Vague Sets or Intuitionistic Fuzzy Sets for Handling Vague Data: Which One Is Better?

An Lu and Wilfred Ng

Department of Computer Science
The Hong Kong University of Science and Technology
Hong Kong, China
{anlu, wilfred}@cs.ust.hk

Abstract. In the real world there are vaguely specified data values in many applications, such as sensor information. Fuzzy set theory has been proposed to handle such vagueness by generalizing the notion of membership in a set. Essentially, in a Fuzzy Set (FS) each element is associated with a point-value selected from the unit interval [0,1], which is termed the grade of membership in the set. A Vague Set (VS), as well as an Intuitionistic Fuzzy Set (IFS), is a further generalization of an FS. Instead of using point-based membership as in FSs, interval-based membership is used in a VS. The interval-based membership in VSs is more expressive in capturing vagueness of data. In the literature, the notions of IFSs and VSs are regarded as equivalent, in the sense that an IFS is isomorphic to a VS. Furthermore, due to such equivalence and IFSs being earlier known as a tradition, the interesting features for handling vague data that are unique to VSs are largely ignored. In this paper, we attempt to make a comparison between VSs and IFSs from various perspectives of algebraic properties, graphical representations and practical applications. We find that there are many interesting differences from a data modelling point of view. Incorporating the notion of VSs in relations, we describe Vague SQL (VSQL), which is an extension of SQL for the vague relational model, and show that VSQL combines the capabilities of a standard SQL with the power of manipulating vague relations. Although VSQL is a minimal extension to illustrate its usages, VSQL allows users to formulate a wide range of queries that occur in different modes of interaction between vague data and queries.

1 Introduction

Fuzzy set theory has long been introduced to handle inexact and imprecise data by Zadeh's seminal paper in [1], since in the real world there is vague information about different applications, such as in sensor databases, we can formalize the measurements from different sensors to a vague set. In fuzzy set theory, each object $u \in U$ is assigned a single real value, called the *grade of membership*, between zero and one. (Here U is a classical set of objects, called the *universe of discourse*.) In [2], Gau et al. point out that the drawback of using the single membership value in fuzzy set theory is that the evidence for $u \in U$ and the evidence against $u \in U$ are in fact mixed together. In order to tackle this problem, Gau et al. propose the notion of *Vague Sets* (VSs), which allow using interval-based membership instead of using point-based membership as in FSs. The interval-based membership generalization in VSs is more expressive in capturing vagueness of data. However, VSs are shown to be equivalent to that of *Intuitionistic Fuzzy Sets* (IFSs) [3–6] in [7]. For this reason, the interesting features for handling vague data that are unique to VSs are largely ignored.

In this paper, we attempt to make a more detailed comparison between VSs and IFSs from various perspectives of algebraic properties, graphical representations and practical applications. We find that there are many interesting features of VSs from a data modelling point of view. Essentially, due to the fact that a VS corresponds to a more intuitive graphical view of data sets, it is much easier to define and visualize the relationship of vague data objects. The classical nulls representing incompleteness can be viewed as a special case of a vague set and then generalized to vague data. In addition, we show that the notions of crisp and imprecision in vague data can be captured by interval relationships.

We further incorporate the notion of VSs in relations and describe Vague SQL (VSQL), which is an extension of SQL for the vague relational model. We show that VSQL combines the capabilities of standard SQL with the power of manipulating vague relations. We refine the notion of *degree of similar equality* (S_{EQ}) defined in [8] for comparing data values in order to process the vague selection predicates used in VSQL. Although VSQL is a minimal extension to illustrate its usages, VSQL allows the users to formulate a wide range of queries that occur in different modes of interaction between vague data and queries.

The main contributions of this paper are fourfold. First, we examine in more diversified ways, the notions of VSs and IFSs, which has so far not been done in the literature and leads to the undermining of the development of VSs. Second, we study the relationships between vague memberships and nulls in VSs and present the crisp and the imprecision orders. Third, we discuss the similarity measure between vague values and sets. Finally, we propose VSQL in order to gain more expressive power to formulate queries involving vague information. We classify the interactions between queries and data in four modes and demonstrate how the queries arising in different modes can be formulated using VSQL.

The rest of the paper is organized as follows. Section 2 presents some basic concepts related to VSs and IFSs. We also briefly discuss some ways of measuring vagueness in practice. In Section 3, we discuss the representations and the graphical view of nulls in VSs. In Section 4, we present the median and the imprecision membership of VSs and the crisp and the imprecision order in VSs. In Section 5, we discuss the similarity of vague values and sets. In Section 6, we propose VSQL, which is powerful enough to retrieve the data of a specified degree of vagueness. In Section 7 we offer our concluding remarks.

2 Vague Sets and Intuitionistic Fuzzy Sets

In this section, we introduce some basic concepts related to Vague Sets (VSs) and Intuitionistic Fuzzy Sets (IFSs). We illustrate that the graphical representation of VSs is more intuitive in perceiving vague values.

2.1 Basics

Let U be a classical set of objects, called the universe of discourse, where an element of U is denoted by u.

Definition 1. (Fuzzy Set) *A fuzzy set* $A = \{< u, \mu_A(u) > | u \in U\}$ *in a universe of discourse* U *is characterized by a membership function,* μ_A*, as follows:* $\mu_A : U \rightarrow [0, 1]$.

Definition 2. (Vague Set) *A vague set* V *in a universe of discourse* U *is characterized by a true membership function,* α_V*, and a false membership function,* β_V*, as follows:* $\alpha_V : U \rightarrow [0, 1], \beta_V : U \rightarrow [0, 1]$*, and* $\alpha_V(u) + \beta_V(u) \leq 1$*, where* $\alpha_V(u)$ *is a lower bound on the grade of membership of* u *derived from the evidence for* u*, and* $\beta_V(u)$ *is a lower bound on the grade of membership of the negation of* u *derived from the evidence against* u*.*

Suppose $U = \{u_1, u_2, \ldots, u_n\}$. A vague set V of the universe of discourse U can be represented by $V = \sum_{i=1}^{n} [\alpha(u_i), 1 - \beta(u_i)]/u_i$, where $0 \leq \alpha(u_i) \leq 1 - \beta(u_i) \leq 1$ and $1 \leq i \leq n$. In other words, the grade of membership of u_i is bounded to a subinterval $[\alpha_V(u_i), 1 - \beta_V(u_i)]$ of $[0, 1]$. Thus, VSs are a generalization of FSs, since the grade of membership $\mu_V(u)$ of u in Definition 1 may be inexact in a VS. The idea of membership generalization via an interval has actually proposed earlier as Intuitionistic Fuzzy Sets (IFSs) [3, 4] as follows:

Definition 3. (Intuitionistic Fuzzy Sets) *An intuitionistic fuzzy set* $A = \{< u, \mu_A(u), \nu_A(u) > | u \in U\}$ *in a universe of discourse* U *is characterized by a membership function,* μ_A*, and a non-membership function,* ν_A*, as follows:* $\mu_A : U \rightarrow [0, 1], \nu_A : U \rightarrow [0, 1]$*, and* $0 \leq \mu_A(u) + \nu_A(u) \leq 1$*.*

As we can see that the difference between VSs and IFSs is due to the definition of membership intervals. We have $[\alpha_V(u), 1 - \beta_V(u)]$ for u in V but $< \mu_A(u), \nu_A(u) >$ for u in A. Here the semantics of μ_A is the same as with α_V and ν_A is the same as with β_V. However, the boundary $(1 - \beta_V)$ is able to indicate the possible existence of a data value, as already mentioned in [7]. This subtle difference gives rise to a simpler but meaningful graphical view of data sets. We now depict a VS in Fig. 1 and an IFS in Fig. 2 respectively. It can be seen that, the shaded part formed by the boundary in a given VS in Fig. 1 naturally represents the *possible existence of data*. Thus, this “hesitation region” corresponds to the intuition of representing vague data.

We will see more benefits of using vague membership intervals in capturing data semantics in subsequent sections. The choice of the membership boundary also has interesting implications on modelling relationship between vague data.

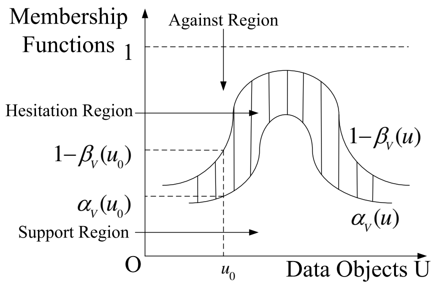

Fig. 1. Membership Functions of a VS

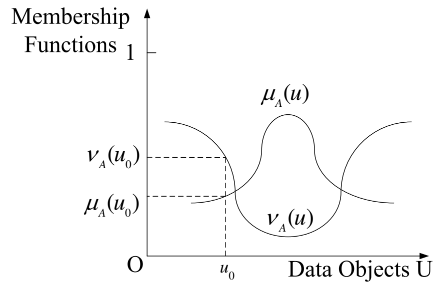

Fig. 2. Membership Functions of an IFS

2.2 Algebraic Operations

In this subsection, we present the basic operations of VSs and IFSs, which include complement, containment, equal, union, intersection, and so on. The details of most operations related to VSs can be consulted from [2].

Definition 4. (Complement) *The complement of a vague set V is denoted by V' and is defined by*

$$\alpha_{V'}(u) = \beta_V(u),$$
$$1 - \beta_{V'}(u) = 1 - \alpha_V(u).$$

Definition 5. (Containment) *A vague set V_A is contained in another vague set V_B, $V_A \subseteq V_B$, if and only if,*

$$\alpha_{V_A}(u) \leq \alpha_{V_B}(u),$$
$$1 - \beta_{V_A}(u) \leq 1 - \beta_{V_B}(u).$$

Definition 6. (Equal) *Two vague sets V_A and V_B are equal, written as $V_A = V_B$, if and only if, $V_A \subseteq V_B$ and $V_B \subseteq V_A$; that is*

$$\alpha_{V_A}(u) = \alpha_{V_B}(u),$$
$$1 - \beta_{V_A}(u) = 1 - \beta_{V_B}(u).$$

Definition 7. (Union) *The union of two vague sets V_A and V_B is a vague set V_C, written as $V_C = V_A \cup V_B$, whose true membership and false membership functions are related to those of V_A and V_B by*

$$\alpha_{V_C}(u) = max(\alpha_{V_A}(u), \alpha_{V_B}(u)),$$
$$1 - \beta_{V_C}(u) = max(1 - \beta_{V_A}(u), 1 - \beta_{V_B}(u)) = 1 - min(\beta_{V_A}(u), \beta_{V_B}(u)).$$

Definition 8. (Intersection) *The intersection of two vague sets V_A and V_B is a vague set V_C, written as $V_C = V_A \cap V_B$, whose true membership and false membership functions are related to those of V_A and V_B by*

$$\alpha_{V_C}(u) = min(\alpha_{V_A}(u), \alpha_{V_B}(u)),$$
$$1 - \beta_{V_C}(u) = min(1 - \beta_{V_A}(u), 1 - \beta_{V_B}(u)) = 1 - max(\beta_{V_A}(u), \beta_{V_B}(u)).$$

As a comparison, we present the counterpart operations for IFSs [3].

Definition 9. *If A and B are two IFSs of the set U, then*
$\bar{A} = \{< u, \nu_A(u), \mu_A(u) > | u \in U\}$,
$A \subseteq B$ *iff* $\forall u \in U, \mu_A(u) \leq \mu_B(u)$ *and* $\nu_A(u) \geq \nu_B(u)$,
$A = B$ *iff* $\forall u \in U, \mu_A(u) = \mu_B(u)$ *and* $\nu_A(u) = \nu_B(u)$,
$A \cup B = \{< u, max(\mu_A(u), \mu_B(u)), min(\nu_A(u), \nu_B(u)) > | u \in U\}$, *and*
$A \cap B = \{< u, min(\mu_A(u), \mu_B(u)), max(\nu_A(u), \nu_B(u)) > | u \in U\}$.

2.3 Measurements of Vagueness in Practice

We now discuss some ideas of how to measure memberships of VSs and IFSs. There is actually no consensus in the interpretation of what a membership grade means in the literature [9].

In [10], Bilgiç and Türkşen present a review of various interpretations of the fuzzy membership function and the ways of obtaining a membership function. VSs also share similar interpretation of membership grades. For example, the vague predicate "John is tall" is given by an interval in the unit interval, [0.6,0.8]. There are several possible views on how to measure the membership:

Likelihood view: 60-80% of a given population declares that John is tall.

Random set view: 60-80% of a given population describes "tall" as an interval containing John's height.

Similarity view: John's height is away from the *benchmark* object which is truly "tall" to the degree 0.2-0.4. Here if we assume a benchmark example of "tall" is 250cm with the full degree [1,1], then John's height is away from 250cm to the degree 0.2-0.4 means his height is between $(1 - 0.4) \times 250$ and $(1 - 0.2) \times 250$cm, that is, 150-200cm.

For IFSs, we may have the following interpretations:

Likelihood view: 60% of a given population declares that John is tall while 20% does not. (Another 20% is neutral.)

Random set view: 60% of a given population describes "tall" as an interval containing John's height while 20% does not. (Another 20% is neutral.)

Similarity view: The same as in the VS case.

The following is a more detailed example which helps to understand the collection of vague data, as well as IFS data.

Example 1. In a sensor database application, suppose in a testing region we have a set of ten sensors $\{S_1, S_2, \ldots, S_{10}\}$. We then obtain ten corresponding measurements, {20, 22, 20, 21, 20, -, 20, 20, -, 20} at a certain time t. Here "-" means that the sensor data is not reachable/accessible at time t. (i.e. we have six 20, one 21, one 22 and two missing values.) Now, we formalize the results to a vague set V_t as follows. There are six occurrences of 20, but two values (21 and 22) are against it. There are also two missing values (neutral), thus the true membership α is 0.6 and the false membership β is 0.2 (i.e. 1-$\beta = 0.8$). Thus, we obtain the vague membership value [0.6,0.8] for 20. Similarly, we obtain the vague membership value [0.1,0.3] for 21 and [0.1,0.3] for 22. Combining these results, we have the VS, V_t = [0.6,0.8]/20+[0.1,0.3]/21+[0.1,0.3]/22. Equivalently, we have the IFS, $A_t = \{< 20, 0.6, 0.2 >, < 21, 0.1, 0.7 >, < 22, 0.1, 0.7 >\}$.

The above example also indicates that, using a VS is more natural than an IFS for merging fuzzy objects. For example, suppose we merge three fuzzy values $0.4/u$, $0.5/u$ and $0.6/u$. We can then directly obtain the vague value $[0.4, 0.6]/u$, which means that the lower bound of the membership of u is the minimum of the fuzzy membership, 0.4, and that the upper bound is the maximum of the fuzzy membership, 0.6. However, by using the intuitionistic fuzzy value we have $< u, 0.4, 0.4 >$, which is much less intuitive.

3 Relationships between VS Memberships and Nulls

In this section, we discuss how VSs capture different notions of incompleteness. We need to define an empty vague set first.

Definition 10. (Empty Vague Set) *A vague set V is an empty vague set, if and only if, its true membership function* $\alpha = 0$ *and false membership function* $\beta = 1$ *for all u. We use* $\emptyset_V$ *to denote it.*

It is worth mentioning that $\emptyset_V$ can be regarded as the generalization of the empty set in the ordinary set theory, which is not the same empty concept as defined in [2]. In [2], a vague set V is empty, if and only if, its true membership function α and false membership function β are both 0, which means that we have no information about whether the corresponding object belongs to the vague set or not. However, our definition of empty vague set means that we know exactly that no object belongs to the empty vague set. Furthermore, we define *empty vague value*, or simply *empty value*, as $[0, 0]$ (i.e. $\alpha = 0$, $1 - \beta = 0$).

We first review the three kinds of classical null values and the crisp value, which are represented in Fig. 3 and then generalize them to the vague domain as shown in Fig. 4.

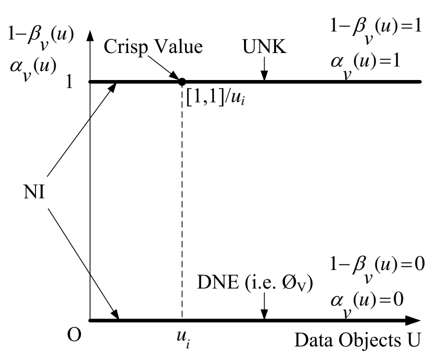

Fig. 3. Classical Nulls in a VS

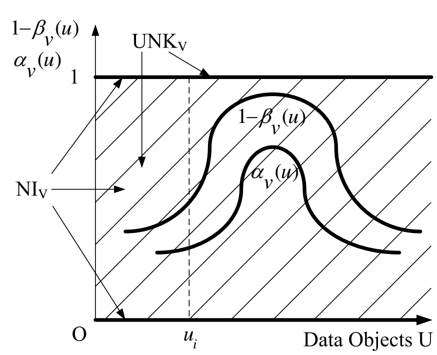

Fig. 4. Generalized Nulls in a VS

1. Unknown (UNK) represents that the value (a classical data object) exists but is unknown at the represent time. In crisp sets, all memberships of objects are assumed as [1,1], which can be regarded as a special case of VSs. For example, it may not be known at the present time the AGE of an EMPLOYEE in an employee relation. We can view UNK in the form of a VS by $V = \sum_{i=1}^{n} [1, 1]/u_i$, which means that each object $u_i \in U$ $(1 \leq i \leq n)$ may "totally" belong to the vague set V. The UNK is represented as the horizontal lines such that $\alpha_V(u_i) = 1 - \beta_V(u_i) = 1$ in Fig. 3.
2. Does Not Exist (DNE) presents that "the value is inapplicable". For example, if Tom has not married, the SpouseName of Tom in an employee relation is denoted as DNE. We can view DNE in the form of a VS, V, as given by $V = \sum_{i=1}^{n} [0, 0]/u_i$ $(1 \leq i \leq n)$, which means that we are sure that u_i does not belong to the vague set and the evidence is totally against it, that is, $\alpha_V(u_i) = 0$ but $\beta_V(u_i) = 1$. Thus, we obtain the empty vague set $\emptyset_V$. The DNE is represented as the horizontal line as shown in Fig. 3.
3. No Information (NI) represents that "no information is available for the values", i.e. it is either UNK or DNE. For example, it is not known if an EMPLOYEE has got married or not when there is no data in STATUS. The NI is represented as horizontal lines of UNK and DNE in Fig. 3.

The above null values are limited to the crisp sets which contain classical data objects. We now generalize the notions of UNK and NI to VSs as follows. Note that such a generalization has no effect on DNE which is still the U-axis.

1. Generalized UNK (UNK_V) represents that memberships can be vague values but are unknown at the represent time. We can view UNK_V as any possible VS such that, $\forall u_i \in U$, $\alpha_V(u_i) \in (0, 1]$ and $\beta_V(u_i) \in (0, 1]$, as shown in Fig. 4, i.e. the shaded area excluding the DNE line (excluding the U-axis) but including the UNK line.
2. Generalized DNE (DNE_V) is the same as DNE.
3. Generalized NI (NI_V) is the region of NI_V which includes the DNE line and the UNK_V region as shown in Fig. 4.

Based on the above discussion, we use Fig. 5 to illustrate the relationships between various cases of nulls discussed so far.

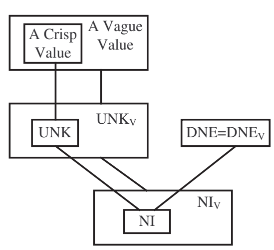

Fig. 5. Relationships between Various Cases of Nulls

4 Relationships between VS Memberships

In this section, we discuss the following relationships of vague membership values (vague values for short) in VSs: crisp and imprecision. Remarkably, there are no such meaningful relationships based on IFS membership values. We also show two lattices arising from the crisp and the imprecision orders.

In order to compare vague values, we need to introduce two derived memberships for discussion.

The first is called the *median membership*, $M_m = (\alpha + 1 - \beta)/2$, which represents the overall evidence contained in a vague value and is shown in Fig. 6. It can be checked that $0 \leq (\alpha + 1 - \beta)/2 \leq 1$. In addition, the vague value [1,1] has the highest M_m, which means the corresponding object totally belongs to the VS (i.e. a crisp value). While the vague value [0,0] has the lowest M_m which means that the corresponding object totally does not belong to the VS (i.e. the empty vague value).

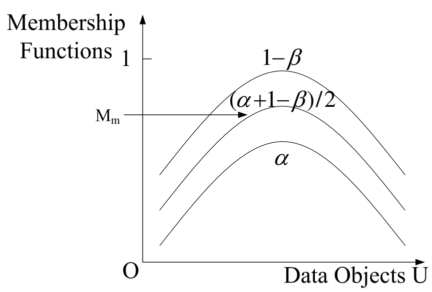

Fig. 6. Median Membership of a VS

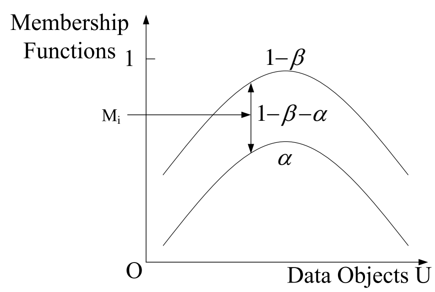

Fig. 7. Imprecision Membership of a VS

The second is called the *imprecision membership*, $M_i = (1 - \beta - \alpha)$, which represents the overall imprecision of a vague value and is shown in Fig. 7. It can be checked that $0 \leq (1 - \beta - \alpha) \leq 1$. In addition, the vague value $[a, a] (a \in [0, 1])$ has the lowest M_i which means that we know exactly the membership of the corresponding object (i.e. a fuzzy value). While the vague value [0,1] has the highest M_i which means that we know nothing about the membership of the corresponding object.

Suppose x and y are two vague values defined for a certain object u such that $x = [\alpha_x, 1 - \beta_x]$, $y = [\alpha_y, 1 - \beta_y]$. We then have the following crisp order and the imprecision order relating the x and y membership intervals, which are analogous to the "degree of truth" and the "amount of knowledge" in [11].

Definition 11. (Crisp Order and Imprecision Order) *Let x and y are two vague values defined for a certain object u. We say x is less crisp than y, denoted as $x \leq_c y$, if $\alpha_x \leq \alpha_y$ and $1 - \beta_x \leq 1 - \beta_y$. We say x is less imprecise than y, denoted as $x \leq_i y$, if $\alpha_x \geq \alpha_y$ and $1 - \beta_x \leq 1 - \beta_y$.*

It is straightforward to check that $\leq_i$ and $\leq_c$ are two orthogonal concepts. For example, [0.7, 0.9] $\leq_c$ [0.8, 0.9] but [0.7, 0.9] $\not\leq_i$ [0.8, 0.9]. On the other hand, [0.7, 0.9] $\leq_i$ [0.6, 0.9] but [0.7, 0.9] $\not\leq_c$ [0.6, 0.9]. The other interesting relationships between the crisp order and the imprecision order, and the three interval relationships of precedence, overlap and contain are depicted in Fig. 8.

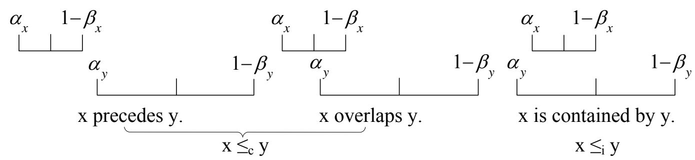

Fig. 8. Interaction between Crisp and Imprecision Order, and Precedence, Overlap and Contain Relationships

The relationship between the crisp and imprecision order, and the median and the imprecision memberships is as follows.

1. If $x \leq_c y$, then $M_m(x) \leq M_m(y)$, but not vice versa.
2. If $x \leq_i y$, then $M_i(x) \leq M_i(y)$, but not vice versa.

We now use $\wedge$ and $\vee$ for c-meet and c-join under $\leq_c$, and $\otimes$ and $\oplus$ for i-meet and i-join under $\leq_i$. We define $[\alpha_x, 1-\beta_x] \wedge [\alpha_y, 1-\beta_y] = [min\{\alpha_x, \alpha_y\}, min\{1-\beta_x, 1-\beta_y\}]$ and $[\alpha_x, 1-\beta_x] \vee [\alpha_y, 1-\beta_y] = [max\{\alpha_x, \alpha_y\}, max\{1-\beta_x, 1-\beta_y\}]$. We define $[\alpha_x, 1-\beta_x] \otimes [\alpha_y, 1-\beta_y] = [max\{\alpha_x, \alpha_y\}, min\{1-\beta_x, 1-\beta_y\}]$ and $[\alpha_x, 1-\beta_x] \oplus [\alpha_y, 1-\beta_y] = [min\{\alpha_x, \alpha_y\}, max\{1-\beta_x, 1-\beta_y\}]$. It is easy to check that the crisp order $\leq_c$ induces a complete lattice by using $\wedge$ and $\vee$. Under the crisp order, [0,0] is the bottom ($\perp_c$), and [1,1] is the top ($\top_c$). On the other hand, the imprecision order $\leq_i$ induces a complete semi-lattice by using $\otimes$ and $\oplus$. It should be noted that i-meet is *not defined* when $max\{\alpha_x, \alpha_y\} > min\{1-\beta_x, 1-\beta_y\}$, since the meet result is not a valid interval. From now on, we restrict our discussion to the i-meet that gives rise to valid intervals as a result. Under the imprecision order, [0,1] is the top ($\top_i$), but there is no bottom for the semi-lattice.

Theorem 1. *The following statements are true.*

1. *If $x \leq_i y$, then*
$$\begin{cases} x \leq_i (x \wedge y) \leq_i y; \\ x \leq_i (x \vee y) \leq_i y. \end{cases} \quad \begin{cases} (x \wedge y) \leq_c x \leq_c (x \vee y); \\ (x \wedge y) \leq_c y \leq_c (x \vee y). \end{cases} \quad \begin{cases} x \otimes y = x; \\ x \oplus y = y. \end{cases}$$
2. *If $x \leq_c y$, then*
$$\begin{cases} x \leq_c (x \otimes y) \leq_c y; \\ x \leq_c (x \oplus y) \leq_c y. \end{cases} \quad \begin{cases} (x \otimes y) \leq_i x \leq_i (x \oplus y); \\ (x \otimes y) \leq_i y \leq_i (x \oplus y). \end{cases} \quad \begin{cases} x \wedge y = x; \\ x \vee y = y. \end{cases}$$

As an example, we discretize [0,1] to the granularity of 0.1 unit for α and $1-\beta$ and show in Fig. 9 a complete lattice induced by the crisp order, which is along the crisp dimension, and a complete semi-lattice induced by the imprecision order, which is along the imprecision dimension. The distance between two adjoining elements is 0.1. For example, it can be checked that [0.3,0.4] $\wedge$ [0.1,0.7] = [0.1,0.4], and [0.3,0.4] $\vee$ [0.1,0.7] = [0.3,0.7] in the crisp dimension. It can also be checked from Fig. 9 in the imprecision dimension that, [0,0.6] $\otimes$ [0.4,0.8] = [0.4,0.6], and [0,0.6] $\oplus$ [0.4,0.8] = [0,0.8]. However, [0,0.1] $\otimes$ [0.2,0.3] is undefined, and there is no Greatest Lower Bound (GLB) for these two vague values in the imprecision lattice. The lattice size is exponential to the square of the discretization on the unit membership interval.

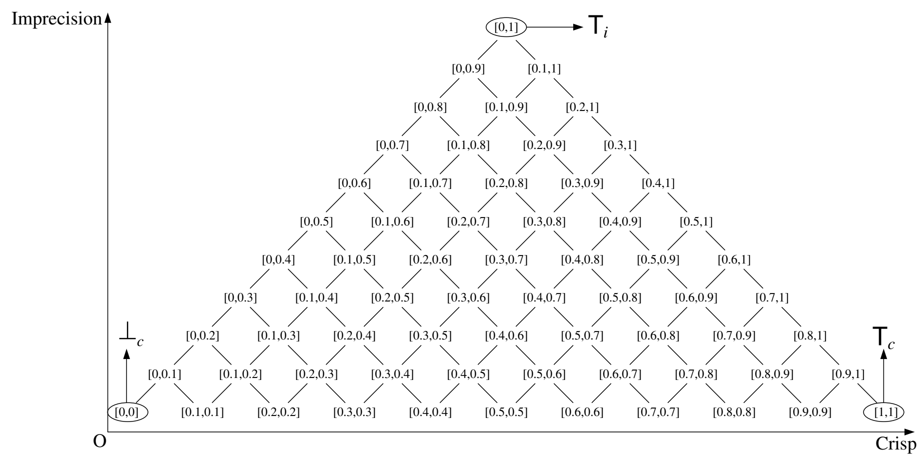

Fig. 9. Complete Imprecision Semi-Lattice and Complete Crisp Lattice

5 Similarity Measures of VSs

In this section, we discuss a similarity measure between two VSs, which is based on the median membership and the imprecision membership.

We now let x and y be two vague values to some $u \in U$ such that $x = [\alpha_x, 1-\beta_x]$, $y = [\alpha_y, 1-\beta_y]$. We define ΔM_m as the difference between median memberships, which is given by $\Delta M_m = |(\alpha_x + 1 - \beta_x) - (\alpha_y + 1 - \beta_y)|/2 = |(\alpha_x - \alpha_y) - (\beta_x - \beta_y)|/2$, such that $0 \leq \Delta M_m \leq 1$. We define ΔM_i as the difference between imprecision memberships, which is given by $\Delta M_i = |(1 - \beta_x - \alpha_x) - (1 - \beta_y - \alpha_y)| = |(\alpha_x - \alpha_y) + (\beta_x - \beta_y)|$, such that $0 \leq \Delta M_i \leq 1$. We define the similarity measure between two vague values x and y as follows:

Definition 12. (Similarity Measure between Two Vague Values)

$$
\begin{aligned}
M(x,y) &= \sqrt{(1-\Delta M_m)(1-\Delta M_i)} \\
&= \sqrt{(1 - \frac{|(\alpha_x - \alpha_y) - (\beta_x - \beta_y)|}{2})(1 - |(\alpha_x - \alpha_y) + (\beta_x - \beta_y)|)}.
\end{aligned}
$$

We extend the similarity measure from two vague values to two vague sets.

Definition 13. (Similarity Measure between Two Vague Sets) *Let X and Y be two vague sets, where*

$$X = \sum_{k=1}^{n} [\alpha_X(u_k), 1-\beta_X(u_k)]/u_k; \; Y = \sum_{k=1}^{n} [\alpha_Y(u_k), 1-\beta_Y(u_k)]/u_k.$$

The similarity measure between the vague sets X and Y can be evaluated as follows:

$$\begin{aligned} M(X,Y) &= \frac{1}{n}\sum_{k=1}^{n} M([\alpha_X(u_k), 1-\beta_X(u_k)], [\alpha_Y(u_k), 1-\beta_Y(u_k)]) \\ &= \frac{1}{n}\sum_{k=1}^{n} \sqrt{(1-\Delta M_{m,k})(1-\Delta M_{i,k})} \\ &= \frac{1}{n}\sum_{k=1}^{n} \sqrt{(1-\frac{|(\alpha_X(u_k)-\alpha_Y(u_k))-(\beta_X(u_k)-\beta_Y(u_k))|}{2})} \cdot \\ &\quad \sqrt{(1-|(\alpha_X(u_k)-\alpha_Y(u_k))+(\beta_X(u_k)-\beta_Y(u_k))|)}. \end{aligned}$$

The following theorem illustrates some good features for similarity measure between vague sets, which follows from Definition 13.

Theorem 2. *Let X and Y be two vague sets. The following statements are true.*

1. *The similarity measure is bounded, i.e.,* $0 \le M(X,Y) \le 1$.
2. $M(X,Y) = 1$ *if and only if* $X = Y$.
3. $M(X,Y) = 0$, *if and only if, (i) a vague value in X is $[0,0]$ then the corresponding vague value in Y is $[1,1]$, or (ii) a vague value in X is $[0,1]$ then the corresponding vague value in Y is $[a,a]$, where $0 \le a \le 1$.*
4. *The similarity measure is commutative, i.e.,* $M(X,Y) = M(Y,X)$.

6 Vague Relations and VSQL

In this section, we propose VSQL in order to gain more expressive power to formulate queries involving vague information. We classify the interactions between queries and data in four modes and demonstrate how the queries arising in different modes can be formulated using VSQL.

6.1 Vague Relations

Definition 14. (Vague Relation) *Let $\mathcal{U} = \{U_1, \ldots, U_m\}$ be a collection of universes of discourse. Let $Dom(A_i)$ be the domain corresponding to the attribute A_i. We define $Dom(A_i) = \{V \mid V \text{ is a } VS \text{ of } U_i\}$. A vague tuple $t = (a_1, a_2, \ldots, a_m)$ over a relation scheme, $R = \{A_1, A_2, \ldots, A_m\}$, is an element in $Dom(A_1) \times Dom(A_2) \times \cdots \times Dom(A_m)$. A vague relation r over R is a subset of $Dom(A_1) \times Dom(A_2) \times \cdots \times Dom(A_m)$.*

Unlike classical and fuzzy relations, in vague relations, $Dom(A_i)$ is a set of vague sets. Vague relations can be considered as an extension of classical relations (all vague values are [1, 1]) and fuzzy relations (all vague values are $[a, a]$, $0 \leq a \leq 1$), which can capture more information about vagueness.

Consider the vague relation r over Product(ID, Weight, Price) given in Table 1. In r, Weight and Price are vague attributes. The attribute ID is crisp, where values are presented as the usual atomic values. The first tuple in r means the product with ID=1 has the weight of $[1, 1]/10$ and the price of $[0.4, 0.6]/50 + [1, 1]/80$.

Table 1. A Vague Relation r

ID	**Weight**	**Price**
1	[1,1]/10	[0.4,0.6]/50+[1,1]/80
2	[1,1]/20	[0.8,0.9]/100+[0.6,0.8]/150
3	[1,1]/20	[1,1]/100+[0.7,0.9]/150
4	[1,1]/10+[0.6,0.8]/15	[1,1]/80+[0.6,0.8]/100
5	[0.6,0.8]/10+[1,1]/15+[0.6,0.8]/20	[0.6,0.8]/80+[1,1]/100

6.2 Similar Equality

In this subsection, we extend the notion of *similar equality* (S_{EQ}) defined in [8] for comparing data values to deal with selection predicates in VSQL.

The *degree of similar equality* (S_{EQ}) of vague relations defined below can be used as a vague similarity measure to compare elements of a given domain. Suppose t_p and t_q are two tuples in a relation r.

Definition 15. (Degree of Similar Equality) *The degree of similar equality of two vague tuples t_p and t_q on the attribute A_i in a vague relation is given by:* $S_{EQ}(t_p[A_i], t_q[A_i]) = M(t_p[A_i], t_q[A_i])$.

The degree of similar equality of two vague tuples t_p and t_q on attributes $X = \{A_1, \ldots, A_s\}$ ($X \subseteq R$) in a vague relation is $S_{EQ}(t_p[X], t_q[X]) = min\{S_{EQ}(t_p[A_1], t_q[A_1]), \ldots, S_{EQ}(t_p[A_s], t_q[A_s])\}$.

6.3 VSQL

There have been some studies which discuss the topic concerning fuzzy SQL queries in fuzzy databases [12, 13] which only cater for true membership (or we can say they combine true membership and false membership together). We now describe the extensions of VSQL to standard SQL. The VSQL is powerful enough to retrieve any set of items of any degree of vagueness.

Data Definition Language. The syntax of VSQL allows users to define semantic domains using the CREATE DOMAIN command as follows:

CREATE DOMAIN <domain name> <data types>.

The command is similar to the SQL standard statement that declares a domain. In Vague SQL DDL, attributes constraints for vague data and vague data definitions are added to the standard DDL. By using the keyword VAGUE, the attributes which store vague data are specified. For example, we define a vague domain SCALE as follows:

CREATE DOMAIN SCALE VAGUE INTEGER.

Vague Data Definition. Vague data to be used in VSQL DML are defined in VSQL DDL. Vague data are defined by specifying the table and attribute in which they are used. As an example, we define the vague data for the attribute "Weight" in Table 1.

CREATE VAGUE DATA ON SCALE(4) AS light = [1,1]/0 + [0.8,1]/10 + [0.7,0.9]/20 + [0.6,0.8]/30 + [0.5,0.7]/40 + [0.4,0.6]/50 + [0.3,0.5]/60 + [0.2,0.4]/70 + [0.1,0.3]/80 + [0,0.2]/90 + [0,0]/100.

CREATE VAGUE DATA ON SCALE(4) AS middle = [0,0]/0 + [0.1,0.3]/10 + [0.3,0.5]/20 + [0.5,0.7]/30 + [0.7,0.9]/40 + [1,1]/50 + [0.7,0.9]/60 + [0.5,0.7]/70 + [0.3,0.5]/80 + [0.1,0.3]/90 + [0,0]/100.

CREATE VAGUE DATA ON SCALE(4) AS heavy = [0,0]/0 + [0,0.2]/10 + [0.1,0.3]/20 + [0.2,0.4]/30 + [0.3,0.5]/40 + [0.4,0.6]/50 + [0.5,0.7]/60 + [0.6,0.8]/70 + [0.7,0.9]/80 + [0.8,1]/90 + [1,1]/100.

Table Definition. An example of Vague SQL DDL is shown as follows. Here SCALE(4) means the value of "Weight" is a 4-digit vague integer.

CREATE TABLE Product (ID INTEGER(8) NOT NULL, Name CHAR(16) NOT NULL, Weight SCALE(4), Size SCALE(8), Price SCALE(8)).

Data Manipulation Language. The expression of a basic VSQL query is given as follows:

SELECT <lists of attributes> [ANY|ALL] [ASC|DESC] FROM <lists of vague relations> WHERE <extended predicates>.

An attribute list is a list of attributes similar to the usual one, except that it provides us with an option that an attribute can be associated with a vague domain.

In Vague SQL DML, vague data can be used for many kinds of operator such as predicates, and the like. An example of Vague SQL DML is shown as follows.

Query: "Find the products which are heavy."

VSQL: SELECT ID, Weight, Price FROM Product WHERE Weight=heavy.

Here "heavy" is a vague set as mentioned above. Following the FROM keyword is a comma separated list of all relations used in a query. A typical form of a semantic comparison is: <attribute> <comparator> <attribute>.

6.4 VSQL Modes

We consider that data vagueness can occur in both relations and query expressions. Thus, we develop VSQL and allow users to formulate a wide range of vague queries that occur in different modes of interaction between the data and the queries. We classify our VSQL Modes as shown in Fig. 10. We now show the power of VSQL in formulating queries in these modes.

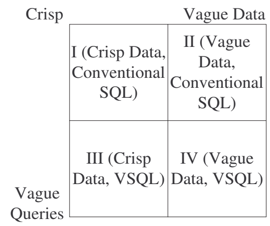

Fig. 10. The four modes of VSQL

Mode I (Crisp Data, Conventional SQL) The first mode concerns only conventional region of SQL usages, where data values and VSQL are both crisp (i.e. no vague data involved). This mode is no different from the classical relational databases and therefore VSQL is downward compatible to SQL. For example, Table 2 represents a classical relational table in which all data are crisp. The query (Q_1) falls into this region.

(Q_1) "Find the products which are equal to 20 kg."

SELECT * FROM Product WHERE Weight = 20.

Thus, we obtain the answers given in Tables 4.

Table 2. A Crisp Product Relation r_1

ID	**Weight**	**Price**
1	10	50
2	20	100
3	20	150
4	50	200
5	80	350

Table 3. A Vague Product Relation r_2

ID	**Weight**	**Price**
1	light	[1,1]/50
2	light	[0.6,0.8]/100
3	[1,1]/20 + [0.5,0.6]/50	[0.5,0.9]/150
4	middle	[0.8,0.9]/200
5	heavy	[0.7,1]/350

Table 4. Answer for (Q_1)

ID	**Weight**	**Price**
2	20	100
3	20	150

Table 5. Answer for (Q_2)

ID	**Weight**	**Price**	*Rank*
3	[1,1]/20 + [0.5,0.6]/50	[0.5,0.9]/150	0.967
1	light	[1,1]/50	0.624
2	light	[0.6,0.8]/100	0.624
4	middle	[0.8,0.9]/200	0.617
5	heavy	[0.7,1]/350	0.551

Mode II (Vague Data, Conventional SQL) The second mode concerns the scenario that data values are vague but queries are conventional. We allow classical SQL statements referencing vague tables to be formulated. For example, Table 3 represents a vague relational table where Weight and Price data are vague. (The vague data "light", "middle" and "heavy" are defined in Section 6.3.) We now formulate the query (Q_2) as below.

(Q_2) "Find the products which are equal to 20 kg."

Table 6. Answer for (Q_3)

ID	Weight	Price	*Rank*
5	80	350	0.624
4	50	200	0.587
2	20	100	0.551
3	20	150	0.551
1	10	50	0.536

Table 7. Answer for (Q_4)

ID	Weight	Price	*Rank*
5	heavy	[0.7,1]/350	1
4	middle	[0.8,0.9]/200	0.750
1	light	[1,1]/50	0.591
2	light	[0.6,0.8]/100	0.591
3	[1,1]/20 + [0.5,0.6]/50	[0.5,0.9]/150	0.578

SELECT * FROM Product WHERE Weight = 20.

We first transform 20 into the VS [1,1]/20, and then determine the S_{EQ} between the "Weight" values in r_2 (also in the form of VSs) and [1,1]/20. For example, for the tuple with ID = 4 in r_2 given in Table 3, according to Definition 15, we obtain $S_{EQ}(t_4[Weight], [1,1]/20) = S_{EQ}(middle, [1,1]/20) = 0.617$. Then we rank the tuples by this S_{EQ} value and obtain the answer given in Table 5.

Mode III (Crisp Data, Vague SQL) The third mode concerns the scenario that data values are crisp but SQL is vague. For example, we have the query (Q_3) in Table 2 as follows.

(Q_3) "Find the products which are heavy in weight."

SELECT * FROM Product WHERE Weight = heavy.

We first transform the "Weight" values in r_1 into VSs, and then determine the S_{EQ} between the "Weight" values and the VS "heavy". For example, for the tuple with ID = 2 in r_1 given in Table 2, we obtain $S_{EQ}(t_2[Weight], heavy) = S_{EQ}([1,1]/20, heavy) = 0.551$. Then we rank the tuples by this S_{EQ} value and obtain the answer given in Table 6.

Mode IV (Vague Data, Vague SQL) In the fourth mode, both data values and SQL are vague. We have the query (Q_4) as follows.

(Q_4) "Find the products which are heavy in weight."

SELECT * FROM Product WHERE Weight = heavy.

We first determine the S_{EQ} between the "Weight" values in r_2 and the VS "heavy". For example, for the tuple with ID = 4 in r_2, we obtain $S_{EQ}(t_4[Weight], heavy) = S_{EQ}(middle, heavy) = 0.750$. Then we rank the tuples by this S_{EQ} value and obtain the answer given in Table 7.

7 Conclusions

We have examined the two known generalizations, VSs and IFSs, of FSs. VSs are based on an interval-based membership and thus more expressive in capturing vagueness of data and the notions of IFSs and VSs are regarded as equivalent, in the sense that an IFS is isomorphic to a VS. We compare VSs and IFSs by their notions, algebraic properties, practical applications, and most importantly, the graphical representations of vague data objects.

Although VSs and IFSs are equivalent by basic definition, we show throughout the paper VSs allow for a more intuitive graphical representation of vague data, which facilitates significantly better analysis in data relationships, incompleteness, and similarity measures. When measuring vagueness in practice, we have showed by example that using a VS is more natural than using an IFS, especially for merging fuzzy objects. Using the interval memberships defined in VSs, we study the interactions of vague membership values and present the interesting relationships of crisp order and imprecision order. We also discuss the similarity measures between vague data.

In order to retrieve vague data by the well-established database query language, we incorporate the notion of vagueness into the relational data model and demonstrate how VSQL can be employed to formulate a wide range of queries arising from four modes of query/data interaction. We are still investigating optimization and processing techniques for vague queries. This also relies on the development of efficient indexing schemes for vague data.

References

1. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–353
2. Gao, W.L., Danied, J.B.: Vague sets. IEEE Transactions on Systems, Man, and Cybernetics **23** (1993) 610–614
3. Atanassov, K.T.: Intuitionistic fuzzy sets. Fuzzy Sets and Systems **20** (1986) 87–96
4. Atanassov, K.T.: More on intuitionistic fuzzy sets. Fuzzy Sets and Systems **33** (1989) 37–45
5. Atanassov, K.T., Gargov, G.: Interval-valued intuitionistic fuzzy sets. Fuzzy Sets and Systems **31** (1989) 343–349
6. Atanassov, K.T.: Intuitionistic Fuzzy Sets: Theory and Applications (Studies in Fuzziness and Soft Computing). Springer-Verlag Telos (1999)
7. Bustince, H., Burillo, P.: Vague sets are intuitionistic fuzzy sets. Fuzzy Sets and Systems **79** (1996) 403–405
8. Lu, A., Ng, W.: Managing merged data by vague functional dependencies. In Atzeni, P., Chu, W.W., Lu, H., Zhou, S., Ling, T.W., eds.: ER. Volume 3288 of Lecture Notes in Computer Science., Springer (2004) 259–272
9. Dubois, D., Ostasiewicz, W., Prade, H.: Fuzzy sets: History and basic notions. In Dubois, D., Prade, H., eds.: Fundamentals of Fuzzy Sets, the Handbooks of Fuzzy Sets Series. Kluwer Academic Publishers, Boston, MA (2000) 195–230
10. Bilgiç, T., Türkşen, I.B.: Measurement of membership functions: Theoretical and experimental work. In Dubois, D., Prade, H., eds.: Fundamentals of Fuzzy Sets, the Handbooks of Fuzzy Sets Series. Kluwer Academic Publishers, Boston, MA (2000) 195–230
11. Fitting, M.: Kleene's logic, generalized. J. Log. Comput. **1** (1991) 797–810
12. Bosc, P., Pivert, O.: Sqlf: A relational database language for fuzzy querying. IEEE Transactions on Fuzzy Systems **3** (1995) 1–17
13. Nakajima, H., Sogoh, T., Arao, M.: Fuzzy database language and library - fuzzy extension to sql. In: Proc. Second IEEE Int. Conf. on Fuzzy Systems. (1993) 477–482