Electronic Theses and Dissertations

2014

Recommendation Framework Based on Subjective Logic in Decision Support Systems

Arshdeep Singh Sidhu
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommendation Framework based on Subjective Logic in Decision Support Systems

by

Arshdeep Singh Sidhu

A Thesis

Submitted to the Faculty of Graduate Studies

through the School of Computer Science

in Partial Fulfillment of the Requirements for

the Degree of Master of Science at the

University of Windsor

Windsor, ON, Canada

2014

Recommendation Framework based on Subjective Logic in Decision Support Systems

by

Arshdeep Singh Sidhu

APPROVED BY:

_____

A. Hussein

Department of Mathematics and Statistics

_____

Z. Kobti

School of Computer Science

_____

R. D. Kent, Advisor

School of Computer Science

April 24, 2014

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

In this thesis our goals are to investigate the suitability of subjective logic within the decision support context that requires connectivity to complex data, user specification of frames of discernment, representation of complex reasoning expressions, an architecture that supports distributed usage of a decision support tool based on a client-server approach that separates user interactions on the browser side from computational engines for calculations on the server side, and analysis of the suitability and limitations of the proposed architecture.

The nature of our framework represents a proof-of-concept approach in that we have limited ourselves to the scope of binomial and multinomial opinions only, foregoing recent work on emerging hyper-nomial opinions, and also on a limited subset of operators, due in part to ongoing work that is moving towards establishing generally agreed upon definitions and properties of all operators.

DEDICATION

To my loving parents and grandparents whose immense

patience and faith in me have got me this far in my life.

ACKNOWLEDGEMENTS

My ultimate gratitude goes to my supervisor Dr. Robert D. Kent, for providing me the opportunity to work in an exciting and challenging field of research. His constant motivations, support, innovative ideas, own research and infectious enthusiasm have guided me toward successful completion of my thesis. My interactions with him have been of immense help in defining my research goals and in identifying ways to achieve them.

I would like to acknowledge Dr. Ziad Kobti and Dr. Abdulkadir Hussein whose suggestions and recommendations greatly improved the quality of this work. I would also like to thank Bryan St. Amour for his advice and help in learning new things. I am thankful to my good friends Sukhmanjot Singh and Manpreet Singh for their valuable and helpful ideas in this work. And I want to thank my best friends Harmeetpal, Bhupinder and Sandeep for always motivating me in tough times.

My special thanks to my grandparents and parents for their patience and love they provided to me during all times.

TABLE OF CONTENTS

CHAPTER

APPENDICES

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A fundamental aspect of the human condition is that nobody can ever determine with absolute certainty whether a proposition about the world is true or false, or determine the probability of something with 100% certainty. [1] When subjective logic is used for decision support, it allows decision makers to be better informed about uncertainties affecting the assessment of specific situations and future outcomes. [1]

The idea of subjective logic is to extend probabilistic logic by also expressing uncertainty about the probability values themselves, meaning that it is possible to reason with argument models in presence of uncertain or incomplete evidence. Subjective logic is directly compatible with binary logic, probability calculus and classical probabilistic logic. The advantage of using subjective logic is that real world situations can be more realistically modeled, and that conclusions more correctly reflect the ignorance and uncertainties that necessarily result from partially uncertain input arguments. [1] It can for example be used for modeling trust networks, for modeling Bayesian networks, for Intelligence Analysis and logical argumentation. In general, subjective logic is suitable for modeling and analyzing situations involving uncertainty, incomplete knowledge and different world views. [1]

Following Jøsang, subjective logic provides a suitable framework for connecting survey data collection directly to a model of evidence based opinions with uncertainty that also support subjective reasoning. [2].

Subjective logic [3] is a type of probabilistic logic [4], [5] where connectives are defined by mathematical expressions instead of look-up truth tables. Subjective logic explicitly takes uncertainty and belief ownership into account, and is suitable for modeling and analyzing situations involving uncertainty and incomplete knowledge. Arguments in subjective logic are opinions about propositions.

With this in mind, we have implemented a very general methodology for decision support systems that provide recommendations. Our starting point still involves the Human Expert as a significant oracular element within the system, but as research continues, one senses how the vision of computationally driven, intelligent support for complex human and machine system activities may evolve. A recommendation system is implemented which is able to populate a set of belief values from datasets, in order to build a model of a complex subjective logic assertion.

1.1 Problem Statement

Since the pioneering work on evidentiary reasoning with uncertainty by Dempster and Shafer (Shafer 1976; 1990) there have been attempts to develop consistent reasoning frameworks of logic and interpretation of belief and uncertainty in the context of evidence. The inclusion of uncertainty was intended to provide a method for dealing with evidence subjectively. Substantial progress towards such a subjective logic framework has been made by Jøsang and co-workers (Jøsang 1997, 2001, 2002, 2007, 2008; Jøsang and McAnally 2004; Jøsang, et al 2005; Jøsang, et al 2006; Jøsang, et al 2010; McAnally and Jøsang 2004; Pope and Jøsang 2005). [6]

Although this evolving framework provides accurate numeric results based on subjective logic (SL) calculations and interpretations, they do have some limitations. The major limitation is a lack of implementation in software, or reasoning systems. Existing web browser applets provided by Jøsang et al., provide a limited approach of solving operator expressions involving two opinion arguments using single SL operators at a time. The applets allow belief values to be entered manually; but, they do not offer mechanisms to build complex subjective logic expressions and solve them, particularly in cases where real data sets are of interest. Similarly, there has been no software framework developed for multinomial opinions where a user can build multiple opinions.

The application of subjective logic to actual data is of considerable interest. There has been no previous work which populates belief values directly from datasets. Most of the previous work limits the user to work with two opinions at a time, and only to perform calculations. Existing proposed applets provide a limited approach of solving two opinion arguments using single SL operators. In existing applets belief values are entered manually and they do not offer mechanism to build complex expressions and solve them. There has been no framework for multinomial opinions where user can build multiple opinions.

For decision support systems, the need to join data to subjective logic to support reasoning, especially dynamic exploration of data, is vital. Simply stated, the problem identified for this thesis is one of designing an architecture for a decision support system that embeds subjective logic to support reasoning over data. This simple statement contains several implications for software and system design and verification that will be addressed in more detail in the following sections.

1.2 Thesis Objective

The aim of this thesis is to outline the objective decision support system elements in a workbench based on subjective logic. Our main objective is to develop a web-based representative and reasoning framework based on Subjective Logic in decision support systems, which consists of a belief model called opinion and set of operations for combining opinions.

In this thesis research, our practical goal is to construct a standard data acquisition interface based on subjective logic which proves to be productivity enhancing tool in decision support system where uncertainty is essential part of decision, while also serving as foundation platform for future research. Our goal is to build a subjective logic workbench which has capabilities of rendering opinion values from datasets, solve simple and complex subjective logic expressions for binomial and multinomial opinions, and provide results based on the dataset used. In accordance to this our goal is to develop an algorithm to solve subjective logic expressions built using subjective logic operators. We need a suitable workbench which connects survey data collection directly to a model of evidence based opinions with uncertainty.

1.3 Thesis Contribution

In this thesis we aim to design and implement a workbench based on subjective logic, which enables the user to build opinions, render belief values, construct simple and complex subjective logic expressions, using subjective logic operators to calculate the degree of uncertainty associated with a hypothesis. Subjective logic can be used to model real world situations and the conclusions reflect the ignorance and uncertainties. In this way we use Subjective logic for our recommendation framework so that decision makers

to be better informed about uncertainties affecting the assessment of specific situations and future outcomes.

The workbench is able to populate a set of belief values by direct query to datasets, in order to build a model of a complex subjective logic assertion. Main goals of this thesis are:-

- Develop an interface which allows user to build "n" number of opinions.
- Display data sets to user and allow user to render belief values as per hypothesis.
- Develop a mechanism to build SL expressions for binomial and multinomial opinions.
- Develop an algorithm to solve subjective logic expressions.
- Decision support: Enable user to construct and interactively investigate hypothesis arguments utilizing SL operators.
- To allow users to define their own frame of discernment.

1.4 Thesis Outline

The aim of this study is to outline the decision support elements in recommendation framework, specifically in context of complex subjective logic expressions, design, and implementation of user interface (UI) based on subjective logic, automated functioning of subjective logic operators, critically examine the influence of the factors that contribute to certain decisions in decision support systems (DSS). In order to discuss this we divide the thesis into following chapters.

In Chapter 2, a literature review and survey is presented on decision support systems (DSS) in the domain of computer science, survey on uncertainty, probability

theory, Bayesian networks, Dempster-Shafer theory and related work. Then we discuss, in detail about subjective logic.

More specifically, we first describe the representations and interpretations of subjective opinions which are the input arguments to subjective logic. We then describe the most important subjective logic operators. Finally, we describe how subjective logic can be applied in decision support systems.

Chapter 3 describes brief overview of workbench by discussing the architecture of the workbench and several components involved in the architecture design and describe the implemented algorithm for handling simple and complex subjective logic expressions.

Chapter 4 presents implementation and verification of workbench and also about usability of system.

Chapter 5 concludes the thesis and proposes some avenues of future work in subjective logic workbench.

CHAPTER II

PRELIMINARIES AND DEFINITIONS

2.1 Decision Support Systems

Decision support systems (DSS) are interactive computer-based systems that help decision makers solve decision problems. They attempt to do this by formalizing knowledge so that it is amenable to mechanical reasoning. DSS can be categorized under knowledge-based systems. One class of DSSs, expert systems, originates from the field of artificial intelligence, and aims at imitating the reasoning of a human domain expert in solving decision problems. DSSs can also be built on formal techniques, such as the methods of operations research, or decision theory.

Decision support systems are gaining an increased popularity in various domains, including engineering, business, and medicine. Although their reasoning power is still rather limited, they can sometimes approach the abilities of human experts and outperform practitioners in some domains [2, 3, 4]. DSSs are valuable in situations where the amount of relevant information that needs to be considered is prohibitive for the intuition of an unaided human decision maker. Such environments are often given the common name of DSSs. A Decision Support System is a class of information systems that supports business and organizational decision making activities. DSS couple the intellectual resources of individuals with the capabilities of the computer to improve the quality of decisions [7]. It is a computer-based support for management decision makers, those who deal with semi-structured problems. A properly designed DSS is an interactive software-based system, intended to help the decision makers compile useful information

from a combination of raw data, documents, personal knowledge, or business models to identify and solve problems and make decisions. Decision Support System is a general term for any computer application that enhances a person's or a group's ability to make decisions. It can also be used as a tool in which user inputs the data and the software component process the data and recommendations are made on the basis of the information given. In order to make the decision making tool, all the major components of the system should be considered in the system to get the optimal results.

2.1.1 DSS Architecture and Categories

The architecture is comprised of four main subsystems: language system, presentation system, knowledge system, and problem-processing system. These determine its capabilities and behaviors (Bonczek et al. 1980, 1981a, Dos Santos and Holsapple 1989, Holsapple and Whinston 1996). By varying the makeup of these four elements, different types of decision support systems are produced.

A language system consists of all messages the DSS can accept. A presentation system consists of all messages the DSS can emit. A knowledge system consists of all knowledge the DSS has stored and retained. By themselves, these three kinds of systems can do nothing, neither individually or in tandem. They are inanimate. They simply represent knowledge, either in the sense of messages that can be passed or representations that have been accumulated for possible future processing. Although they are merely systems of representation, the KS, LS, and PS are essential elements of a DSS. Each is used by the fourth element: the problem processing system.
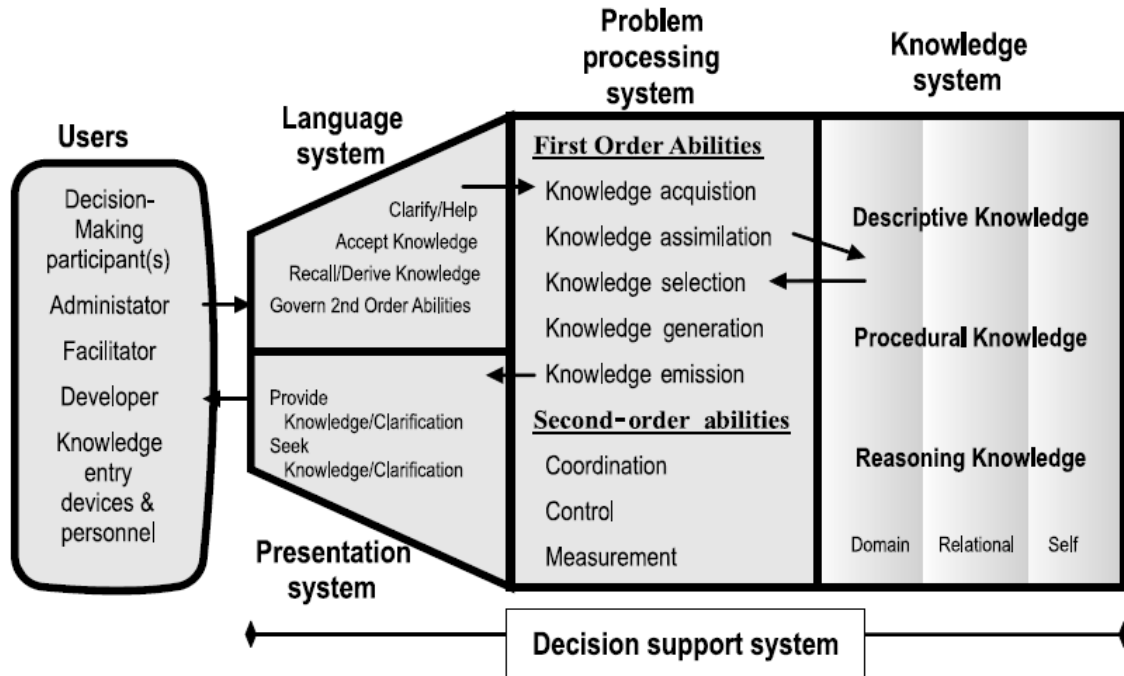
Figure 2.1: Basic architecture for decision support systems [32]

This system is the active component of a DSS. A problem processing system is the DSS's software engine. As its name suggests, a PPS is what tries to recognize and solve problems (i.e., process problems) during the making of a decision. Figure 2.1 illustrates how the four subsystems of a DSS are related to each other and to a DSS user. The user is typically a decision maker or a participant in decision making.

2.1.2 Web-Based Decision Support

Web-based decision support systems (WB-DSS) are decision support systems that are accessible on the Web. They have the same broad boundaries as those of desktop DSSs. Nevertheless, WB-DSS can be identified by certain characteristics:

1. Accessible on the Web

2. Supporting individuals/customers/employees/managers/groups in their decision-making process regardless of their physical locations or time of access

3. Having outcomes that are specific to a predetermined context that is either unique to the Web environment or as the interface for desktop DSS

4. Dealing with decision processes that are semi-structured or unstructured at different stages of the decision process, some of which could take place on the Web

5. Utilizing data, knowledge base, document, model and heuristics, which appeal to a culturally varied and large user group

6. Being an optional tool for Web users in their decision processes.

2.1.3 Benefits of Decision Support Systems

It is important to identify the benefits of a decision support system (DSS). Systems that are implemented without understanding the prospective benefits for a particular context will not achieve their full potential in contributing to organizational performance. After implementation, it is important that the benefits be apparent, or the system will fall into disuse because DSS use is typically optional. Furthermore, a record of producing DSSs with benefits that can be identified, elaborated, and quantified creates more opportunities for those who created and implemented the systems. It also contributes to an organization's learning about how to plan for and realize future DSS success.

Decision support systems provide benefits when the combination of the system plus a decision maker (or makers) is superior to the performance of software or humans alone. Often, combining the best attributes of fast computation, large disk storage, graphic displays, and intelligent software with the insights of human decision makers will

achieve excellent decision quality or an excellent decision making process. Generally, the benefit of a DSS is better decisions, a better decision- making process, or both. Figure 2.2 illustrates this idea.
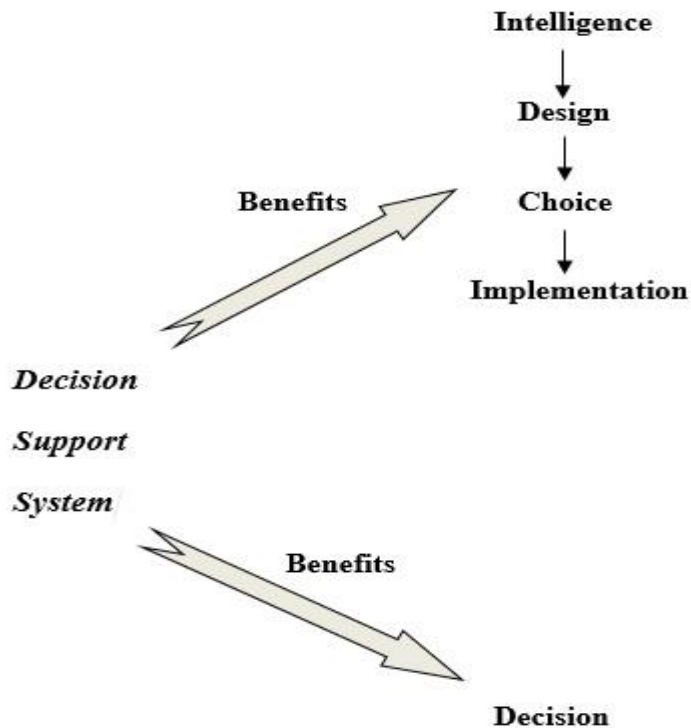
Figure 2.2: Decision support system benefits via improvements to decision-making processes or outcomes [32]

2.2 Uncertainty

There are many systems that are designed and developed based on precision and certainty. They provide unrealizable solutions based on the assumption of closed environments. For the most real applications uncertainty is inevitable and cannot be ignored.

2.2.1 Uncertainty Categories

Information imperfection is the most difficult, but unavoidable problem faced by agents in an open environment. According to Smets approach [24] it can be generally grouped into imprecision, inconsistency or uncertainty.

1) Imprecision presents the ambiguity, vagueness or approximation of information.

2) Inconsistency expresses that contradictory conclusions can be drawn based on given information or statements.

3) Uncertainty is caused by a lack of knowledge about the environment when agents need to decide the truth of statements. Uncertainty can be distinguished objectively and subjectively. Objective uncertainty relates to randomness which likely qualifies the occurrence possibility of an event, whereas subjective uncertainty depends on the subjective opinions of agents about the truth value of information. Imprecision and inconsistency are essential properties related to information content whereas uncertainty is a property of the relation between the information and our knowledge about the world. Besides the classification based on Smets approach, another viewpoint describing perspectives on computational perception and cognition under uncertainty, is proposed by Zadeh [25]. Two broad categories of uncertainty, U-Type One and U-Type Two, are suggested:

- The first type of uncertainty deals with information arising from the random behaviour of physical systems.

- The second type of uncertainty deals with information arising from human perception and cognition processes.

The first type has been investigated for centuries with efforts of statistical theory. The statistical methodologies are very useful to model this type, but lack the sophistication to process the second type. In order to deal with the second type, several effective methods have been proposed, including fuzzy logic, neural networks and so on.

2.3 Probability Theory

2.3.1 Probabilistic Reasoning in Decision Support Systems

Uncertainty is an inherent and prevalent property of most types of knowledge. It arises from sources like incomplete knowledge, disagreement between various information sources, linguistic imprecision, statistical variation in the measured population, measurement error, or approximations. Arguably all practical decisions involve uncertainty. We might cope with uncertainty simply by worrying about it or pretending it is not there, but there are situations in which we would like to estimate, reduce, and, if this is not feasible, take it into account when making the decision of the calculi developed for dealing with uncertainty, the oldest and most widely used is probability theory. Uncertainty in probability theory is measured by a real number between 0:0 (impossible event) and 1:0 (sure event), called probability.

2.3.2 Interpretations of Probability

There are several interpretations as to what probability means. These can be roughly divided into three classes: the frequency interpretation, the propensity interpretation, and the subjectivist interpretation. In the frequency interpretation, the probability of an outcome is given the meaning of the relative proportion with which that outcome would be obtained if the process were repeated a large number of times under similarconditions. The probability of "heads" in a coin toss can be empirically verified by tossing the coin a

large number of times and counting the proportion of times that the coin fell "heads" with respect to the total number of tosses. In the propensity interpretation, the probability is thought of as a property of the physical system that generates the events. A coin has two sides and because of symmetry considerations, these can be assumed to be equally likely, and therefore the probability of "heads" is equal to the probability of "tails" and, hence, has to be equal to 0.5 (a biased coin will have the propensity to fall "heads" with a different probability, but this probability will be again a property of the coin).

The frequency and the propensity views are often called *objectivist*, because they interpret probability as an objective property of the world. In the subjectivist view, often called *personalist, personal*, or *Bayesian* view, probability of an outcome is a measure of personal degree of belief in that outcome, given the person's current state of knowledge. A person with no special information about the coin or the way in which it is tossed might regard both "heads" and "tails" equally likely, but he or she might equally well give it a different assignment given the previous experiences with other coins, other relevant information. The person canalso change this assignment in the course of observations.

2.3.3 Subjectivist Bayesian Approach

The probability of a proposition in the subjectivist Bayesian view is a measure of Personal belief in that proposition. As two different people may have different information relevant to the event, they can have legitimately different measures of belief in that event. Effectively, there is no measure that can be termed as probability. Bayesian view of probability theory includes methods for eliciting and evaluating accuracy of judgments. As there are doubts whether people have clear intuitions about their

probabilities, proponents of the Bayesian view advocate indirect measurement in which a person is observed making choice between bets [30]. A person is offered choice between gambles involving the proposition in question and the choices made between these gambles are used to estimate the measure of belief that the person has in the proposition. A fundamental principle of Bayesian reasoning is belief updating, which means starting with an initial belief in a proposition and changing this belief as new evidence accumulates. The initial belief is called the prior probability and the belief that results from taking evidence into consideration is called the posterior probability. As evidence can be processed stepwise, the posterior probability obtained in one step can be used as the prior probability in the next step. The fundamental rule used inbelief updating is Bayes theorem. The simplest form of the Bayes theorem is:

$$Pr\,(H/E) = \frac{Pr(H \cap E)}{Pr(E)} = \frac{Pr(E|H)Pr(H)}{\sum H_i \in \Omega Pr(E|H_i)Pr(H_i)} \tag{2.1}$$

Bayes theorem provides a rule for updating belief in a hypothesis H given evidence E. $Pr(H)$ on the right hand side of the equation is the prior probability of the hypothesis H, while $Pr(H|E)$ on the left hand side is its posterior probability. $Pr(E|H)$ and $Pr(E)$ are measures that jointly express the value of the evidence E for the hypothesis H. One of the ways to obtain $Pr(E)$ is summing its probability over all possible hypotheses. [1]

2.3.4 Decision Theory and Decision Analysis

Bayesian probability theory forms the foundation of a theory of decision making, usually known as decision theory. While probability theory provides formalism for treatment of uncertainty, decision theory extends it with a set of principles for

consistency among preferences and decisions. Preferences describe relative valuations of outcomes, while decisions are actions that are under decision maker's control. Applied branch of decision theory, known as decision analysis [28] has been developed as a normative aid to human cognitive deficiencies in decision making. Decision analysis is based on the paradigm that people are able to reliably store and retrieve their personal beliefs about uncertainty and preferences for different outcomes, but are much less reliable in aggregating these fragments into a global inference. Decision analysis includes quantities of methods for model construction, such as methods for elicitation of probability distribution that allow to minimize human bias, methods for checking the sensitivity of a model to imprecision in the data, etc. [28,29]. It should be pointed out that decision theory does not address the first and arguably the most important step of any decision-making process, notably framing of the decision problem and generation of the decision alternatives. Although modern textbooks for decision analysis provide numerous advices and heuristics that aid this stage, framing a decision problem is essentially an art, requiring much creativity on the part of decision analysts.

2.4 Dempster-Shafer theory

Following is a brief description of elements of Dempster-Shafer theory. The theory is a system for qualifying one's beliefs using numerical expressions of degrees of support. Shafer (1976) provides a fuller theoretical treatment for the interested reader. Shafer described several, inter-related measures, conveying slightly different messages about evidential weight, and the transformation functions connecting them. One of these, 'Bel' is termed a belief function and is a commonly employed measure from the system. Here, a different measure is elicited, the basic probability assignment, or what we shall

call the reserve function. Both measures capture a degree of belief. The two measures have a 1–1 correspondence and are mathematically inter-transformable, so the selection for assessment is a matter of experimenter preference. The reserve function measure is chosen here as being most conceptually like probabilities. Both probabilities and reserve functions can be characterized as dividing the whole of one's belief (1.0) into smaller elements. Consequently, the measure is believed to be an intuitive one for individuals to assess. Which of the two measures might be better for assessment is an open empirical question that is not addressed here. We do argue that the assessments obtained in this study are meaningful and informative. For brevity of exposition, hereafter belief is used interchangeably with "degree of belief." Other terminology from the theory that is used in this work includes:

1.   **Frame of Discernment**: A finite set of possible values for a variable X, such that one, and only one, element of the set are true. These elements are the possible states of nature or hypotheses. In general, the items within the frame of discernment develop as evidence accumulates i.e., one can assign belief to ɵ without specifying what elements might be contained within it. However, in this study for experimental control, the elements in the frame are given to subjects, ɵ = {a, b, c, d, e, f, and g}.

2.   **Dempster's Rule**: A method for combining two independent functions, m_1 and m_2, into a new function,

(a) **Conflict**

Evidence 1:
Target set {d}, Strong (.7)

Evidence 2:
Target set {g}, Moderate (.4)
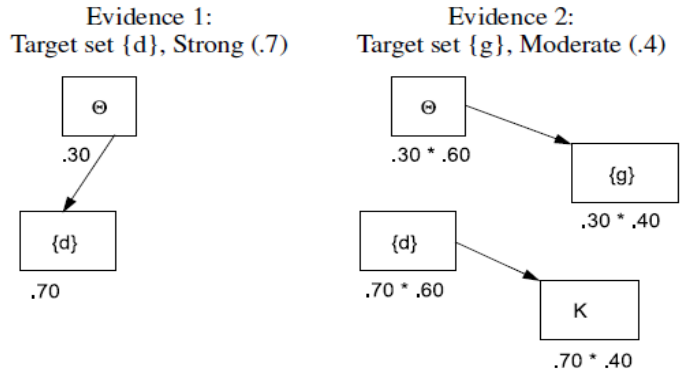
(b) Dempster's Rule
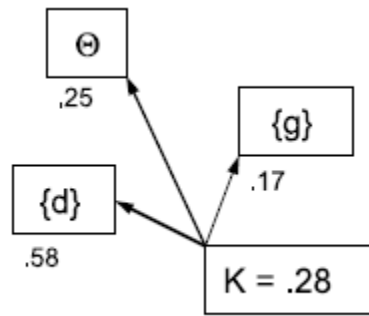
Figure 2.3 Movement of belief where evidence creates conflict (K>0) in (a), (b)

$$m\ (A) = (1 - K)^{-1}\ P\sum m_1(A_i)m_2(A_j),$$

for all $A_i \subseteq \theta$, $A_j \subseteq \theta$

Where $A_i \cap A_j = A$; and

$$K = \sum m_1\ (A_i)\ m2\ (A_j),$$

for all $A_i \cap A_j = \emptyset;$

18

The parameter K is a measure of *conflict* in the evidence. The idea behind the combination rule is that initially your belief is undifferentiated and allocated to Θ. As evidence becomes available, you partition your belief into smaller subsets. Although shown successively, Dempster's Rule is commutative; the order of evidence is irrelevant. Initially, there is no evidence and all support (1.0) is in the undifferentiated set Θ. As shown, the first piece of evidence implicates a and d, not differentiating between them. The function m_1 moves a portion of the weight of evidence into the set {a, d} to convey this, leaving the remainder of the weight in the set Θ. How much weight is moved depends on the reliability, credibility and strength of the evidence. The second piece of evidence implicates a, b and c. The function m2 moves a portion of the weight from Θ into {a, b, c} and moves the same proportion of the weight from {a, d} to the intersection of the two sets: {a}, in this way, as evidence accumulates, support becomes differentiated into finer subsets capturing the justification for the possible evidential conclusions.

2.5 Subjective Logic

Since the pioneering work on evidentiary reasoning with uncertainty by Dempster and Shafer (Shafer 1976; 1990) there have been attempts to develop consistent frameworks of logic and interpretation of belief and uncertainty in the context of evidence. Substantial progress towards such a subjective logic framework has been made by Jøsang and co-workers (Jøsang 1997, 2001, 2002, 2007, 2008; Jøsang and McAnally 2004; Jøsang, et al 2005; Jøsang, et al 2006; Jøsang, et al 2010; McAnally and Jøsang 2004; Pope and Jøsang 2005) [6]. The idea of subjective logic is to extend probabilistic logic by also expressing uncertainty about the probability values themselves, meaning

that it is possible to reason with argument models in presence of uncertain or incomplete evidence. Subjective logic is directly compatible with binary logic, probability calculus and classical probabilistic logic [7].

It is nearly impossible to determine with absolute certainty about the truthfulness or falseness about a proposition in the world, or to determine the probability of something with 100% certainty. Important aspects are missing in the way standard logic and probabilistic logic capture our perception of reality and that these reasoning models are more designed for an idealized world than for the subjective world in which we are all living. A limitation of probabilistic logic, and binary logic alike, is that it is impossible to express ignorance in the input arguments as e.g. reflected by the expression "I don't know". An analyst who does not have a reliable value for a given input argument can be tempted or even forced to set a value without any evidence to support it. This practice will generally lead to unreliable conclusions, often described as the "garbage in – garbage out" problem [7]. Arguments in subjective logic are called "subjective opinions" or "opinions" for short. An opinion can contain degrees of uncertainty in the sense of "uncertainty about "probability estimates". The uncertainty of an opinion can be interpreted as ignorance about the truth of the relevant states, or as second order probability about the first order probabilities [7]. The advantage of subjective logic over traditional probability calculus and probabilistic logic is that real world situations can be modeled and analyzed more realistically. The analyst's partial ignorance and lack of information can be taken explicitly into account during the analysis, and explicitly expressed in the conclusion. When used for decision support, subjective logic allows

decision makers to be better informed about uncertainties affecting the assessment of specific situations and future outcomes.

2.5.1 Belief Representations in Subjective logic [7]

Explicit expression of uncertainty is one of the main characteristics of subjective logic. Uncertainty comes in many flavors, and a good taxonomy is described in [8]. It describes four different syntactic representations of beliefs that can be applied in subjective logic. Although quite different in notation, these representations are mathematically and semantically equivalent. The subjective opinion notation is the classical and original representation used in subjective logic. Subjective opinions can be visualized in the form of opinion triangles and opinion simplexes which can aid human interpretation. The subjective opinion representation forms the basis for the subjective logic operators, and the other representations are useful to better understand the correspondence between subjective logic and other mathematical formalisms, for solicitation of beliefs. The evidence representation, which is the second type, provides a classical mathematical representation often used in statistics which can also give useful and intuitive visualisations in the form of probability density functions. The evidence representation also provides the most intuitive way of including new evidence an observation into opinions. The probabilistic representation, which is the third type, might seem simple because it explicitly contains the probability expectation value.

This representation provides the most direct correspondence with probability calculus, but it does not seem to facilitate any particularly intuitive visualisations of uncertain probabilities. The fuzzy category representation is the fourth type and provides

a way of expressing opinions in terms of common verbal expressions such as *"unlikely"* or "very likely".

2.5.2 Elements of Subjective Opinions

An opinion is a composite function consisting of belief masses, uncertainty mass and base rates which are described separately below. An opinion applies to a frame, also called a state space, and can have an attribute that identifies the belief owner. The belief masses are distributed over the frame or over the reduced power set of the frame in a sub-additive fashion, meaning that the sum of belief masses normally is less than one. An important property of opinions is that they are equivalent Beta or Dirichlet probability density functions (pdf) under a specific mapping.

The Reduced Power set of Frames [7]

Let X is a frame of cardinality k. The power set of X, denoted as P(X) equivalently as 2^X, has cardinality 2^k and contains all the subsets of X, including X and $\emptyset$. In subjective logic, the belief mass is distributed over the reduced power set denoted as R(X). More precisely, the reduced power set R(X) is defined as:

$$R(X) = 2X \setminus \{X, \emptyset\} = \{xi \mid i = 1 \ldots k, xi \subset X\} \qquad (2.2)$$

It means that all proper subsets of X are an element of R(X), but X itself is not in R(X). The empty set $\emptyset$ is also not considered to be a proper element of R(X).Let κ denote the cardinality of R(X), i.e. $\kappa = |R(X)|$. Given the frame cardinality $k = |X|$, then we have $\kappa = (2^k - 2)$, i.e. there are only $(2^k - 2)$ elements in the reduced power set R(X) because it is assumed that X and $\emptyset$ are not elements of R(X). It is practical to define the first k

elements of R(*X*) as having the same index as the corresponding singletons of *X*. The remaining elements of R(*X*) should be indexed in a simple and logical way. The elements of R(*X*) can be grouped in classes according to the number of singletons from *X* that they contain.

Belief Distribution over the Reduced Power set [7]

Subjective logic allows various types of belief mass distributions over a frame *X*. The distribution vector can be additive or sub-additive, and it can be restricted to elements of *X* or it can include proper subsets of *X*. A belief mass on a proper subset of *X* is equivalent to a belief mass on an element of R(*X*). When the belief mass distribution is sub-additive, the sum of belief masses is less than one, and the complement is defined as uncertainty mass. When the belief mass distribution is additive, there is no uncertainty mass. The sub-additivity of the belief vector and the complement property of the uncertainty mass are expressed by

Belief sub-additively: $\sum_{x_i \in R(X)} \vec{b}_X(x_i) \leq 1, \vec{b}_X(x_i) \in [0,1]$ (2.3)

Belief and uncertainty additively: $u_X + \sum_{x_i \in R(X)} \vec{b}_X(x_i) = 1, \vec{b}_X(x_i), u_X \in [0,1]$ (2.4)

Base Rates over Frames [7]

The concept of base rates is central in the theory of probability. Base rates are for example useful for default and for conditional reasoning. Traditional belief theory does not specify base rates. [7] Without base rates however, there are many situations where belief theory does not provide an adequate model for expressing intuitive beliefs. This

section specifies base rates for belief functions and shows how it can be used for probability projections.

Given a frame of cardinality k, the default base rate of for each singleton in the frame is $1/k$, and the default base rate of a subset consisting of n singletons is n/k. In other words, the default base rate of a subset is equal to the number of singletons in the subset relative to the cardinality of the whole frame. A subset also has default relative base rates with respect to every other fully or partly overlapping subset of the frame. However, in practical situations it would be possible and useful to apply base rates that are different from the default base rates. For example, when considering the base rate of a particular infectious disease in a specific population, the frame can be de-fined as { *"infected"*, *"not infected"*}. Assuming that an unknown person enters a medical clinic, the physician would *a* priori be ignorant about whether that person is infected or not before having assessed any evidence. This ignorance should intuitively be expressed as a vacuous belief function, i.e. with the total belief mass assigned to (*"infected"* ∪ *"not infected"*). The probability projection of a vacuous belief function using default base rate of 0.5 would dictate that the *a* priori probability of having the disease is 0.5. Of course, the base rate of diseases is normally much lower, and can be determined by relevant statistics from a given population. The actual base rate can often be accurately estimated, as e.g. in the case of diseases within a population. Typically, data is collected from hospitals, clinics and other sources where people diagnosed with a specific disease are treated. The amount of data that is required to calculate a reliable base rate of the disease will be determined by some departmental guidelines, statistical analysis, and expert opinion about the data that it is truly reflective of the actual number of infections – which is itself a subjective

assessment. After the guidelines, analysis and opinion are all satisfied, the base rate will be determined from the data, and can then be used with medical tests to provide a better indication of the likelihood of specific patients having contracted the disease [9].

Integrating base rates with belief functions provides a basis for a better and more intuitive interpretation of belief functions facilitates probability projections from belief functions and provides a basis for conditional reasoning. The base rate function is a vector denoted as $\vec{a}_X$ so that $\vec{a}_X(x_i)$ represents the base rate of the elements $x_i \in X$.

(**Base Rate Function**) Let X be a frame of cardinality k, and let $\vec{a}_X$ be the function from X to $[0, 1]^k$ satisfying:

$$\vec{a}_X(\emptyset) = 0, \vec{a}_X(x_i)\epsilon\ [0,1]\ \ and\ \ \textstyle\sum_{i=1}^{k} \vec{a}_X(x_i) = 1 \tag{2.5}$$

Then $\vec{a}_X$ is a base rate distribution over X. Two different observers can share the same base rate vectors. However, it is obvious that two different observers can also assign different base rates to the same frame, in addition to assigning different beliefs to the frame. This naturally reflects different views, analyses and interpretations of the same situation by different observers. Base rates can thus be partly objective and partly subjective. Events that can be repeated many times are typically frequent in nature, meaning that the base rates for these often can be derived from statistical observations. For events that can only happen once, the analyst must often extract base rates from subjective intuition or from analyzing the nature of the phenomenon at hand and any other relevant evidence. However, in many cases this can lead to considerable uncertainty about the base rate, and when nothing else is known, the default base rate of the single-tons in a frame should be defined to be equally partitioned between them, following a

25

uniform distribution. More specifically, when there are k singletons in the frame, the default base rate of each element is $1/k$.

2.5.3 Opinion Classes [7]

Subjective opinions express beliefs about the truth of propositions under degrees of uncertainty, and can indicate ownership (of the opinion) whenever required. A subjective opinion is normally denoted as ω_x^A where A is the opinion owner, also called the subject, and X is the target frame to which the opinion applies. An alternative notation is ω (A: X). There can be different classes of opinions, of which hyper opinions are the most general. Multinomial opinions and binomial opinions represent specific sub-classes of general hyper opinions, as will be explained below. In case of binomial opinions, the notation is ω_x^A or alternatively ω (A: x), where x is a single proposition that is assumed to belong to a frame X, but the frame is normally omitted, and only implicitly assumed in the notation for binomial opinions.

The propositions of a frame are normally assumed to be exhaustive and mutually disjoint, and belief owners are assumed to have a common semantic interpretation of propositions. The belief owner (subject) and the propositions (object) are optional attributes of an opinion. The opinion itself is a composite function consisting of the belief vector $\vec{b}_X$, the uncertainty mass u_X and the base rate vector $\vec{a}_X$. More specific opinion classes can be defined, such as DH opinion (Dogmatic Hyper), UB Opinion (Uncertain Binomial) etc. The six main opinion classes defined in this way are listed in Table 2.1 below,

	Binomial **Cardinality |X|=2** **X=R(X)**	**Multinomial** **Cardinality |X|>2** **Focal elements x ∈ X**	**Hyper Cardinality** **|X|>2** **Focal elements x ∈ R(X)**
Uncertain **u > 0**	UB opinion beta pdf	UM opinion Dirichlet pdf over X	UH opinion Dirichlet pdf over R(X)
Dogmatic **u = 0**	DB opinion Scalar probability	DM opinion Probabilities on X	DH opinion Probabilities on R (X)

Table 2.1 Opinion classes with equivalent probabilistic representations [7]

The intuition behind using the term "dogmatic" is that a totally certain opinion (i.e. where u = 0) about a real-world proposition can be seen as an extreme opinion. From a philosophical viewpoint nobody can ever be totally certain about anything in this world, so when it is possible to explicitly express degrees of uncertainty as with opinions, it can be seen as arrogant and extreme when somebody explicitly expresses a dogmatic opinion. This interpretation is confirmed when considering that a dogmatic opinion has an equivalent probability density function in the form of a singularity requiring an infinite amount of evidence. This does not mean that traditional probabilities should be interpreted as dogmatic, because their representation does not allow uncertainty to be expressed explicitly. Instead it can implicitly be assumed that there is some uncertainty

associated with every probability estimate. One advantage of subjective logic is precisely that it allows explicit expression of this uncertainty.

The notation ω_x^A is traditionally used to denote opinions in subjective logic, where the subscript indicates the frame or proposition to which the opinion applies, and the superscript indicates the owner entity of the opinion. Subscripts can be omitted when it is clear and implicitly assumed to which frame an opinion applies, and superscripts can be omitted when it is irrelevant who the belief owner is.

Each opinion class will have an equivalence mapping to a type of Dirichlet or a Beta pdf (probability density function) under a specific mapping so that opinions can be interpreted as a probability density function. This mapping then gives subjective opinions a firm basis in notions from classical probability and statistics theory.

Binomial Opinions

Opinions over binary frames are called binomial opinions, and a special notation is used for their mathematical representation. A general n-ary frame X can be considered binary when seen as a binary partitioning consisting of one of its proper subsets x and the complement $\bar{x}$.

(Binomial Opinion) Let $X = \{x,\bar{x}\}$ be either a binary frame or a binary partitioning of an n-ary frame. A binomial opinion about the truth of state x is the ordered quadruple $\omega_x = (b, d, u, a)$ where:

    b (belief)   : the belief mass in support of x being true,

    d (disbelief)   : the belief mass in support of x being false,

u (uncertainty): the amount of uncommitted belief mass,

a (base rate) : the *a priori* probability in the absence of committed belief mass.

These components satisfy $b + d + u = 1$ and $b, d, u, a \in [0, 1]$. The characteristics of various binomial opinion classes are listed below. A binomial opinion:

where $b = 1$ is equivalent to binary logic TRUE,

where $d = 1$ is equivalent to binary logic FALSE,

where $b + d = 1$ is equivalent to a traditional probability,

where $b + d < 1$ expresses degrees of uncertainty, and

where $b + d = 0$ expresses total uncertainty.

The probability projection, or expectation probability, of a binomial opinion on proposition *x* is defined below.

$$\mathrm{E}_x = b + au \qquad\qquad (2.6)$$

Binomial opinions can be represented on an equilateral triangle as shown in Figure 2.5. A point inside the triangle represents a (b, d, u) triple. The belief, disbelief, and uncertainty-axes run from one edge to the opposite vertex indicated by the b_x axis, d_x axis and u_x axis labels. For example, a strong positive opinion is represented by a point towards the bottom right belief vertex. The base rate is shown as a point on the base line, and the probability expectation, E_x, is formed by projecting the opinion point onto the base, parallel to the base rate director line. The opinion $\omega_x = (0.2, 0.5, 0.3, 0.6)$ with expectation value $\mathrm{E}_x = 0.38$ is shown in Figure 2.4 as an example. The class of binomial opinions where $u \geq 0$ is called UB opinion (Uncertain Binomial), whereas the opinion class where $u = 0$ is called DB opinion (Dogmatic Binomial). A DB opinion is equivalent to a classical scalar probability. It can be seen that for a frame *X* of cardinality $k = 2$ a

multinomial and a hyper opinion both have 3 degrees of freedom which is the same as for binomial opinions. [7]

In case the opinion point is located at one of the three vertices in the triangle, i.e. with $b = 1$, $d = 1$ or $u = 1$, the reasoning with such opinions becomes a form of three-valued logic that is compatible with Kleene logic [10]. However, the three-valued arguments of Kleene logic do not contain base rates, so that probability expectation values cannot be derived from Kleene logic arguments. In case the opinion point is located at the left or right bottom vertex in the triangle, i.e. with $b = 1$ or $d = 1$ and $u = 0$, the opinion is equivalent to Boolean TRUE or FALSE, and is called an ABO (Absolute Binomial Opinion). Reasoning with ABOs is the same as reasoning in binary logic. A general UBO corresponds to a Beta pdf (probability density function) normally denoted as Beta $(p \mid \alpha, \beta)$ where α and β are its two evidence parameters. Beta pdfs are expressed as:

$$Beta(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{\alpha-1}(1-p)^{\beta-1} \qquad (2.7)$$

where $0 \leq p \leq 1$, $\alpha > 0$, $\beta > 0$,

With the restriction that the probability variable p 0 if $\alpha < 1$, and p 1 if $\beta < 1$. Let r denote the number of observations of x, and let s denote the number of observations of x. The α and β parameters can be expressed as a function of the observations (r, s) in addition to the base rate a.

$$\begin{cases} \alpha = r + Wa \\ \beta = s + W(1-\alpha) \end{cases} \qquad (2.8)$$

Alternate representation of the Beta pdf is:

$$Beta(p|r,s,a) = \frac{\Gamma(r+s+W)}{\Gamma(r+Wa)\Gamma(s+W(1-a))} p^{(r+Wa-1)}(1-p)^{(s+W(1-a)-1)}, \qquad (2.9)$$

Where $0 \le p \le 1, (r + Wa) > 0, (s + W(1 - a)) > 0,$ with the restriction that the probability variable p 0 if (r + Wa) < 1, and p 1 if (s + W (1 − a)) < 1.The non-informative prior weight denoted by W is normally set to W = 2 which ensures that the prior (i.e. when r = s = 0) Beta pdf with default base rate a = 0.5 is a uniform pdf.

The probability expectation value of the Beta pdf is defined by Eq. below:

$$E(Beta(p|\alpha,\beta) = \alpha/(\alpha + \beta) = \frac{r+wa}{r+s+W}$$

(2.10)

The mapping from the parameters of a binomial opinion $\omega_x = (b,d,u,a)$ to the parameters of a Beta pdf Beta (p | r, s, a) is defined by:

(Binomial Opinion-Beta Mapping)

Let $\omega_x = (b,d,u,a)$ be a binomial opinion, and let Beta (p | r, s, a) be a Beta pdf, both over the same proposition x, or in other words over the binary state space {x, x}. The opinions ω_x and Beta (p | r, s, a) are equivalent through the following mapping:

$$\begin{cases} b = \frac{r}{W+r+s} \\ d = \frac{s}{W+r+s} \\ u = \frac{W}{W+r+s} \end{cases} \Longleftrightarrow \left(\begin{array}{cc} For\ u \ne 0: & For\ u = 0: \\ \begin{cases} r = \frac{Wb}{u} \\ s = \frac{Wd}{u} \\ 1 = b+d+u \end{cases} & \begin{cases} r = b\infty \\ s = d\infty \\ 1 = b+d \end{cases} \end{array} \right) \qquad (2.11)$$

The default non-informative prior weight W is normally defined as W = 2 because it produces a uniform Beta pdf in case of default base rate a = 1/2. The equivalence between binomial opinions and Beta pdf is very powerful because subjective logic operators then can be applied to density functions and vice versa, and also because binomial opinions can be determined through statistical observations. Multinomial opinions described next are a generalisation of binomial opinions in the same way as Dirichlet pdfs are a generalisation of Beta pdfs.

Multinomial Opinions

An opinion on a frame *X* larger than binary where the set of focal elements is restricted to class-1 elements in addition to *X* itself is called a multinomial opinion. The special characteristic if this opinion class is thus that possible focal elements in *R(X)* are always singletons of *X* which by definition are never overlapping.

The frame *X* can have uncertainty mass assigned to it, but is not considered as a focal element. In case $\mu_x \neq 0$ it is called a UMO (Uncertain Multinomial Opinion), and in case $\mu_x = 0$ it is called a DMO (Dogmatic Multinomial Opinion). In case of multinomial opinions the belief vector $\vec{b}_X$ and the base rate vector $\vec{a}_X$ both have *k* parameters each. The uncertainty parameter μ_x is a simple scalar. A multinomial opinion thus contains (2*k* + 1) parameters. It is interesting to note that for binary state spaces there is no difference between hyper opinions and multinomial opinions, because uncertain binomial opinions are always 3-dimensional.

Hyper Opinions [7]

An opinion on a frame *X* of cardinality k > 2 where any element $x \in R(X)$ can be a focal element is called a hyper opinion. The special characteristic if this opinion class is that possible focal elements $x \in R(X)$ can be overlapping subsets of the frame *X*. The frame *X* itself can have uncertainty mass assigned to it, but is not considered as a focal element. In case $\mu_x \neq 0$ it is called a UH opinion (uncertain hyper opinion), and in case $\mu_x = 0$ it is called a DH opinion (dogmatic hyper opinion). In [35] Jøsang and Hankin describe belief fusion with general hyper opinions in subjective logic, and explain how to select the most appropriate belief fusion operator according to the nature of the situation to be modelled.

Definition Hyper Opinion

Assume *X* be to a frame where *R(X)* denotes its reduced power-set, of cardinality $2^{|X|}$-2. Let $\vec{b}_X$ be a belief vector over the elements of *R(X)*, let μ_x be the complementary uncertainty mass, and let $\vec{a}_X$ be a base rate vector over the frame *X*, all seen from the viewpoint of the opinion owner A. The composite function $\omega_X^A = (\vec{b}_X, \mu_x, \vec{a}_X)$ is then A's hyper opinion over *X*. Hyper opinions, with inherent exponential scalability of opinions, represent the most general class of opinions. It is challenging to design meaningful visualisations of hyper opinions because belief masses are distributed over the reduced power-set with partly overlapping elements.

In this thesis we chosen to avoid dealing with hyper-opinions, in large part due to its recent entry to subjective logic research, and the large number of challenges to design for incorporation into our approach, which is more specifically directed at design and development of a decision support software framework.

2.5.4 Operators of Subjective Logic [7]

Table below provides a brief overview of the main subjective logic operators. Additional operators exist for modeling special situations, such as when fusing opinions of multiple observers. Most of the operators correspond to well-known operators from binary logic and probability calculus, whereas others are specific to subjective logic.

Subjective logic is a generalization of binary logic and probability calculus. This means that when a corresponding operator exists in binary logic, and the input parameters are equivalent to binary logic TRUE or FALSE, then the result opinion is equivalent to the result that the corresponding binary logic expression would have produced. We will consider the case of binary logic AND which corresponds to multiplication of opinions [10]. For example, the pair of binomial opinions (in probabilistic notation) $\omega_x = (1, 1, a_x)$ and $\omega_y = (0, 1, a_y)$ produces $\omega_{x \wedge y} = (0, 1, a_x a_y)$ which is equivalent to TRUE $\wedge$ FALSE = FALSE. Similarly, when a corresponding operator exists in probability calculus, then the probability expectation value of the result opinion is equal to the result that the corresponding probability calculus expression would have produced with input arguments equal to the probability expectation values of the input opinions. For example, the pair of argument opinions (in probabilistic notation): $\omega_x = (E_x, 1, a_x)$ and $\omega_y = (E_y, 1, a_y)$ produces $\omega_{x \wedge y} = (E_x E_y, 1, a_x a_y)$ which is equivalent to $p(x \wedge y) = p(x) p(y)$.

In the following sections in this chapter we are discussing some general operators. More operators and their details can be found in [7].

Addition and Subtraction [7]

The addition of opinions in subjective logic is a binary operator that takes opinions about two mutually exclusive alternatives (i.e. two disjoint subsets of the same frame) as arguments, and outputs an opinion about the union of the subsets. The operator for addition first described in [9] is defined below.

(Addition) Let x and y be 2 disjoint subsets of the same frame X, i.e. $x \cap y = \emptyset$. The opinion about $x \cup y$ as a function of the opinions about x and y is defined as:

$$\omega_{x \cup y}: \begin{cases} b_{x \cup y} = b_x + b_y, \\ d_{x \cup y} = \frac{a_x(d_x - b_y) + a_y(d_y - b_x),}{a_x + a_y} \\ u_{x \cup y} = \frac{a_x u_x + a_x u_y}{a_x + a_y} \\ a_{x \cup y} = a_x + a_y \end{cases} \qquad (2.12)$$

By using the symbol "+" to denote the addition operator for opinions, addition can be denoted as $\omega_{x \cup y} = \omega_x + \omega_y$.

(Subtraction) Let x and y be subsets of the same frame X so that x and y, i.e. $x \cap y = y$. The opinion about x\y as a function of the opinions about x and y is defined as: The opinion about x\y is given by

$$\omega_{x \cup y}: \begin{cases} b_{x \backslash y} = b_x - b_y, \\ d_{x \backslash y} = \frac{a_x(d_x + b_y) - a_y(1 + b_y - b_x - u_y),}{a_x - a_y} \\ u_{x \backslash y} = \frac{a_x u_x - a_x u_y}{a_x - a_y} \\ a_{x \backslash y} = a_x - a_y \end{cases} \qquad (2.13)$$

Since $u_{x\setminus y}$ should be nonnegative, then this requires that $a_y u_y \leq a_x u_x$, and since $d_{x\setminus y}$ should be nonnegative, then this requires that $a_x (d_x + b_y) \geq a_y (1 + b_y - b_x - u_y)$. By using the symbol"−" to denote the subtraction operator for opinions, subtraction can be denoted as $\omega_{x\setminus y} = \omega_x - \omega_y$.

Binomial Division and Co-division

The inverse operation to binomial multiplication is binomial division. The quotient of opinions about propositions x and y represents the opinion about a proposition z which is independent of y such that $\omega_x = \omega_{y \wedge z}$. This requires that:

Normal Binomial Division Let $X = \{x, x\}$ and $Y = \{y, y\}$ be frames, and let $\omega_x = (b_x, d_x, u_x, a_x)$ and $\omega_y = (b_y, d_y, u_y, a_y)$ be binomial opinions on x and y satisfying below equation. The division of ω_x by ω_y produces the quotient opinion $\omega_{x \wedge y} = (b_{x \wedge y}, d_{x \wedge y}, u_{x \wedge y}, a_{x \wedge y})$ defined by

$$\omega_{x\bar{\wedge}y} : \begin{cases} b_{x\bar{\wedge}y} = \dfrac{a_y(b_x + a_x u_x)}{(a_y - a_x)(b_y + a_y u_y)} - \dfrac{a_x(1 - d_x)}{(a_y - a_x)(1 - d_y)}, \\[2mm] d_{x\bar{\wedge}y} = \dfrac{d_x - d_y}{1 - d_y}, \\[2mm] u_{x\bar{\wedge}y} = \dfrac{a_y(1 - d_x)}{(a_y - a_x)(1 - d_y)} - \dfrac{a_y(b_x + a_x u_x)}{(a_y - a_x)(b_y + a_y u_y)}, \\[2mm] a_{x\bar{\wedge}y} = \dfrac{a_x}{a_y}, \end{cases} \tag{2.14}$$

By using the symbol "/" to denote this operator, division of opinions can be written as $\omega_{x \wedge y} = \omega_x / \omega_y$.

Subjective Logicopinion operator	SL Symbol	Binary Logic set operator	BL Symbol	Subjective Logic notation
Addition[34]	$+$	XOR	$\cup$	$\boldsymbol{\omega}_{x \cup y} = \boldsymbol{\omega}_x + \boldsymbol{\omega}_y$
Subtraction[34]	$-$	Difference	$\backslash$	$\boldsymbol{\omega}_{x \backslash y} = \boldsymbol{\omega}_x - \boldsymbol{\omega}_y$
Multiplication[12]	$\bullet$	AND	$\wedge$	$\boldsymbol{\omega}_{x \wedge y} = \boldsymbol{\omega}_x \cdot \boldsymbol{\omega}_y$
Division[12]	$/$	UN-AND	$\overline{\wedge}$	$\boldsymbol{\omega}_{x \overline{\wedge} y} = \boldsymbol{\omega}_x / \boldsymbol{\omega}_y$
Co-multiplication[12]	$\sqcup$	OR	$\vee$	$\boldsymbol{\omega}_{x \vee y} = \boldsymbol{\omega}_x \sqcup \boldsymbol{\omega}_y$
Co-division[12]	$\overline{\sqcup}$	UN-OR	$\overline{\vee}$	$\boldsymbol{\omega}_{x \overline{\vee} y} = \boldsymbol{\omega}_x \overline{\sqcup} \boldsymbol{\omega}_y$
Complement[22]	$\neg$	NOT	$\overline{x}$	$\boldsymbol{\omega}_{\overline{x}} = \neg \boldsymbol{\omega}_x$
Deduction[11,14]	$\odot$	MP	$\parallel$	$\boldsymbol{\omega}_{Y \parallel X} = \boldsymbol{\omega}_X \odot \boldsymbol{\omega}_{Y \mid X}$
Abduction[11,15]	$\overline{\odot}$	MT	$\overline{\parallel}$	$\boldsymbol{\omega}_{Y \overline{\parallel} X} = \boldsymbol{\omega}_{X \overline{\odot}} \boldsymbol{\omega}_{Y \mid X}$
Discounting[16]	$\otimes$	Transitivity	$:$	$\boldsymbol{\omega}_x^{A:B} = \boldsymbol{\omega}_B^A \otimes \boldsymbol{\omega}_x^B$
Cumulative Fusion[16]	$\oplus$	n.a.	$\Diamond$	$\boldsymbol{\omega}_X^{A \Diamond B} = \boldsymbol{\omega}_X^A \oplus \boldsymbol{\omega}_X^B$
Cumulative Un-fusion[17]	$\ominus$	n.a.	$\overline{\Diamond}$	$\boldsymbol{\omega}_X^{A \overline{\Diamond} B} = \boldsymbol{\omega}_X^A \ominus \boldsymbol{\omega}_X^B$
Averaging Fusion[16]	$\underline{\oplus}$	n.a.	$\underline{\Diamond}$	$\boldsymbol{\omega}_x^{A \underline{\Diamond} B} = \boldsymbol{\omega}_x^A \underline{\oplus} \boldsymbol{\omega}_x^B$
Averaging Fusion[17]	$\underline{\ominus}$	n.a.	$\underline{\overline{\Diamond}}$	$\boldsymbol{\omega}_X^{A \underline{\overline{\Diamond}} B} = \boldsymbol{\omega}_X^A \underline{\ominus} \boldsymbol{\omega}_X^B$
Belief Constraint[18]	$\odot$	n.a.	$\&$	$\boldsymbol{\omega}_X^{A \& B} = \boldsymbol{\omega}_X^A \odot \boldsymbol{\omega}_X^B$

Table 2.2: Correspondence between probability, set and logic operators. Note that some SL operators do not have a corresponding BL operator, indicated as not applicable (n.a.).

The Averaging Fusion Operator [7]

Assume a frame X containing k elements. Assume two observers A and B who observe the outcomes of the process over the same time periods. Let the two observers' respective observations be expressed as $\vec{r}^A$, $\vec{r}^B$. The evidence opinions resulting from these separate bodies of evidence can be expressed as $(\vec{r}^A,\vec{a})$ and $(\vec{r}^B,\vec{a})$

Averaging Fusion Rule Let ω^A and ω^B be opinions respectively held by agents A and B over the same frame $X = \{x^i | i = 1, , 1\}$. Let $\omega^{A\underline{\lozenge}B}$ be the opinion such that:

$$Case\ I: For\ u^A \neq 0 \vee u^B \neq 0$$

$$\begin{cases} b^{A\underline{\circ}B}(x_i) = \dfrac{b^A(x_i)u^B + b^B(x_i)u^A}{u^A + u^B} \\[2mm] u^{A\underline{\circ}B} = \dfrac{2u^Au^B}{u^A + u^B} \end{cases}$$

$$Case\ II: For\ u^A = 0 \wedge u^B = 0$$

$$\begin{cases} b^{A\circ B}(x_i) = \gamma^A b^A(x_i) + \gamma^B b^B(x_i) \\ u^{A\circ B} = 0 \end{cases} where \begin{cases} \gamma^A = \lim_{\substack{u^A\to 0 \\ u^B\to 0}} \dfrac{u^B}{u^A + u^B} \\[2mm] \gamma^B = \lim_{\substack{u^A\to 0 \\ u^B\to 0}} \dfrac{u^A}{u^A + u^B} \end{cases}$$

$\omega^{A\underline{\lozenge}B}$ is called the averaged opinion of ω^A and ω^B, representing the combination of the dependent opinions of A and B. By using the symbol '$\underline{\oplus}$' to designate this belief operator, we define $\omega^{A\underline{\lozenge}B} \equiv \omega^A\underline{\oplus}\omega^B$.

Trust Transitivity

Assume two agents A and B where A trusts B, and B believes that proposition x is true. Then by transitivity, agent A will also believe that proposition x is true. This assumes that B recommends *x* to A. In our approach, trust and belief are formally expressed as opinions. The transitive linking of these two opinions consists of discounting B's opinion about *x* by A's opinion about B, in order to derive A's opinion about x. This principle is illustrated in Figure.2.4 below. The solid arrows represent initial direct trust, and the dotted arrow represents derived indirect trust.

$$\omega_x^{A:B}$$
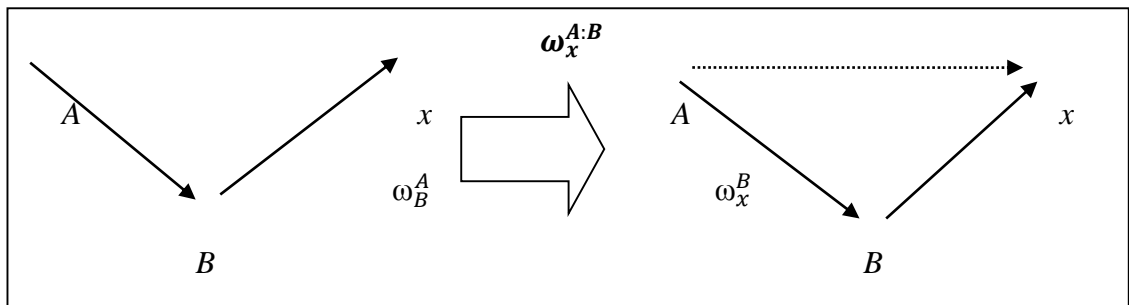
Figure 2.4: Principle of the discounting operator [7]

Trust transitivity, as trust itself, is a human mental phenomenon, so there is no such thing as objective transitivity, and trust transitivity therefore lends itself to different interpretations. We see two main difficulties. The first is related to the effect of "A" disbelieving that "B" will give a good advice. What does this exactly mean? We will give two different interpretations and definitions. The second difficulty relates to the effect of base rate trust in a transitive path. We will briefly examine this, and provide the definition of a base rate sensitive discounting operator as an alternative to the two previous which are base rate insensitive.

The Belief Constraint Operator

The belief constraint operator described here is an extension of Dempster's rule which in Dempster-Shafer belief theory is often presented as a method for fusing evidence from different sources [20]. Many authors have however demonstrated that Dempster's rule is not an appropriate operator for evidence fusion [21], and that it is better suited as a method for combining constraints [15].

Assume two opinions $\omega_X^A \, \omega_Y^B$ over the frame *X*. The superscripts *A* and *B* are attributes that identify the respective belief sources or belief owners. These two opinions can be mathematically merged using the belief constraint operator denoted by"$\odot$", with representation: $\omega_X^{A\&B} = \omega_X^A \odot \omega_X^B$. Belief source combination denoted with "A&B" referring to the joint sources of belief *A* and *B*, thus represents opinion combination with "$\odot$" referring to mathematical combinational algebra. The algebraic expression of the belief constraint operator "$\odot$" for subjective opinions is defined next.

Belief Constraint Operator

$$\omega_X^{A\&B} = \omega_X^A \odot \omega_X^B = \begin{cases} \vec{b}^{A\&B}(x_i) = \dfrac{Har(x_i)}{(1-Con)}, \forall (x_i) \in R(X), x_i \neq \emptyset \\[2mm] u_X^{A\&B} = \dfrac{u_X^A u_X^B}{(1-Con)} \\[2mm] \vec{a}^{A\&B}(x_i) = \dfrac{\vec{a}^{A\&B}(x_i)(1-u_X^A)+\vec{a}^B(x_i)(1-u_X^B)}{2-u_X^A-u_X^B}, \forall x_i \in X, x_i \neq \emptyset \end{cases} \quad (2.15)$$

The term *Har(x_i)* represents the degree of Harmony, or in other words overlapping belief mass, on x_i. The term *Con(x_i)* represents the degree of belief *Conflict*, or in other words non-overlapping belief mass, between ω_X^A and ω_X^B . These are defined below:

$$Har(x_i) = \vec{b}^A(x_i)u_X^B + \vec{b}^B(x_i)u_X^A \sum_{y\cap Z=x_i} \vec{b}^A(y)\vec{b}^B(z), \qquad \forall x_i \varepsilon R(X) \qquad (2.16)$$

$$Con = \sum\nolimits_{y \cap Z = 0_i} \vec{b}^{A}(y)\vec{b}^{B}(z),$$ (2.17)

Expressing Preferences with Subjective Opinions

Preferences can be expressed e.g. as soft or hard constraints, qualitative or quantitative, ordered or partially ordered etc. It is possible to specify a mapping between qualitative verbal tags and subjective opinions which enables easy solicitation of preferences [23]. Table 2.3 describes examples of how preferences can be expressed. All the preference types of Table 2.3 can be interpreted in terms of subjective opinions and further combined by considering them as constraints expressed by different agents. The examples that comprise two binary frames could also have been modeled with a quaternary product frame with a corresponding 4-nomial product opinion.

2.5.5 Applications

Subjective logic represents a generalization of probability calculus and logic under un-certainty. Subjective logic will always be equivalent to traditional probability calculus when applied to traditional probabilities, and will be equivalent to binary logic when applied to TRUE and FALSE statements.

Fusion of Opinions

The cumulative and averaging rules of belief fusion make it possible to use the theory of belief functions for modeling situations where evidence is combined in a cumulative or averaging fashion. Such situations could previously not be correctly modeled within the framework of belief theory. It is worth noticing that the cumulative, averaging rules and Dempster's rule apply to different types of belief fusion, and that,

strictly speaking, is meaningless to compare their performance in the same examples. The notion of cumulative and averaging belief fusion as opposed to conjunctive belief fusion has therefore been introduced in order to make this distinction explicit. [7]

2.6 Related Work

2.6.1 Probabilistic Reasoning in DSS: From Computation to Common Sense

The global objective of this research is to open ways for normative methods, make probability theory more acceptable for DSSs, and to reduce the barriers to dissemination of computer-aided decision making. The objective of this research is to lay a formal foundation for the better understanding of probabilistic models and to improve the user's insight into advice generated by decision support systems by providing a common sense interpretation of probabilistic models and probabilistic reasoning.

This research addresses the problem of reasoning and computerized decision support under uncertainty. The scenario view of decision-theoretic inference provides a useful insight into logic-based Artificial Intelligence schemes for reasoning under uncertainty. They have developed a proposition for decision making under ambiguity using the expected utility theory under the belief-function framework. [33]

2.6.2 Dynamic Decision Support System Based on Bayesian Networks

They described an application of decision support system to the hospitalized patients in the ICU. This system aims at helping the physicians to estimate the nosocomial infections (NI) appearance. The dynamic decision system evolves and proceeds in several stages corresponding to the increasing levels of the patient situation comprehension (scale of time). On each level, a set of knowledge can be generated.

Example & Type	Opinion Expression	
"Ingredient x is mandatory" Hard Positive	Binary frame Binomial opinion	$X = \{x, \bar{x}\}$ $\omega_x : \left(1, 0, 0, \dfrac{1}{2}\right)$
"Ingredient x is totally out of the question" Hard negative	Binary frame Binomial opinion	$X = \{x, \bar{x}\}$ $\omega_x : \left(0, 1, 0, \dfrac{1}{2}\right)$
"My preference rating for x is 3 out of 10 Quantitative	Binary frame Binomial opinion	$X = \{x, \bar{x}\}$ $\omega_x : \left(0.3, 0.7, 0.0, \dfrac{1}{2}\right)$
"I prefer x or y, but z is also acceptable" Quantitative	Ternary frame Trinomial opinion	$\Theta = \{x, y, z\}$ $\omega_\Theta : b(x) = b(y) = 0.6,$ $b(z) = 0.3,$ U=0.1, a(x, y, z)=$\frac{1}{3}$)
"I like x, but I like y even more" Positive rank	Two binary frames Binomial opinions	$X = \{x, \bar{x}\} \ and \ Y = \{y, \bar{y}\}$ $\omega_x : \left(0.6, 0.3, 0.1, \dfrac{1}{2}\right),$ $\omega_y : \left(0.7, 0.2, 0.1, \dfrac{1}{2}\right)$
"I don't like x, and I dislike y even more" Negative rank	Two binary frames Binomial Opinions	$X = \{x, \bar{x}\} \ and \ Y = \{y, \bar{y}\}$ $\omega_x : \left(0.3, 0.6, 0.1, \dfrac{1}{2}\right),$ $\omega_y : \left(0.2, 0.7, 0.1, \dfrac{1}{2}\right)$

Table 2.3: Example preferences and corresponding subjective opinions [7]

In this study we used the Knowledge Discovery from Databases (KDD) as a decisional tool. A data pre-treatment is used in order to transform medical data into standardized data usable by the system. The KDD technique used is the Dynamic Bayesian Networks (DBN). It is used for the modeling of complex systems when the situations are dubious and/or the data are of complex structure. They have implemented the dynamic BNs based on fixed (at t=0 that gives a static BN) and temporal data (daily taken measurements during the hospitalization stay). The application of the developed models for the NI prediction gives good results. [31]

2.6.3 The application of Dempster-Shafer theory

This research explores the weight or justification that evidence affords propositions, with subjects communicating using a belief function in hypothetical legal situations, where justification is a relevant goal. The study demonstrates the potential usefulness of this evidential weight measure as an alternative or complement to the more-studied probability measure. The study identifies the value of understanding evidential weight as distinct from likelihood, informs our understanding of the psychology of individuals' judgments of evidential weight, and furthers the application and meaningfulness of belief functions as a communication language. [32]

2.6.4 Visualizing opinions on opinion triangles

Opinions can be visualized on opinion triangles. Binomial opinions can be mapped to a point in an equal sided triangle. The relative distances from the left side edge to the point represent belief, from the right side edge to the point represent disbelief, and from the base line to the point represents uncertainty. For an arbitrary opinion the three parameters thus determine the position of the opinion point in the triangle. The base line

is the probability axis, and the base rate value can be indicated as a point on the probability axis. Fig.1 illustrates an example opinion about   with the value   = (0.7, 0.1, 0.2, 0.5) indicated by a black dot in the triangle. [7]

The projector going through the opinion point, parallel to the line that joins the uncertainty corner and the base rate point, determines the probability expectation value $p(x) = b_x + a_x u_x$. The parameters b_x, d_x and u_x are equivalent to the traditional $Bel(x)$ (Belief) and $Pl(x)$ (Plausibility) pair of Shaferian belief theory through the correspondence $Bel(x) = b_x$ and $Pl(x) = b_x + u_x$. As by this substantial progress towards a subjective logic framework has been made by Jøsang and co-workers, but existing proposed applets provide a limited approach. Following are some of the limitations and problems in existing applet, which inspire us set the thesis platform based on those.

- Existing applet has limitation of solving only two opinion arguments using single SL operators.
- In existing applet belief values are entered manually only, there is no method to render opinion values directly from dataset.
- Existing applet do not offer mechanism to build complex expressions using multiple opinions and solve them.
- Applet is limited to binomial calculations only.

This thesis presents an approach to build multiple opinions, for which belief values can be rendered from dataset or can be entered manually. An algorithm is implemented to solve complex subjective logic expressions. Our research, based on SL approach,

facilitates the user to explore his hypothesis around a dataset using our framework, in context of binomial and multinomial opinions.
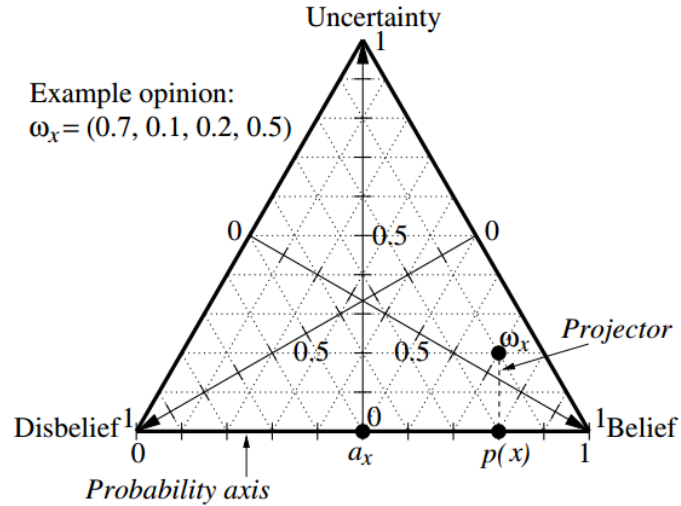

Figure 2.5 Opinion Triangle with example opinion [7]

2.6.5 Legal reasoning with subjective logic

Judges and jurors must make decisions in an environment of ignorance and uncertainty for example by hearing statements of possibly unreliable or dishonest witnesses, assessing possibly doubtful or irrelevant evidence, and enduring attempts by the opponents to manipulate the judge's and the jurors' perceptions and feelings. Three important aspects of decision making in this environment are the quantification of sufficient proof, the weighing of pieces of evidence, and the relevancy of evidence. Jøsang proposes a mathematical framework for dealing with the two aspects, namely the quantification of proof and weighing of evidence. This approach is based on subjective logic, which is an extension of standard logic and probability theory, in which the notion of probability is extended by including degrees of uncertainty. Subjective Logic is a

framework for modeling human reasoning and Jøsang showed how it can be applied to legal reasoning. [14]

There seems to be a consensus between the judicial and statistical professions that probability theory is insufficient for modeling legal reasoning, mainly because probability is not able to express uncertainty. Jøsang and Bondi [14] described a calculus for uncertain probabilities called Subjective Logic, and explored how this calculus can be applied to legal reasoning. The main difficulty with applying Subjective Logic is that there is no consistent way of determining opinions when the evidence at hand cannot be analyzed statistically.

2.7 Summary

In this chapter we discussed about decision support systems (DSS), architecture of DSS, benefits of DSS, also discussed about uncertainty, fuzzy approach, probability theory, Bayesian approach, Dempster-Shafer theory and subjective logic. Along with this we discussed related work in implementation of the above mentioned approaches.

The flexibility of subjective logic makes it simple to express positive and negative preferences within the same framework, as well as indifference/uncertainty. Subjective logic represents a generalisation of probability calculus and logic under uncertainty. Subjective logic will always be equivalent to traditional probability calculus when applied to traditional probabilities, and will be equivalent to binary logic when applied to TRUE and FALSE statements. The advantage of using subjective logic is that real world situations can be more realistically modelled, and that conclusions more correctly reflect the ignorance and uncertainties that necessarily result from partially uncertain input

arguments. Table 2.4 briefly shows the comparison of existing and our approach based on subjective logic operators and opinions

Features	SL Workbench	Jøsang's Opinion Visualization Model	Application of Dampster-Shafer Theory	Model based on Bayesian Approach
Representation of Belief Functions	Yes	Yes	Yes	No
Creating Multiple Opinions	Yes	No	No	No
Rendering belief values from dataset	Yes	No	No	No
Handling Complex SL Expressions	Yes	No	No	No
Considering uncertainty modeling real world problems	Yes	Yes	No	No

Table 2.4 Comparison of existing and our implemented approach

CHAPTER III

THESIS OBEJCTIVE AND RESEARCH METHODOLOGY

3.1 Introduction

Subjective logic has been implemented in different ways to model real world situations and the conclusions reflect the ignorance and uncertainties associated with respective scenarios. Users used to work with maximum two opinions and one operator at one instance with hard coded belief values supplied by the user. An improvement that can be applied is to build multiple opinions, construct and solve subjective logic expressions which contains multiple operators. And another improvement is to populate belief values from existing datasets.

In this chapter, we present the details of our framework. We present an architecture that enable users to access evidence, build opinions and reason data. This thesis introduces an interface to build multiple opinions for binomial and multinomial opinions in which user can build "n" number of opinions, a process to populate belief values from dataset and an algorithm to solve complex subjective logic expressions. In this workbench, 6 subjective logic operators have been coded in C sharp (C#), computational module takes simple and complex expressions into consideration and performs the required calculations as per subjective logic operators. Computational module is intelligent enough to perform the calculations by taking the numeric values of opinions and perform calculations as per subjective logic operators selected by the user.

Framework includes:-

- An interface to display multiple data sets to user.

- An interface to form more than two opinions.

- Selection of dataset and populate belief values.

- Build simple and complex binomial, multinomial expressions.

- Define frame of discernment to filter data.

3.2 Synopsis of Problems and Limitations

DSS is a computer-based support for management decision makers, who deal with semi-structured problems. A properly designed DSS is an interactive software- based system, intended to help decision makers compile useful information from a combination of raw data, documents, personal knowledge, or business models to identify and solve problems and make decisions. While subjective logic has been applied in domains such as trust network modeling and decision support systems, its application in computer vision related domains appears to be limited.

In our research we found that existing proposed applets by Jøsang et al., provide a limited approach of solving two opinion arguments using single SL operators. In existing applets belief values are entered manually and they do not offer mechanism to build complex subjective logic expressions and solve them. There has been no framework for multinomial opinions where user can build multiple opinions.

3.3 Statement of Objectives

The objective is to demonstrate that how subjective logic can be used to express preferences over a variable represented as the possible states in a frame. The flexibility of subjective logic makes it simple to express positive and negative preferences within the same framework, as well as indifference/uncertainty. The input and output parameters of subjective logic are beliefs in the form of opinions. As described in section 3.1.2, the three different equivalent notations of opinions provide rich interpretations of opinions. This also allows the analyst to choose the opinion representation that best suits a particular situation. [7]

In this thesis research, our practical goal is to construct workbench based on subjective logic which proves to be productivity enhancing tool in decision support system where uncertainty is essential part of decision, while also serving as foundation platform for future research. Our goal was to build a subjective logic workbench which has capabilities of rendering opinion values from datasets, solve simple and complex subjective logic expressions for binomial and multinomial opinions, and provide results based on the dataset used. In accordance to this our goal was to develop an algorithm to solve subjective logic expressions built using subjective logic operators. We need a suitable workbench which connects survey data collection directly to a model of evidence based opinions with uncertainty.

3.4 Research Methodology

In order to discuss and decompose the research methodologies, we will follow an approach based on our proposed subjective logic workbench. Disciplines such as statistics, economics, and operations research developed various methods for making

rational choices. These methods, often enhanced by a variety of techniques originating from information science, cognitive psychology, and artificial intelligence, have been implemented in the form of computer programs, either as stand-alone tools or as integrated computing environments for complex decision making.

Recommendation system is framework used to deliver recommendations to the end users. Recommender system is an active research area in the data mining and machine learning areas. There are two basic architectures for a recommendation system, Content-based filtering and collaborative filtering. Content-based systems focus on properties of items. Similarity of items is determined by measuring the similarity in their properties. Collaborative-filtering systems focus on the relationship between users and items. Similarity of items is described by the similarity of the ratings of those items by the users who have rated both items. The term hybrid recommender system is used to describe any recommender system that combines multiple recommendation techniques together to produce its output.

In our research study we found that a framework can be built based on subjective logic in which user will interact with the system in order to build opinions and build subjective logic expressions as per the formulated hypothesis around available dataset.

In our research we investigate the suitability of subjective logic within the decision support context that requires connectivity to complex data, user specification of frames of discernment, representation of complex reasoning expressions, an architecture that supports distributed usage of a decision support tool based on a client-server approach that separates user interactions on the browser side from computational engines for calculations on the server side, and analysis of the suitability and limitations of the

proposed architecture. Benefit of using a client side approach is that user can work on the workbench at any time by accessing it through web.

A computational module is developed which takes simple and complex expressions into consideration and performs the required calculations as per subjective logic operators. Computational module is capable of performing calculations by taking the numeric values of opinions and perform calculations as per subjective logic operators. There is a wide range of subjective logic operators, but we have implemented few basic operators initially, only because the process of coding for operators is time consuming, so in order to save time we implemented basic operators. In the future other operators can be added.

A web-based interface needs to be designed and developed, in which we need to retain the existing notation of subjective logic. For an interactive interface, point and click technique will be used to build opinions and subjective logic expressions.

3.5 Architecture of the workbench

We have followed service-oriented modeling methodology to develop a web-based client side standard data acquisition interface. The discussion in the previous section lead us to design and implement the system architecture with the following features:-

- A platform based on subjective logic for calculating opinion results associated with a subjective logic expression, based upon the inputted opinions. This workbench is suitable for performing queries on datasets of different nature.

- A user interface is designed, which helps the user to form opinions based on the datasets available. This system is capable of populating belief values from selected datasets.

- A user is able to build simple and complex subjective logic expressions based on selection of opinions from persistent storage and subjective logic operators built into our system. Basically a user model, which allows the user to interact, perform and provides results, based on user input which eventually helps in decision making.

- Suitable system level, end-to-end management of a constructed user model, consisting of opinions and subjective logic expressions.

- In the multinomial opinions, user can define his own frame of discernment, where user can define base rate values as well. Based on the frame of discernment the outliers can be excluded from the dataset and user selection is refined.
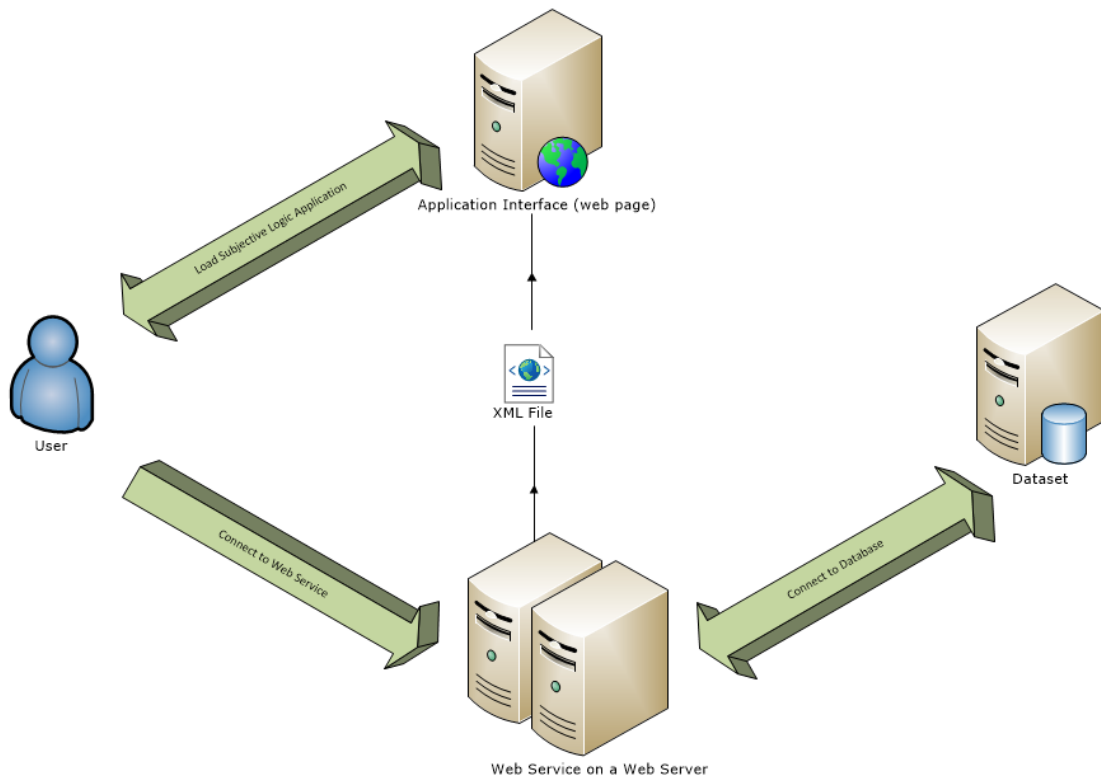
Figure 3.1 Architecture of the system

In Figure 3.1 the architecture of the system is shown, which incorporates:

- **User:** This represents user using the Subjective Logic project. User makes https request to the Subjective Logic page deployed on application server. To make the system secure user need to follow user authentication process, a valid username and password is required to enter the system.

- **Application Platform:** In this a user interface is designed using Extensible Application Markup Language (XAML) which is the language to build Silverlight applications. All the client side functionality is performed in application platform

such as creating opinions, performing calculations to solve simple and complex expressions. Formulas for subjective logic operators are coded in C Sharp (C#).

- **Web Server:** This is web server where we deploy html page with a web service. The html page will internally interact with the web service to get the data from the database. As per the coded formulas defined in section 4.5, Web Service fetches belief values from database and transfer data into an xml file. This way user need not to hit the database again and again to fetch belief values and it decreases run time of the system. Web Services acts as a layer between your application and the database.

- **Dataset:** In the implementation of our framework, we use MySQL database. Our database consists of tables. We use this database to extract the Belief and Disbelief values. This Java web service will extract belief values from dataset. We are using the dataset which contains survey results. In the survey each question X is assigned a question opinion, $\omega_x = (b_x,\ u_x,\ a_x)$, and a complete survey opinion ω_Q is formed using the addition of question evidence frames [6].

    We created a variety of datasets that are modeled from available data within our labs. (R. D. Kent, 2012. Private Communication)

3.6 User Role

As mentioned our starting point still involves the Human Expert as a significant oracular element within the system, but as research continues, one senses how the vision of computationally driven, intelligent support for complex human and machine system

activities may evolve. During the development of workbench we retained notational devices approach to build the user interface. We have retained the existing notations of subjective logic and the user is using point and click technique to build opinions and subjective logic expressions, which are in interactive design today. This workbench is not a standalone system, it is deployed on the web. Standalone versions in general are not portable, therefore building a web-based approach, follows a computer enigma well providing a generic browser interface.

Expert user has the knowledge about subjective logic, user is aware of as how to construct expressions to deduce results. User is given the option of adding as many opinions he want, user access the datasets and select table as per requirement. Then as per selection the data is fetched from the selective tables. So by introducing user role to our framework we can easily manage user profile so that each user has access to his previous queries and results. Benefit of using a client side approach is that user can work on the workbench at any time by accessing it through web.

3.7 Opinions

- User has the option of adding as many opinions needed. This is one special feature which helps user to add "n" number of opinions. User can build simple and complex expressions by using two or more opinions. In case of binomial opinions there are four tuples associated with every opinion namely belief (b), disbelief (d), uncertainty (u) and base rate (a), and in case of multinomial there are 3 tuples namely belief (b), uncertainty (u), and base rate (a). By clicking on each tuple user get the option of selecting data from dataset or user can enter values

57

manually for each opinion. By selecting table the user get to select respective column and then calculation is performed to get the belief value from the selected table. Following formulas have been coded in case of binomial opinions to fetch the belief values from the datasets. We are using the dataset which contains survey results. In the survey each question X is assigned a question opinion, $\omega_x = (b_x,\ u_x,\ a_x)$, and a complete survey opinion ω_Q is formed using the addition of question evidence frames [6]. Dataset we are using in our system has tables which contains binary values. As per subjective logic fundamentals those are observations. For a particular table "r" donate number of observations for "x" and "s" denote number of observations for "$\bar{x}$". A Java web service will extract belief values from dataset as per following formulas and save belief values in an xml file locally. Web Services acts as a layer between your application and the database.

$$b = \frac{r}{W + r + s} \tag{3.1}$$

$$d = \frac{s}{W + r + s} \tag{3.2}$$

b – Belief, d – Disbelief, u – Uncertainty

r, s – Observations,

W – Non informative prior weight

Base rate "a" has been set to 0.5 (default base rate), the default non-informative prior weight "W" is normally defined as W=2 because it produces a uniform Beta pdf in case

of default base rate, a=1/2. Value for Uncertainty "*u*" will be calculated by the formula "*u=1 – (b + d)*", after the user select opinion values for belief "*b*" and disbelief "*d*".

In case of multinomial opinions user can create multiple sub opinions under single opinion as per the hypothesis, by clicking add sub opinion button. A special approach is followed to allow the user to define multiple opinions. Also user can define his own frame of discernment and base rate values, then respective base rate value is fetched from the defined table as per the value. Belief values are fetched in the same way as it is done in case of binomial opinions.

We have the data validation in the code, data will be filtered before getting into the application. If we are looking for belief value from a table in DB and if the column data has a "garbage value" instead of some double value like "0.5" then our system consider that values as zero. This helps to get rid of outliers, and eventually our system does not provide wrong results.

3.8 Simple and Complex Expressions

In the framework, expression builder allows the user to build simple and complex expressions based on subjective logic operators and opinions. Simple or complex expression in our thesis refers to a type of query, created by the user in order to execute his hypothesis. As described in chapter 2, we followed the same approach to solve subjective logic expressions. Our workbench allows user to construct any expression using opinions (created by user) and operators. Expression is parsed into an xml and send to computational module for calculations. In computational module at server side, the expression is parsed using bit string method. After performing the calculations, output

of the expression is shown as an opinion, which contains belief values. Example of a

complex subjective logic expression is $\omega_D^A = (\omega_B^A \otimes \omega_D^B) \oplus (\omega_C^A \otimes \omega_D^C)$

3.9 Explanation of Implemented Algorithm

This algorithm is designed to solve complex subjective logic expressions. We

tried to keep it simple by allowing the user to create an expression in the same format as a

regular mathematics expression is created by using brackets "( )" to make the expression

meaningful. In our system user need to use regular brackets to build an expression. An

example as how user should build his expression is given during the initial orientation

with the system.

To describe the algorithm we consider an example of an expression, namely:

$$\Big(\big((\omega_1 \text{ MUL } \omega_2) \text{ ADD } (\omega_1 \text{ MUL } \omega_3)\big) \text{ ADD } \big((\omega_1 \text{ DIV } \omega_4) \text{ SUB } (\omega_3 \text{ DIV } \omega_5)\big)\Big)$$

When the user presses "Analyze" button to execute expression then whole expression is

parsed into an xml file and this file is sent to the computational module to solve the

expression. After parsing, the calculations are performed in sub-sets, (as defined in

section 3.7) and a new interim, opinion name is assigned to the result of sub expression as

shown below.

$$\Big(\big((\omega_6) \text{ ADD } (\omega_7)\big) \text{ ADD } \big((\omega_8) \text{ SUB } (\omega_9)\big)\Big)$$

$$\Big(\big((\omega_{10}) \text{ ADD } (\omega_{11})\big)\Big)$$

And in the end the resultant opinion is obtained as

$$\left(\omega_{12}\right)$$

By implementing iteration process any complex expression which is in the above format can be handled easily and result is obtained. This algorithm is valid only for expressions which satisfy the notations for binomial and multinomial opinions, which consists of an ordered tuples containing the specific belief masses. Below we describe the pseudo code for the algorithm.

Step 1   parseQuery(QUERY)

Step 2   Check_Validity = Process_Query(QUERY)

Step 3   IF (Check_Validity)

Step 4   FOR (i=0; i < query.length; i++)

Step 5   sub_query = parse_query(QUERY)

Step 6   CreateOpinion = "w" + i;

Step 7   replace (sub_query, CreateOpinion ,QUERY)

Step 8   Operator_Type = Check_Operator(sub_query)

Step 9   Operand1 = Get_Operand1(sub_query)

                              Operand2 = Get_Operand2(sub_query)

Step 10   IF Operator_Type = ADD

                              sub_result = Perform_ADD(Operand1, Operand2)

                ELSE IF Operator_Type = OR

                              sub_result = Perform_OR(Operand1, Operand2)

                ELSE IF Operator_Type = SUB

sub_result = Perform_SUB(Operand1, Operand2)

      ELSE IF Operator_Type = DIV

sub_result = Perform_DIV(Operand1, Operand2)

      ELSE IF Operator_Type = FUSION

sub_result = Perform_FUSION(Operand1, Operand2)

      ELSE IF Operator_Type = UNION

sub_result = Perform_UNION(Operand1, Operand2)

  ELSE

    DisplayInvalidMessage()

A function "*parseQuery*" is created, firstly it checks if the query is valid, then function starts with a loop *for(int i=0;i<query.length;i++)*, this loop run through entire expression. This can handle "n" number of opinions and repeatedly. We have a recursive calling for the function *parseQuery* until the main expression is resolved. Then "*if*" statement executes, which is inside out for loop. Function *sub_query* will solve the sub expression, for example: $\omega_1 AND \omega_2$, the operator can be different. Calculation for different operators has been written in the same function, similarly for other operators. Under each condition we write the code to calculate the expression for different operators. Then we have a function *CreateOpinion* which will create a new opinions and replace the sub expression in the main expression, it replace the sub expression results with ωx_1, ωx_2...... ωx_n. *sub_query* is replaced by *CreateOpinion* in the QUERY. There is a main array where we store all opinions with *b, u, a, d* values for each opinion *source.observablecollection(Opinion).*

62

Link to self-explanatory sequence diagram for algorithm can be found in appendix.

3.10 Workflow of the Workbench

Client make a request to the web server for the html page, this html page will be the response from the server on to the client's machine. This html page runs on the Silverlight Plug-in on client side and performs all the client side functionality like (creating opinions, performing calculations). We have separate webpages for binomial and multinomial opinions. In binomial page firstly user build opinions. By using add opinion button user can add "n" number of opinions, and can delete using delete opinion button. Then by double click on the belief "b" textbox user is redirected to a new window where user has the option to enter belief value manually or user can fetch belief value from a dataset linked in the backend. In this user is able to have a look at all the tables and their respective columns in the dataset, then as per his hypothesis user can select certain table and its column, then an asynchronous call is made to the java web service on the web server, which fetches belief value from the selected table by performing defined calculation, which is described in section 4.5. And the fetched data is transferred into an xml file for later use. The data in the database is stored in the form of tables containing columns of naming value and belief value. Example of a table in the dataset:

Example: 1

Let us assume, that Alice needs treatment for her elbow, and asks her GP (general practitioner) Bob to recommend a good physiotherapist. When Bob recommends David, Alice would like to get a second opinion, so she asks Claire for her opinion about David.

When trust and referrals are expressed as subjective opinions, each transitive trust path can be computed with the transitivity operator (also called discounting operator), where the idea is that the referrals from Bob and Claire are discounted as a function of Alice's trust in Bob and Claire respectively. Finally, the two opinions can be combined using the cumulative or averaging fusion operator. The subjective logic expression for combining the opinions in this example is:

$$\omega_D^A = (\omega_B^A \otimes \omega_D^B) \oplus (\omega_C^A \otimes \omega_D^C)$$

So, opinion ω_B^A represents Alice asking Bob for his opinion on a good physiotherapist, similarly ω_D^B represents Bob's opinion about David, and ω_C^A represents Alice's asking Claire's opinion for David, and ω_D^C represents Claire's opinion about David. More specifically,

$\omega_B^A = (b_B, d_B, u_B, a_B)$, represents Alice asking Bob for his opinion on a good physiotherapist

$\omega_D^B = (b_D, d_D, u_D, a_D)$, represents Bob's opinion about David

$\omega_C^A = (b_C, d_C, u_C, a_C)$, represents Alice's asking Claire's opinion for David

$\omega_D^C = (b_D, d_D, u_D, a_D)$, represents Claire's opinion about David

$\omega_D^A = (b_D, d_D, u_D, a_D)$, represents the resultant opinion.

As our workbench allows user to enter exact values for opinion tuples, so user can enter belief values by themselves and can build the above complex expression. After building

the expression when the user press the "Analyze Expression" button, whole expression is solved in single query and the result is shown to user in subjective logic opinion format.

Example 2:

In this example we will discuss how our system fetches belief values from a given Dataset. We discussed the formulas to calculate belief values in section 3.7. Consider table 3.1, which shows data collected for "Group A" on visit to "ABC shop". Data shows number of people visited "ABC shop" in particular time frame.

In this table value of **r = 20, s = 30, w = 2.0**

Then, following Section 3.7, formulae defined in equation (3.1) and equation (3.2) are implemented to calculate the belief values

$$b = r/(r+s+w), \ d = s/(r+s+w), \ w = 2.0$$

XYZ Shop: Visit Group A	
Column Name	**Column Value**
Windsor	20
Chatham	30

Table 3.1 Example of Table in Dataset

By performing calculation on Table 3.1 we get,

$$b = 20 / (20+30+2) = 0.385$$

$$d = 30 / (20+30+2) = 0.577$$

Similarly we have another dataset for "Group B" as shown in Table 3.2, so in this case

$$r = 28, s = 22, w = 2.0$$

Then, by performing calculations on Table 3.3, we get,

$$b = 28 / (28+22+2) = 0.54$$

$$d = 22 / (28+22+2) = 0.42$$

XYZ: Shop Visit Group B	
Column Name	Column Value
London	28
Hamilton	22

Table 3.2 Example of Table in Dataset

Now our web service saves these belief values in an xml file. After that, the user can build subjective logic expressions using opinions and subjective logic operators by choosing the opinions and operators from respective dropdown functionality. A proper format needs to be followed to build a complex expression. We kept it simple by allowing the user to create a nested expression in the same format as a regular

mathematics expression is created by using brackets "( )" to make the expression meaningful.

Based on the above data user is interested to analyse the total number of visits in a particular frame of time. User can build a SL expression using the above calculated belief values. For example as shown below:

$\omega_{XYZ}^{A} = (b_{XYZ}, d_{XYZ}, u_{XYZ}, a_{XYZ})$, represents opinion for Group A referring their visit to XYZ shop.

$\omega_{XYZ}^{B} = (b_{XYZ}, d_{XYZ}, u_{XYZ}, a_{XYZ})$, represents opinion for Group B referring their visit to XYZ shop.

So as per above example,

$\omega_{XYZ}^{A} = (0.38, 0.58, 0.04, 0.5)$ and $\omega_{XYZ}^{B} = (0.54, 0.42, 0.04, 0.5)$

Now user can build a SL expression (as discussed in section 3.8) using these two opinions based on his hypothesis. User can evaluate the following expression using our system.

$$\omega_{XYZ}^{AB} = ((\omega_{XYZ}^{A} + \omega_{XYZ}^{B}) \underset{-}{\oplus} \omega_{XYZ}^{B})$$

So, the system performs calculations in the computational module and provide resultant opinion in the following format:

$$\omega_{XYZ}^{AB} = (b, d, u, a)$$

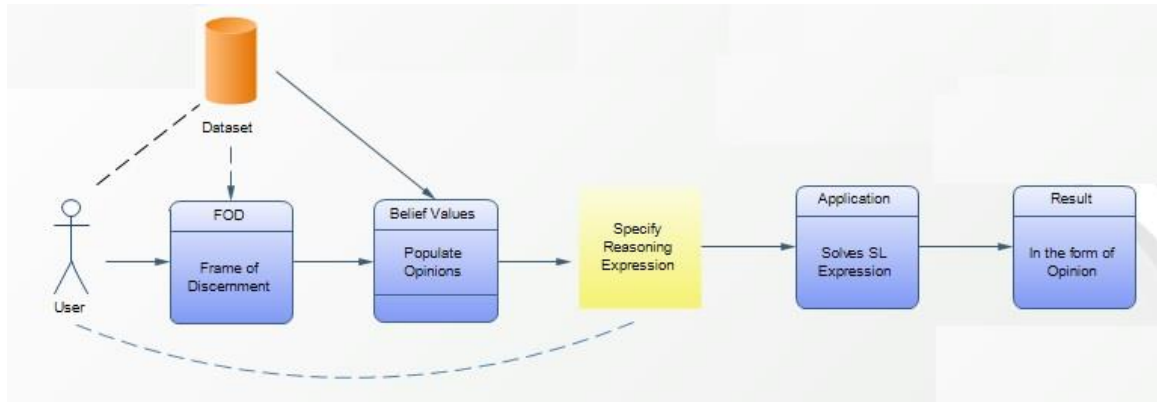Figure 3.2 shows a high-level diagram of working of our system.

Figure 3.2 High-Level Work flow diagram of system

Then user builds his own simple or complex expression and by pressing the Analyze Expression button, the expression is calculated in the computational module, where the engine parse the expression into xml and sent to for calculations. An algorithm mentioned in section 3.8 is implemented to handle complex subjective logic expressions. The computational module works in the back end. It takes expression and follows parsing technique to perform calculations as per the operators used in the expression.
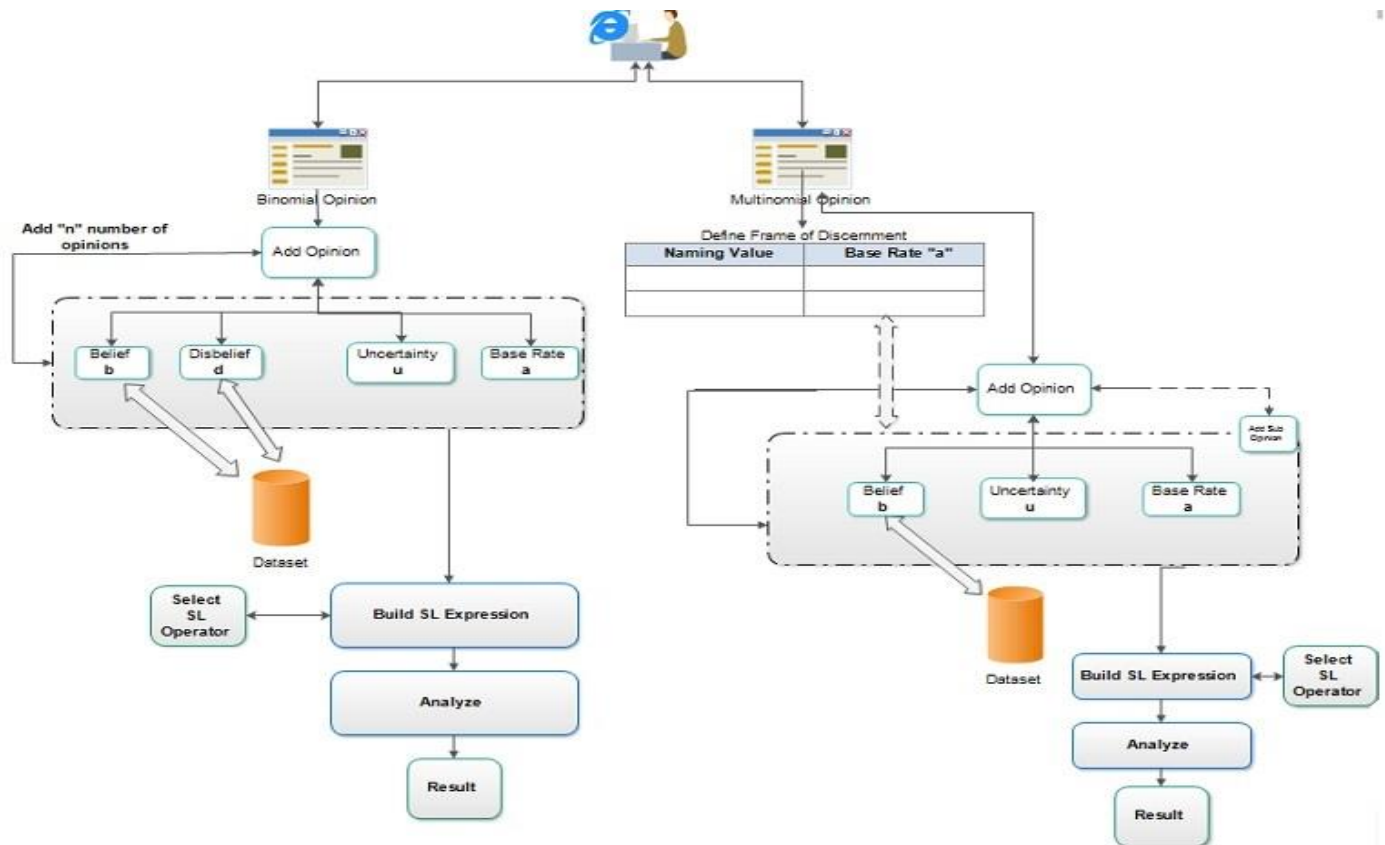
Figure 3.3 Work flow of workbench

3.11 Summary

In this chapter, we have discussed the implementation of our proposed framework in detail. We leave discussion of verification of our approach and results to Chapter 4. The implemented framework is not domain specific. Our first goal was to design an architecture of the system, motivated and guided, in part, by previous work done in the area of decision support systems. An architecture of the framework is presented which has the capability of building multiple opinions for both binomial and multinomial cases, with enhanced options for a user to extract belief values direct from datasets, then enabling the user to construct both simple and complex subjective logic expressions

based on opinions and subjective logic operators. In section 3.9, an algorithm for solving complex expressions is discussed, and step by step functionality of algorithm is defined. This chapter explains further the design and technical details of the architecture, based on actual implementation; more detailed discussion of our testing approach is provided in Chapter 4 along with details of verification of the system.

CHAPTER IV

IMPLEMENTATION AND VERIFICATION

4.1 Background

The main objective of this thesis research is to provide a workbench to build and solve complex subjective logic expressions. In our framework, we present a user interface to build multiple opinions for binomial and multinomial frames and allowing the user to fetch belief values from dataset as per hypothesis. Moreover, we introduce a mechanism to build complex subjective logic expressions based on opinions and then implementing an algorithm to solve the expressions.

4.2 Implementation

In this thesis for workbench development, design of the application is based on XAML which is the language to build Silverlight applications. Silverlight technology is a complete client side scripting and interacts with server via a web-service. The following features in particular make Silverlight a viable technology for building applications:

- **WCF RIA Services:** Windows Communication Foundation (WCF) Rich Internet Application (RIA) Services provide an elegant solution for handling the transmission of data across the tiers of an application, data validation, and change tracking. In doing so, they provide a unified model for client-side and server-side development, making a traditionally difficult job much easier for the developer.

- **Rich Data Controls:** Silverlight provides a rich library of over sixty controls complimented by open source and vendor control packs. The new,

functionality-rich, data bound controls such as the *DataGrid*, *ContentControl*, *DatePicker*, and charting controls provided by the Silverlight Toolkit make it much easier to display data in an attractive manner. New controls such as the *RichTextArea* control make it much easier to capture formatted text input. Working with large quantities of data and handling data paging is also much easier with the *DataPager* control, which largely automates this job.

A Java web service has been developed to interact with the dataset. Due to client-side characteristics, Silverlight applications need to perform particular tasks to get data. It does not support client-side databases, so the way to retrieve data is through services. A java web-service is developed to fetch data from datasets. One significant advantage of Silverlight is that it can run from any type of server. Silverlight also runs on the client side. The plugin has a CLR (Common Language Runtime) embedded, so that it hosts our application. On the server side, the only thing we need to do is to serve the files (most importantly **.xap* file) that will be downloaded to the client side when requested.

An XAP file is the compressed output file for the Silverlight application. These XAP files are essentially .zip files that contain an assembly manifest file and one or more assemblies. So, the XAP file includes *AppManifest.xaml*, compiled output assembly of the Silverlight project (*.dll*) and any other resource files referred by the Silverlight application. Web pages like *.aspx* files and .html files use the Silverlight components by loading the *.xap* files using the *<object>* tag in the HTML or by using *<asp:Silverlight>* tag in the ASP.NET pages. The flow diagram of a Silverlight application from creation to running at client browser can be depicted as in Figure 4.1.
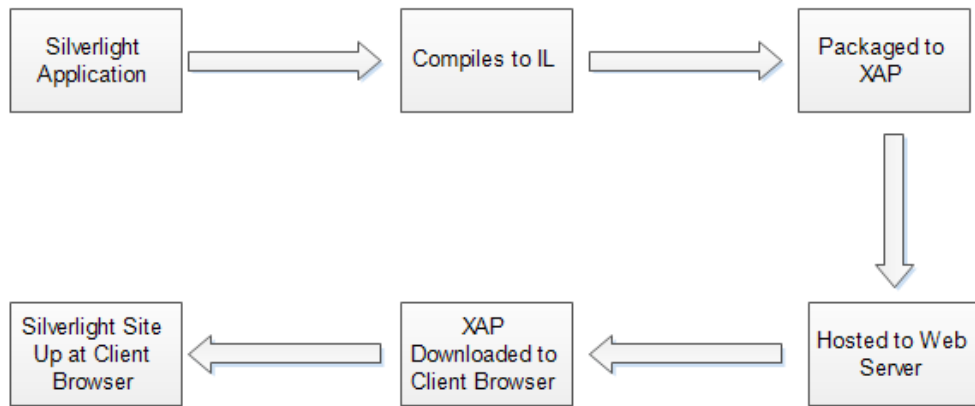
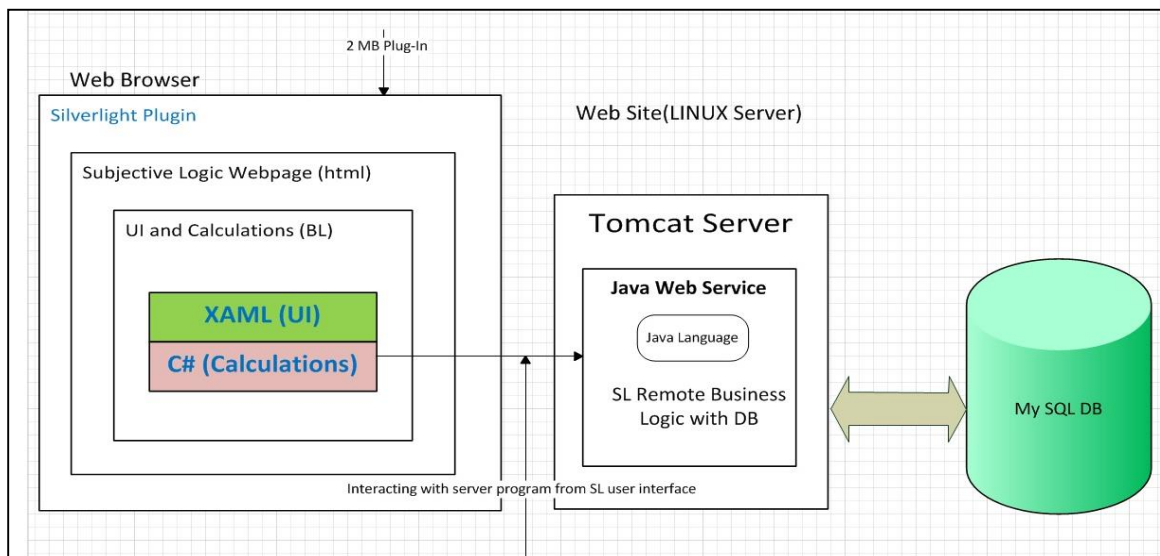**Figure 4.1 Flow Diagram of a Silverlight Application**

Figure 4.2 Structure of the System

Silverlight Plug-in: This is a cross platform technology which can run on any browser and any platform and perform some basic client side functionalities. In our framework Silverlight coding has two parts:

- **Extensible Application Markup Language (XAML):** This is to design the user Interface of the application like (buttons, data grids and graphs etc.).

- **C Sharp (C#):** All the client side validations and calculations are performed and programmed in the language. It is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, procedural, generic, object-oriented (class-based), and component-oriented programming disciplines. C# offers XML support for Web-based component interaction and full platform support for existing code integration.

Java Web Service: This is our service side programming. It receives an asynchronous web request from the html page and accordingly sends a query to data base to fetch the data. The same data will be sent to html page after performing calculations. Web Service fetches belief values from database and transfer data into an xml file. This web service is written in Java Language.

To develop the workbench with Silverlight application development, we used Visual Studio 2012, Silverlight SDK and Silverlight 5 Toolkit. (http://silverlight.codeplex.com/releases/view/78435).

In future development, depending upon the nature of the dataset, we just need to develop a web service to join our application platform to fetch data from the database.

4.3 Computational Module

Coding has been done in C Sharp (C#) for selected formulae of subjective logic operators. C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, procedural, generic, object-oriented (class-based), and component-oriented programming disciplines. The main advantage of C# is that it runs

on the CLR, making it easy to integrate with components written in other languages (specifically, CLR-compatible languages).

Computational module takes simple and complex expressions into consideration and performs the required calculations as per subjective logic operators. Computational module is capable of performing calculations by taking the numeric values of opinions and perform calculations as per subjective logic operators (defined in section 2.5.4) selected by the user. There is a wide range of subjective logic operators, but we have implemented few basic operators initially, only because the process of coding for operators is time consuming, so in order to save time we implemented basic operators. In case of some operators, subsequent to their calculations, there is a controversy, for these reasons we have not implemented those operators. In the future other operators can be added. Our system implemented only the following operator subset for calculations, thereby establishing the *proof of concept* for the system.

- Multiplication/Conjunction/AND
- Co-multiplication/Disjunction/OR
- Division/Un-conjunction/UN-AND
- Addition/SUM
- Subtraction/Difference
- Averaging Fusion

Working of computational module shows how it delivers opinion to the end users. User enters in the system with its unique id or new user can create its new profile. After login authorization and authentication is done which validate users, user can start using the

system. User build simple and complex expressions using opinions and subjective logic operators, the expression and the belief values of the opinions fetched from the dataset are sent to the computational module, where calculations are performed. Computational module contains formulas for performing calculations. Here the queries made by user are parsed into xml and sent to for calculations. An algorithm mentioned in section 3.9 has been followed to handle the complex subjective logic expressions.

We have two modules to build opinions and expressions and perform calculations, based on binomial and multinomial opinions.

4.3.1 Binomial Module

In Binomial, user can build opinions, select a belief value from dataset. The add/delete button inserts or deletes rows from the table. Once you add an opinion, the opinion is added to the dropdown to build an expression and perform calculation. You can Reset/Analyze. The result is displayed in the corresponding output window on the right with their respective graph results. Screen shots of our system can be found in appendix.

Code:

The UI for Binomial is in XAML.

Some of the main events are:

btn_AddOpinion_Click – to insert rows

dgOpinions_CellEdit – to select a value for belief from the pre-defined tables (this is the data grid edit option, clicking on it would open a popup window to select the respective values)

objChild_Closed – closes the popup window and populates the result into the data grid

btnDelOpinion_Click – Deletes selected rows

ValidateQuery() – Validates the selected query

Calculate – Performs the expression calculation and returns *ObservableCollection<Opinion>* based on the results and outputs the result the corresponding output window which is used to build graph results

4.3.2 Multinomial Module

Multinomial is similar to the Binomial, except for in the data grid you have opinions, where you could add multiple subset of one opinion by clicking on the corresponding row 'add' button. Events are pretty similar to the ones in binomial. Additional events include Frame of Discernment table open/save event.

Frame of Discernment

User can define his own frame of discernment, where user could enter his own set of values and user will be able to fetch belief values based on the values defined in frame of discernment. User can define the naming value and the respective base rate for that value, when the user will select the similar value from dataset then respective base rate will be fetched from the frame. This way user can filter the dataset as per the hypothesis.

4.4 Verification

In order to test our framework, we have implemented two approaches. Our system is based on conceptual reasoning. We do not claim our system to be a complete recommendation framework, but we are sure that it will serve as foundation platform for future research. We cannot verify that the system is correct because it is consisted with

non-idealized framework. But we considered two approaches to verify the correct working of the system by satisfying the fundamentals. We do not claim that it is the optimal solution, it is one of the solutions.

To verify the system design, 10 hand crafted subjective logic expressions are built and calculation is done both manually and on our system with boundary level cases and then results are compared. In our thesis research, we are focused on providing recommendation to users based on their hypothesis.

4.4.1 Verification of System Design

The verification of basic requirements is to test the core elements of the application. Initially user needs to provide a valid username and password for authentication. After the authentication process, the user is redirected to the homepage. The Figure 4.3 represents the login page.
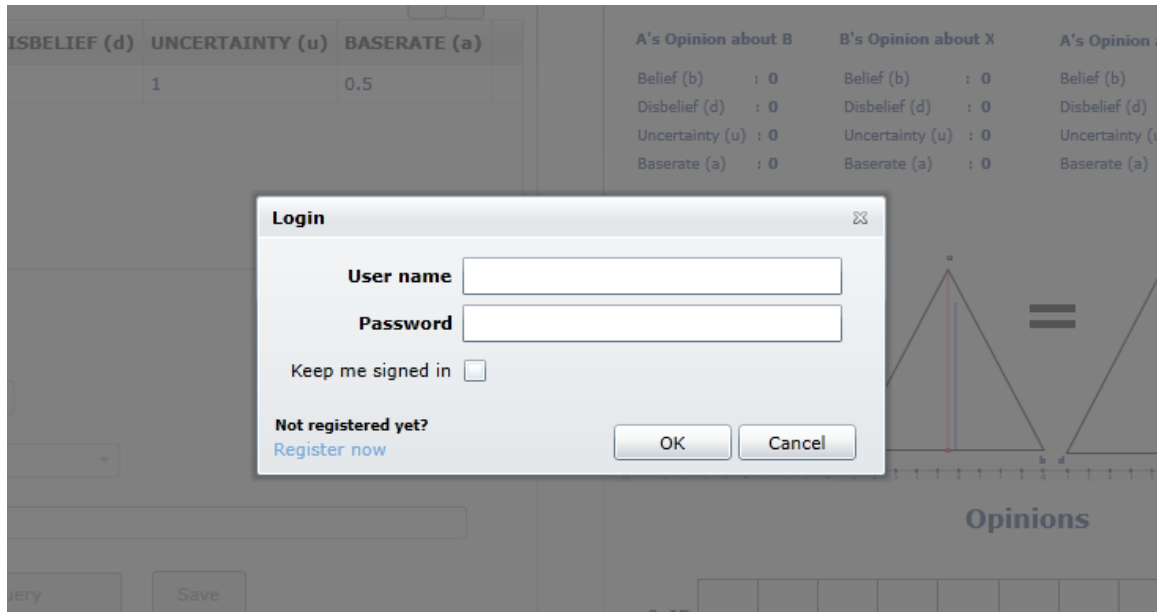

Figure 4.3 Login page for user

Username appears on the right of the header section there is a logout button available that deactivates user session and redirects user to the login page. On the top right user can select "Binomial" and "Multinomial" pages. In "Binomial" page left frame down to the header named under "Add Opinion and Build Expression for binomial operators" is used to build opinions, user can add "n" number of opinions by using add button and can delete by using delete button as per the requirement. Opinions are represented by ω_1, ω_2, ω_3….and so on. User need to double click on the first textbox i.e. "Belief (b)", by this user is redirected to another pop-up, where user need to select Table and column name to fetch belief value from the respective dataset. Same procedure is followed to fetch "Disbelief (d)" value. And then the "Uncertainty" value is calculated automatically by the formula b + d + u = 1, value of Base rate (a) is set to 0.5, which is default base rate value for uniform beta pdf. Then as per the hypothesis user builds an expression by selecting opinions and operators from respective dropdowns, and the expression can be seen in "Expression" textbox. And when user press the "Analyze Expression" button, the result is calculated in the back end and shown in right side frame, with result values for Belief (b), Disbelief (d), Uncertainty (u) and Base rate (a). With the result values user get to know about this hypothesis outcome.

In "Multinomial" page, user can define his own "Frame of Discernment" by clicking on the button on top left in left frame. In this, based on his hypothesis user can define values and their respective base rates, by doing this the system will filter the dataset as per the frame of discernment and the outliers can be distinguished and excluded. And when user select the value from the dataset then respective base rate is

fetched from the defined frame value. Rest of the steps remains same as "Binomial page" to fetch belief (b) and uncertainty (u) values.

In order to verify that our system gives correct results as per user input, we need to verify the system design. We built 10 hand crafted subjective logic expressions and did manual calculations with boundary level cases. Then, the same expressions were built and run with the same tuple values and operators on our workbench. In this we also verify that our system should firmly hold the fundamentals of subjective logic. As in case of tuple values of an opinion following formulas should hold correctness.

$b + d + u = 1$ (**In case of binomial**) » $u = 1 - (b + d)$

As our system in first step takes belief (b) and disbelief (d) values then in next step as per this formula calculates values for uncertainty (u).

$\sum \vec{b} + u = 1$ (**In case of multinomial**)

Finally, results are compared to verify the working of the workbench. Along with that, a few binomial expressions have been calculated on the existing subjective logic operators demo [36] by Jøsang and on our implemented workbench.

These verification results show that subjective logic operators have been implemented correctly and our implemented algorithm also perform correct calculations. Along with that we did positive and negative testing. In positive testing correct values were used as input $(0 \geq n \geq 1)$ and we found that the result obtained is also correct. And in case of negative testing wrong values were used as input $(0 < n < 1)$ and we found that

our system gives error message as a result. A list of hand crafted queries can be found in the Appendix. Although not a formal proof of our system software, we have tested extensively, using extreme cases, and the underlying subjective logic software approach is consistent.

4.5 Test Results

4.5.1 Verification of System Design

In this the system design is verified by comparing the results obtained for 10 hand crafted complex subjective logic expressions. We found that the results obtained by performing manual calculations and results obtained by running the same expressions on our workbench, comes out to be same for all of the 10 queries, which includes boundary level cases. Also, results for a few simple binomial expressions have been compared with the results obtained for same expressions from existing subjective logic operators demo by Jøsang [36]. Based on our verification approach we can state that we have reasonable confidence on the results obtained, but the system must be rigorously analyzed for correctness.

4.6 Summary

In this chapter, we have discussed the implementation of our framework in detail. We have also verified our approach and presented the results from our testing. The framework is not domain specific. Our focus in this thesis was on constructing a software module that supports opinion formation, application of well-defined operators for subjective reasoning and a toolkit and workbench that provides a platform for users to create and explore scenarios (different hypothesis) based on datasets. Our system will

help the user to be better informed about the degree of uncertainty associated with a hypothesis, which further helps in decision making. The framework can be used independently of any another block to increase the user experience, and also contributes in the field of decision making by providing direct evidence suitable for validating strategies, intelligence based prediction and automation of user reasoning on complex data.

CHAPTER V

CONCLUSION AND FUTURE WORK

In this chapter, we conclude our framework and discuss some areas for future work.

5.1 Conclusion

This thesis work presents a reasoning framework based on Subjective Logic in decision support systems, which consists of a belief model called opinion and set of operations for combining opinions. Subjective Logic is directly compatible with traditional mathematical frameworks, but is also suitable for handling ignorance and uncertainty. We followed Jøsang's approach of belief reasoning with subjective logic. This research has been accomplished in a number of steps.

Initially, the existing Jøsang's subjective logic demonstrations for belief visualization, subjective logic operators and trust networks are studied and based on our problem statement described in section 1.1, a new framework is built. In our framework, we provide a suitable workbench which connects survey data collection directly to a model of evidence based opinions with uncertainty that also support subjective reasoning. As we mentioned in section 3.6, in our framework we enable the user to add 'n' number of opinions and populate a set of belief values by direct query of our datasets, in order to build a model of a complex subjective logic assertion. Secondly, other contributions is to display multiple datasets to user. This helps the user to select datasets as per his hypothesis.

Once the user creates opinions and populates belief values then as described in section 3.7 our system allows the user to build simple and complex subjective logic expressions using opinions and subjective logic operators to deduce a hypothesis. A computational model to handle complex expressions is one of the main contributions of this thesis. An algorithm has been implemented which is described in section 3.9. In reference to section 4.3.1 and 4.3.2, our workbench allows the user to build binomial and multinomial opinions. Two separate interfaces has been developed for both. In case of binomial user can add "n" number of opinions and can populate belief values for all the opinions from the dataset, and in case of multinomial user can define his own frame of discernment, by this user can filter the dataset as per the hypothesis.

In order to test our workbench, we did the verification of the system design. We constructed 10 hand crafted expressions and perform the calculations manually and on the workbench and compared the results.

Although our work is still preliminary, the prototype framework can be used to support and conduct further research, and provide benchmarks and new research hot spots. The framework can be used independently to increase the user experience, and contributes in the field of decision making by providing direct evidence suitable for validating strategies for further, intelligence based, prediction and automation of user intention.

5.2 Future Work

We address briefly some potential areas which can be addressed in future work based on the experience gained in this thesis research. There is still considerable scope for improvement, both theoretical and practical.

5.2.1 Subjective Logic Operators

In this thesis research, we have implemented a limited set of operators, which thereby limits the use of our workbench. Additional operators can be implemented to enrich the user experience and opportunities for increasingly sophisticated reasoning. Most of the operators we have implemented correspond to well-known operators from binary logic and probability calculus. There is still scope for exploring operators beyond the scope of the current set established by Jøsang and others. [7]

5.2.2 Extension to Hyper Opinions

Our system is limited to work for binomial and multinomial opinions. But, this work can be taken forward to work with hyper opinions. An opinion on a frame X of cardinality $k > 2$ where any element $x \in R(X)$ can be a focal element is called a hyper opinion. The nature of such opinions involves exponential scaling on the opinion tuples and on the computational complexity. These pose challenges for software development and for algorithmic performance.

5.2.3 Enrich user experience

In our interface we tried to make built an interface based on subjective logic approach, which is easy to understand and work efficiently. But still, there is a lot of scope for improvement. User can be better informed of the outcomes by extending the

analytical capabilities of the system by incorporating Beta Probability density functions and Dirichlet Probability Density Functions, as well as other modes of visualization that enable users to observe and detect belief patterns of interest.

In addition, in developing a proof-of-concept software system for laboratory use, one focuses on fundamental issues of design and testing; however, there are many features that would enhance the user experience of a full-fledged decision support system. Such features should include support for interacting with data directly during the creation of multiple frames of discernment, modification of opinion values dynamically to support scenario exploration, improvements to error detection and reporting, and many other similar factors.
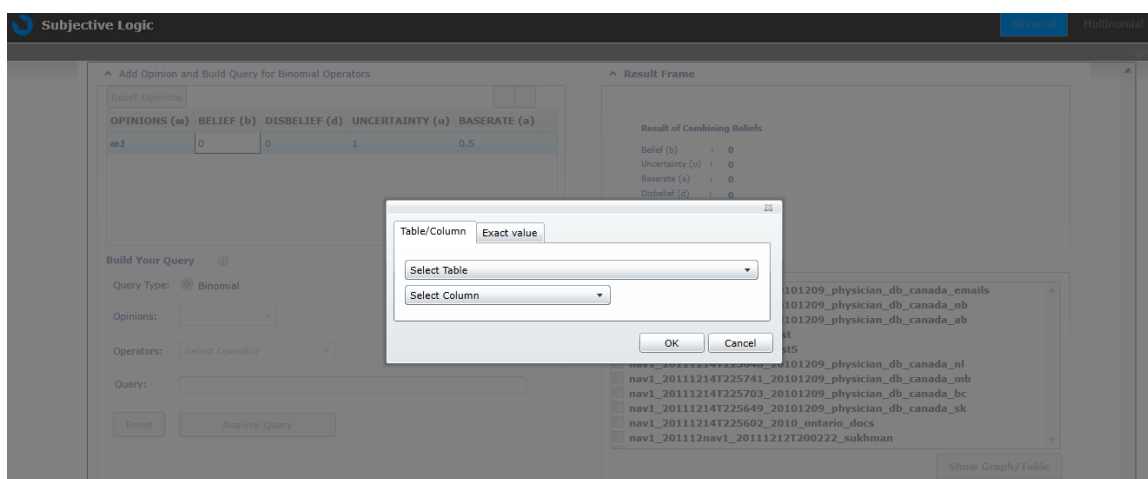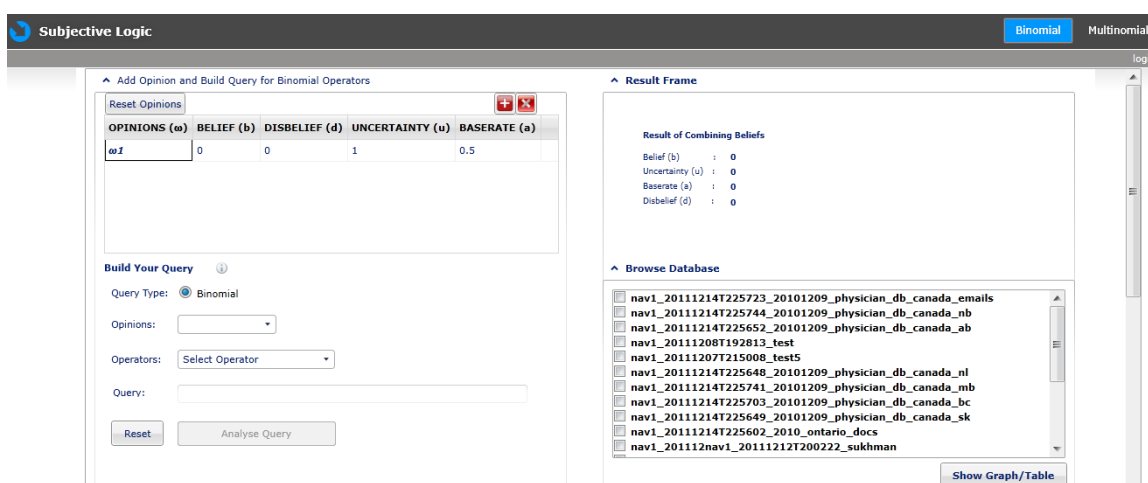
APPENDICES

APPENDIX A

Subjective Logic Workbench

The following figures illustrate the data visualization framework:

Binomial Page:

Subjective Logic

Add Opinion and Build Query for Binomial Operators

Add New Opinion

OPINIONS (ω)	BELIEF (b)	DISBELIEF (d)	UNCERTAINTY (u)	BASERATE (a)
ω1	0.2	0.33	0.47	0.5
ω2	0	0	1	0.5

Build Your Query ⓘ

Query Type: ● Binomial

Opinions: [▼]

Operators: [Select Operator ▼]

Query:

Subjective Logic

OPINIONS (ω)	BELIEF (b)	DISBELIEF (d)	UNCERTAINTY (u)	BASERATE (a)
ω1	0.2	0.33	0.47	0.5
ω2	0.12	0.422	0.458	0.5

Build Your Query ⓘ

Query Type: ● Binomial

Opinions: [▼]

ω1
ω2

Operators: [▼]

Query: [ ]

Subjective Logic

OPINIONS (ω)	BELIEF (b)	DISBELIEF (d)	UNCERTAINTY (u)	BASERATE (a)
ω1	0.2	0.33	0.47	0.5
ω2	0.12	0.422	0.458	0.5

Build Your Query ⓘ

Query Type: ⦿ Binomial

Opinions: ω1 ▾

Operators: ▾

Select Operator
Multiplication/ADD
Disjunction/OR
Subtraction/SUB
Division/DIV
Averaging Fusion/FUSE
Addition/UNION

Query:

Reset   Save

Subjective Logic

▲ Add Opinion and Build Query for Binomial Operators

OPINIONS (ω)	BELIEF (b)	DISBELIEF (d)	UNCERTAINTY (u)	BASERATE (a)
ω1	0.24	0.2	0.56	0.5
ω2	0.2	0.24	0.56	0.5

Build Your Query ⓘ

Query Type: ⦿ Binomial

Opinions: ω2 ▾

Operators: Subtraction/SUB ▾

Query: ω1SUBω2

Reset   Analyse Query   Save

89

Multinomial Page:
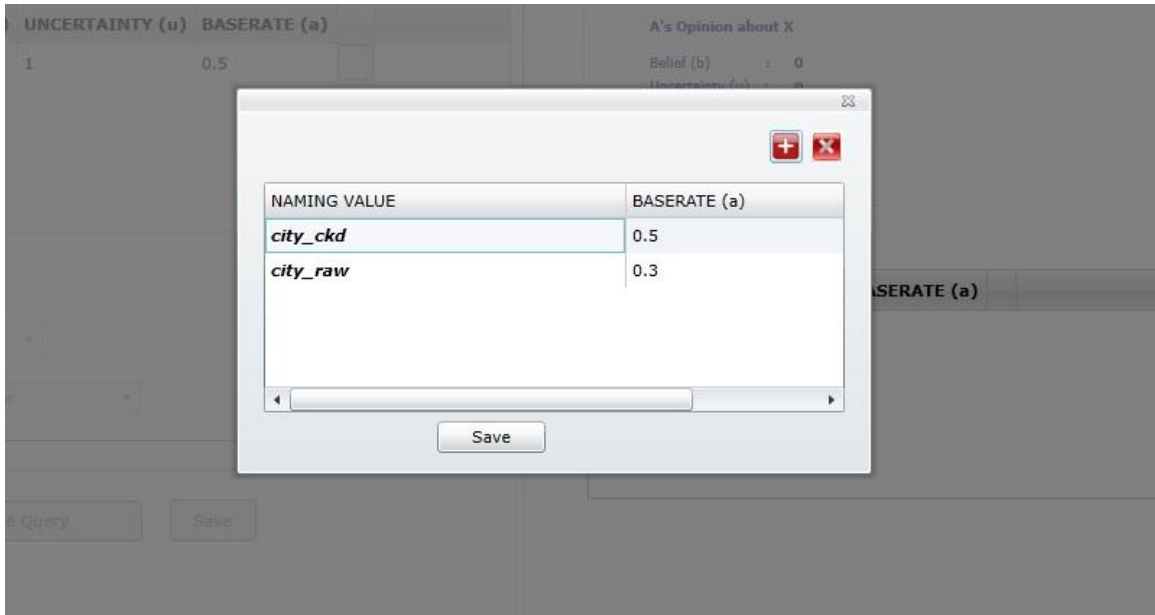
Link to "How to use SL Workbench" document:

https://docs.google.com/document/d/1MprFbrNamA4bNb4x8gb8J9Z1MIJFfCU7rwtZv_H5hnM/edit?pli=1

Link to "Sequence Diagram for Algorithm to solve simple and complex SL expression"

https://docs.google.com/document/d/15DuYh8T6V63Tv4oCKPFwYfDFcQvvqlT1Qus4bdK1RdI/edit?usp=sharing

REFERENCES

1. Probabilistic Reasoning in Decision Support Systems: From Computation to Common Sense by Marek J. Druzdzel

2. F. T. de Dombal, D. J. Leaper, J. R. Staniland, A. P. McCann, and Jane C. Horrocks. Computer-aided diagnosis of acute abdominal pain. British Medical Journal, 2(NUMBER):9 13, April 1972.

3. Randolph A. Miller, Harry E. Pople, Jr., and Jack D. Myers. Internist {1, an experimental computer-based diagnostic consultant for general internal medicine}. New England Journal of Medicine, 307(8):468, 476, August 1982.

4. Victor L. Yu, Lawrence M. Fagan, Sharon Wraith Bennett, William J. Clancey, A. Carlisle Scott, John F. Hannigan, Robert L. Blum, Bruce G. Buchanan, and Stanley N. Cohen. An evaluation of MYCIN's advice. In Buchanan and Short life [16], chapter 31, pages 589-596.

5. A. Jøsang, and V. A. Bondi, "Legal Reasoning with Subjective Logic", Artificial Intelligence and Law, 8: 289–315, 2000.

6. Robert D. Kent, Jason McCarrell, Gilles Paquette, Bryan St. Amour, Ziad Kobti, and Anne W. Snowdon (2010), "Application of Subjective Logic to Health Research Surveys," in Advances in Intelligent Decision Technologies, vol. 4, Phillips-Wren, Gloria; Jain, Lakhmi C; Nakamatsu, Kazumi; Howlett, Robert J., Springer Berlin Heidelberg, 2010, pp. 383-392

7. Audun Jøsang. (2010 January 04). Subjective Logic (Draft book) [Online], Available: http://folk.uio.no/Jøsang/papers/subjective_logic.pdf

8. M. Smithson. Ignorance and Uncertainty: Emerging Paradigms. Springer, 1988.

9. U. Hoffrage, S. Lindsey, R. Hertwig, and G. Gigerenzer. Communicating statistical information. Science, 290(5500):2261–2262, December 2000.

10. M. Fitting. Kleene's three-valued logics and their children. Fundamental Informaticae, 20:113–131, 1994.

11. A. Jøsang. Conditional Reasoning with Subjective Logic. Journal of Multiple-Valued Logic and Soft Computing, 15(1):5–38, 2008.

12. A. Jøsang and D. McAnally. Multiplication and Comultiplication of Beliefs. International Journal of Approximate Reasoning, 38(1):19–51, 2004.

13. A. Jøsang, and V. A. Bondi, "Legal Reasoning with Subjective Logic", Artificial Intelligence and Law, 8: 289–315, 2000.

14. A. Jøsang, S. Pope, and M. Daniel. Conditional deduction under uncertainty. In Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005), 2005.

15. Simon Pope and Audun Jøsang. Analysis of Competing Hypotheses using Subjective Logic. In Proceedings of the 10th International Command and Control Research and Technology Symposium (ICCRTS). United States Department of Defense Command and Control Research Program (DoDCCRP), 2005.

16. A. Jøsang, S. Pope, and S. Marsh. Exploring Different Types of Trust Propagation. In Proceedings of the 4th International Conference on Trust Management (iTrust), Pisa, May 2006.

17. A. Jøsang. Cumulative and Averaging Unfusion of Beliefs. In The Proceedings of the International Conference on Information Processing and Management of Uncertainty (IPMU2008), Malaga, June 2008.

18. Audun Jøsang. Multi-Agent Preference Combination using Subjective Logic. In International Workshop on Preferences and Soft Constraints (Soft'11), Perugia, Italy, 2011.

19. G. Shafer. A Mathematical Theory of Evidence. Princeton University Press, 1976.

20. L.A. Zadeh. Review of Shafer's A Mathematical Theory of Evidence. AI Magazine, 5:81–83, 1984.

21. A. Jøsang and S. Pope. Dempster's Rule as Seen by Little Coloured Balls (in press). Computational Intelligence Journal, 2011. Available from http://folk.uio.no/Jøsang/.

22. A. Jøsang. A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9(3):279–311, June 2001.

23. Philippe Smets. Imperfect information: Imprecision and uncertainty. In Amihai Motro and Philippe Smets, editors, Uncertainty Management in Information Systems: From Needs to Solution, pages 189–224. Kluwer Academic Publishers, 1996.

24. A. M. Gupta Solo and M. M. Gupta. Perspectives on computational perception and cognition under uncertainty. In Proceedings of IEEE International Conference on Industrial Technology 2000, volume 2, pages 221– 224, Goa, India, January 2000. IEEE Computer Society.

25. L. A. Zadeh. Fuzzy sets. Information and Control, 8(3):338–353, June 1965.

26. J. M. Mendel. Uncertain rule-based fuzzy logic systems: Introduction and new directions. Prentice Hall, 2001.

27. Howard Raiffa. Decision Analysis. Addison-Wesley, Reading, MA, 1968

28. Detlof von Winterfeldt and Ward Edwards. Decision Analysis and Behavioral Research. Cambridge University Press, Cambridge, 1988.

29. Leonard J. Savage. The Foundations of Statistics. John Wiley & Sons, New York, NY, 1954.

30. Hela Ltifi, Ghada Trabelsi, Mounir Ben Ayed, Adel M. Alimi, "Dynamic Decision Support System Based on Bayesian Networks" in (IJARAI) International Journal of Advanced Research in Artificial Intelligence, Vol. 1, No. 1, 2012, pp 22-29

31. Shawn P. Curley, "The application of Dempster-Shafer theory demonstrated with justification provided by legal evidence" in Judgment and Decision Making, Vol. 2, No. 5, October 2007, pp. 257–276

32. Frada Burstein, Clyde W. Holsapple (2008) Handbook on Decision Support Systems 1. [Online], Springer Berlin Heidelberg. Available from http://link.springer.com/book/10.1007/978-3-540-48713-5/page/1

33. M. J. Druzdzel, "Probabilistic Reasoning in Decision Support Systems: From Computation to Common Sense" PhD thesis, Dept. Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA, 1993.

34. D. McAnally and A. Jøsang. Addition and Subtraction of Beliefs. In Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2004), Perugia, July 2004.

35. Audun Jøsang and Robin Hankin. Interpretation and Fusion of Hyper Opinions in Subjective Logic. 15th International Conference on Information Fusion (FUSION 2012). Singapore, July 2012 1225 - 1232. ISBN 978-0-9824438-4-2.

36. Jøsang, Audun (2011, March 24). Subjective Logic. Retrieved from

http://folk.uio.no/Jøsang/sl/Op.html

VITA AUCTORIS

Arshdeep Singh Sidhu was born in 1986 in Punjab, India. He received his B.Tech (Bachelor of Technology) degree in Computer Science & Technology from Punjab Technical University, India in 2008. After obtaining his undergraduate degree, he worked at Sebiz Infotech Ltd. (Netsmartz) as Senior Executive. Currently, he is a candidate for the Master's degree in Computer Science at the University of Windsor and hopes to graduate in early 2014.