$
3) If (location is suburban) and (standard is bad) then (price is low) $\{3,5,10\}$
4) If (view is nice) then (price is high) $\{6,7\}$
5) If (standard is excellent) then (price is high) $\{4,7\}$
6) If (view is average) and (location is downtown) then (price is high) $\{1,6,7\}$
7) If (size is medium) then (price is medium) $\{1,3,4,5,6,7,10\}$

Figure 27: IRIS presentation slide 10

Example application

- Define fuzzy membership functions for linguistic variables
- Example *Location, in distance from City Centre*:

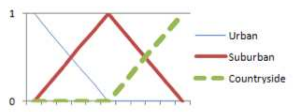
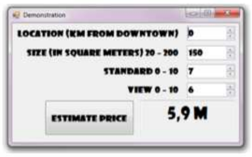


Figure 28: IRIS presentation slide 11

Example application

- Application user interface



LOCATION (KM FROM DOWNTOWN) 0
SIZE (IN SQUARE METERS) 20 - 200 150
STANDARD 0 - 10 7
VIEW 0 - 10 6
ESTIMATE PRICE 5,9 M

Figure 29: IRIS presentation slide 12

Appendix B - Induced decision rules from section 5.3

1. If Hook load >= 102079 and Surface torque < 50,2218 and Top of string Velocity < 100,01 then Friction is NA
2. If Flow rate < 0,0565267 and Hook load < 104169 and Top of string Velocity >= 99,9899 then Friction is NA
3. If Flow rate >= 0,0496416 and Hook load >= 87828,8 and Hook load <= 114529 and Top of string Velocity < 100,01 then Friction is NA
4. If Bit depth >= 2627,94 and Flow rate < 0,00496823 and Hook load >= 104216 and Hook load <= 111187 and SPP >= 184640 and Surface torque < 12245,8 then Friction is NA
5. If Hook load >= 100363 and Surface torque >= 52,0395 and Top of string Velocity < 100,01 and Surface RPM < 0,212717 then Friction is NA
6. If Bit depth >= 3459,92 and Hook load >= 98440,9 and Hook load <= 112910 and SPP < 291090 then Friction is NA
7. If Bit depth >= 3933,02 and Top of string Velocity >= 100,017 and Hook load <= 100,019 then Friction is NA
8. If Flow rate >= 0,0502192 and Flow rate <= 0,0502215 then Friction is NA
9. If Hook load >= 102079 and Hook load <= 113859 and Surface torque < 27,2778 then Friction is NA
10. If Hook load >= 112910 and Top of string Velocity >= 100,018 then Friction is Accepted
11. If Flow rate < 0,0497812 and Hook load < 102079 and Surface torque < 24268,9 and Top of string Velocity < 99,9892 then Friction is Accepted
12. If Bit depth < 3874,3 and Hook load >= 111187 and Top of string Velocity >= 100,01 and Top of string Velocity <= 100,035 then Friction is Accepted

13. If Bit depth >= 3114,73 and Hook load < 87828,8 and Top of string Velocity < 99,9899 then Friction is Accepted

14. If Bit depth < 3130,62 and Hook load >= 106218 and SPP < 777460 and Top of string Velocity >= 100,01 then Friction is Accepted

15. If Flow rate >= 0,00694186 and Hook load >= 106218 and Top of string Velocity >= 100,01 then Friction is Accepted

16. If Bit depth < 3459,92 and Hook load >= 111187 and Top of string Velocity >= 100,012 then Friction is Accepted

17. If Hook load >= 119120 and Top of string Velocity >= 100,012 then Friction is Accepted

18. If Bit depth < 2773,95 and Flow rate >= 0,0565267 then Friction is Accepted

19. If Bit depth >= 2631,82 and Flow rate < 0,0499071 and Surface torque >= 42,8162 and Surface RPM < 0,000903488 then Friction is Accepted

20. If Flow rate >= 0,0591413 then Friction is Accepted

21. If Bit depth >= 2771,54 and Bit depth <= 2773,95 then Friction is Accepted