Information Tables with Neighborhood Semantics

Y.Y. Yao

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2

ABSTRACT

Information tables provide a convenient and useful tool for representing a set of objects using a group of attributes. This notion is enriched by introducing neighborhood systems on attribute values. The neighborhood systems represent the semantics relationships between, and knowledge about, attribute values. With added semantics, neighborhood based information tables may provide a more general framework for knowledge discovery, data mining, and information retrieval.

Keywords: Information tables, neighborhood systems, relationships between attribute values, semantic closeness

1. INTRODUCTION

In many information processing systems, a set of objects are typically represented by their values on a finite set of attributes. Such information may be conveniently described in a tabular form. Each column corresponds to an attribute and each row corresponds to an object. The cell, defined by a pair of object and attribute, gives the value of that object on the attribute. The table represents all available information and knowledge about the objects under consideration. In other words, objects are perceived, observed, or measured based only on a finite number of properties. Similar representation approaches have been used either implicitly or explicitly in many fields, such as decision theory, pattern recognition, machine learning, data analysis, data mining, cluster analysis, databases, and information retrieval. Pawlak suggested using information systems to label this tabular knowledge representation approach.[15] To avoid confusion with the commonly associated meaning of "information systems", we prefer using information tables.

With respect to the notion of information tables, there are extensive studies on the relationships between values of *different* attributes and relationships between values of the *same* attribute. Studies of the two kinds of relationships correspond to the *horizontal* analysis and the *vertical* analysis of an information table. It is also possible to combine horizontal and vertical analysis.

Analysis of the horizontal relationships reveals the similarity, association, and dependency of different attributes.[31] Such relationships are normally characterized by the problem of determining the values of one set of attributes based on the values of another set of attributes. Two levels of dependencies, referred to as the *local* and *global* dependencies, may be observed. The local dependencies show how *one* specific combination of values on one set of attributes determines *one* specific combination of values on another set of attributes. The global dependencies show *all* combinations of values on one set of attributes determine *all* combinations of values on another set of attributes. Finding local dependencies is one of the main tasks of machine learning and data mining.[13,17] For instance, the well known association rules, which state the presence of one set of items implies the presence of another set of items, may be considered as a special kind of local dependencies. Functional dependency in relational databases is a typical example of global dependency.[1,4] Attribute (data) dependency studied in the theory of rough sets is another example of global dependency.[17] There are differences between functional dependency in relational database and data dependency in rough set theory. The functional dependency states the semantics constraints on objects in taking their attribute values. The data dependency summarizes the dependency of attribute with respect to a particular information table.

Analysis of vertical relationships deals with semantic closeness of values of an attribute. Examples of vertical analysis include the discretization of real-valued attributes, and the use of binary relations, order relations, concept hierarchies, fuzzy binary relations, similarity measures or distance functions on attribute values.[5–7,21,30,32] Using the vertical relationships between attribute values, one may study relationships between objects. Objects may be

E-mail: yyao@cs.uregina.ca, WWW: http://www.cs.uregina.ca/∼yyao

clustered and classified based on their attribute values. The semantic closeness of attribute values also offers a basis for approximate retrieval.[25]

The horizontal and vertical analyses of information tables focus on different aspects of an information table. In comparison, less work has been done on the vertical analysis. There are systematic studies on, and formal framework of, the horizontal analysis, such as the investigation of functional dependency in database, the study of probabilistic dependency (independency) in Bayesian networks, and the study of decision and association rules in machine learning and data mining. In many horizontal analysis methods, the vertical semantics are not incorporated. Different values of the same attribute are treated as distinct symbols without any connections, and hence horizontal analyses rely, to a large extent, on simple pattern matching. More specifically, one uses the trivial equality relation $=$ on values of an attribute. By taking into consideration of vertical analysis, one may introduce more flexibility in horizontal analysis. For example, attribute values can be clustered or discretization to obtain more generalized decision rules in machine learning.[10,18] The use of concept hierarchies in data mining can produce multi-level association rules.[5] It is evident that a systematic study on vertical analysis of information table is needed.

Lin suggested the use of neighborhood systems for modeling approximation in databases.[8,12] In this framework, an element of a universe is associated with a nonempty family of subsets of the universe. The family is called a neighborhood system of the element, and each subset is called a neighborhood of the element. A neighborhood of an element consists of all elements that are semantically related to the element. A neighborhood system represents various such semantically related groups. Neighborhood systems on values of an attribute can therefore be used to represent their semantics relationships. The commonly used methods, such as binary relations, fuzzy binary relations, distance functions, dissimilarity and similarity measures, and hierarchical structures, can be understood as special classes of neighborhood systems.[25] In addition, the notion of neighborhood systems can be applied to situations where the meaning of a distance or a similarity function is not clear. It may also be applied when only the qualitative information (for example, the order implied by the numeric values) is useful rather than the precise numeric values.[25]

The objective of this paper is to combine the notions of information tables and neighborhood systems. A neighborhood system based semantics is suggested for the vertical analysis of information tables. This combination enriches information tables and may enlarge their application domains.

The rest of the paper is organized as follows. Section 2 briefly reviews the basic concepts of neighborhood systems. Section 3 first describes information tables without vertical semantics, and then introduces neighborhood systems on attribute values. To illustrate the usefulness of neighborhood semantics, two related fundamental problems involved in information tables are discussed. They are information granulation and information retrieval. The former deals with clustering of objects based on their properties, and the latter concerns searching objects with certain properties. For information tables without neighborhood semantics, one can obtain a partition of the set of objects, and carry out exact retrieval. With the added neighborhood semantics, one can obtain more general structures on the set of objects, and perform approximate retrieval.

2. NEIGHBORHOODS AND NEIGHBORHOOD SYSTEMS

The concept of neighborhood systems was introduced by Sierpiński and Krieger for the study of Féchet (V)spaces.[19] It is originated from the abstraction of geometric notion of closeness. Two points in a space are *close to*, or *approximate to*, each other if one point is in a neighborhood of the other.[9] Lin adopted neighborhood systems to describe relationships between objects in database systems for the purpose of approximate retrieval.[8] Yao used the notion for investigating information granulation and rough set approximations.[24]

Let U denote a finite and non-empty set called the universe. For an element $x \in U$, one may associate with it a subset $n(x) \subseteq U$ called a *neighborhood* of x. By associating a non-empty family of neighborhoods $\mathrm{NS}(x) \subseteq \mathcal{P}(U)$ to x, one obtains a *neighborhood system* of x, where $\mathcal{P}(U)$ is the power set of U. A neighborhood system may be formally interpreted as an operator from U to $\mathcal{P}(\mathcal{P}(U))$ that maps each element of U to a family of subsets of U. The collection of neighborhood systems of all elements in U, denoted by $\mathrm{NS}(U)$, determines a Fréchet (V)space $(U, \mathrm{NS}(U))$. There is no additional requirements on neighborhood systems. A neighborhood of x may or may not contain x. If $x \in n(x)$, $n(x)$ is called a reflexive neighborhood of x. If every neighborhood in a neighborhood system is reflexive, the system is called a reflexive neighborhood system. If a neighborhood system consists of only one neighborhood, it is called an 1-neighborhood system.[23] If a neighborhood system consists of a sequence of nested neighborhoods, it is called a nested neighborhood systems.[25]

Neighborhood systems represent the information or knowledge about relationships between elements of a universe. Intuitively speaking, elements in a neighborhood of an element are somewhat close to, or similar to, that element. Elements of $n(x)$ are drawn towards x by indistinguishability, similarity, or functionality.[9] A neighborhood system $\mathrm{NS}(x)$ of x groups the universe into classes. Distinct neighborhoods of x consist of elements having different types of, or various degrees of, similarity to x. Elements in the same neighborhood $n(x)$ are regarded to be indiscernible or at least not noticeably distinguishable from x. In the present study, we focus on similarity based interpretations of neighborhood systems by considering only reflexive neighborhood systems.

For an element of the universe, different neighborhood systems may be defined. Each of them represents the available knowledge about the universe from different points of views. For example, different neighborhood systems may be supplied by a group of experts. Neighborhood systems can be combined by set-theoretic operations based on the notion of power algebras.[3,25] For two neighborhood systems $\mathrm{NS}_1(x)$ and $\mathrm{NS}_2(x)$, the complement, intersection and union are defined by:

$$\begin{aligned}
\neg\mathrm{NS}_1(x) &= \{\sim n_{i_1}(x) \mid n_{i_1}(x) \in \mathrm{NS}_1(x)\},\\
\mathrm{NS}_1(x) \sqcap \mathrm{NS}_2(x) &= \{n_{i_1}(x) \cap n_{i_2}(x) \mid n_{i_1}(x) \in \mathrm{NS}_1(x), n_{i_2}(x) \in \mathrm{NS}_2(x)\},\\
\mathrm{NS}_1(x) \sqcup \mathrm{NS}_2(x) &= \{n_{i_1}(x) \cup n_{i_2}(x) \mid n_{i_1}(x) \in \mathrm{NS}_1(x), n_{i_2}(x) \in \mathrm{NS}_2(x)\}. \qquad (1)
\end{aligned}$$

They may be interpreted as extensions of set-theoretic operations in a framework of set-based computations.[28] An 1-neighborhood system corresponds to a subset of the universe. The extended operations reduce to the standard set-theoretic operations if only 1-neighborhood systems are considered. Operations $\sqcap$ and $\sqcup$ are commutative and associative. They are not distributive over each other. The unit element of $\sqcap$ is $\{U\}$ and the zero element of $\sqcup$ is $\{\emptyset\}$. Operations $\sqcap$ and $\sqcup$ are not idempotent. In general, $\emptyset \in \mathrm{NS}(x) \sqcap \neg\mathrm{NS}(x)$ and $U \in \mathrm{NS}(x) \sqcup \neg\mathrm{NS}(x)$. A detailed study of such a system can be found in a paper by Brink on second-order Boolean algebras.[2]

3. INFORMATION TABLES: INFORMATION GRANULATION AND RETRIEVAL

In this section, we introduce neighborhood semantics into information tables after pointing out some difficulties when such semantics information is not available. For simplicity, the discussion is focused on two specific problems, one is information granulation, and the other is information retrieval. The arguments can be applied to the problems of machine learning and data mining.

3.1. Information Tables without Vertical Semantics

An information table, a notion studied by many authors,[11,14,15,17,20,32] is formally defined by a quadruple:

$$T = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\}), \qquad (2)$$

U is a finite and nonempty set of objects,
At is a finite and nonempty set of attributes,
V_a is a nonempty set of values for each attribute $a \in At$,
$I_a : U \longrightarrow V_a$ is an information function for each attribute $a \in At$.

Each information function I_a is a total function that maps an object of U to exactly one value in V_a. For an object $x \in U$, $I_a(x)$ is the value of x on the attribute a. One can extend those information functions to subsets of attributes. For $A \subseteq At$, $I_A(x)$ is the values of x on a set of attributes A.

Information tables as defined above do not incorporate both horizontal and vertical semantics of attribute values. Let C_r denote a set of constraints on objects in taking their values with respect to the set of attributes At. A constrained information table may be expressed by:

$$CT = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\}, C_r). \qquad (3)$$

Examples of constraints may be functional dependency representing global dependency between two subsets of attributes. One may also impose local dependency between two subsets of attributes by stating how particular combinations of values on one subset of attributes determine the possible combinations of values on another subset of attributes. When C_r is the empty set, a constrained information table is reduced to a standard information table. By explicitly adding the horizontal constraints on attribute values into information tables, we may have a more accurate and realistic formal model for describing real world problems.

When vertical semantics information is unavailable, we can only use the trivial equality relation $=$ on the values of an attribute. If an order relation is implied by the values, we may use other relations such as $>$ and $<$. However, those relations can be easily re-expressed in terms of the equality relation. Thus, we will only consider the equality relation.

One of the issues in the theory of rough sets deals with information granulation according to the knowledge provided by an information table.[16,17] Relationships between objects are defined based on the equality relation on their attribute values. With respect to an attribute $a \in At$, a relation $\Re_a$ is given by: for $x, y \in U$,

$$x \Re_a y \Longleftrightarrow I_a(x) = I_a(y). \tag{4}$$

That is, two objects are considered to be indiscernible, in the view of single attribute a, if and only if they have exactly the *same* value. $\Re_a$ is an equivalence relation. The reflexivity, symmetry and transitivity of $\Re_a$ follow trivially from the properties of the equality relation $=$. For a subset of attributes $A \subseteq At$, this definition can be extended as follows:

$$\begin{aligned} x \Re_A y &\Longleftrightarrow (\forall a \in A) I_a(x) = I_a(y) \\ &\Longleftrightarrow I_A(x) = I_A(y). \end{aligned} \tag{5}$$

That is, in terms of all attributes in A, x and y are indiscernible, if and only if they have the same value for every attribute in A. The extended relation is still is an equivalence relation.[14]

For an object $x \in U$, its equivalence class (neighborhood) defined by using an attribute $a \in At$ is given by:

$$\begin{aligned} (\Re_a)_p(x) &= \{y \in U \mid y \Re_a x\} \\ &= \{y \in U \mid I_a(y) = I_a(x)\} \\ &= \{y \in U \mid I_a(y) \in (=_a)_p(I_a(x))\}, \end{aligned} \tag{6}$$

where $=_a$ denotes the equality relation $=$ on attribute V_a, and $(=_a)_p(v) = \{v\}$ for $v \in V_a$. We express the result in this more general form for possible generalizations. As will be shown later, the same expression can be used in cases where arbitrary binary relations or neighborhood systems are defined on V_a. With respect to a set of attributes, we have:

$$\begin{aligned} (\Re_A)_p(x) &= \{y \in U \mid y \Re_A x\}, \\ &= \bigcap_{a \in A} (\Re_a)_p(x) \\ &= \bigcap_{a \in A} \{y \in U \mid I_a(y) \in (=_a)_p(I_a(x))\}. \end{aligned} \tag{7}$$

We therefore obtain a partition of the universe of objects, $\{(\Re_A)_p(x) \mid x \in U\}$, consisting of a family of disjoint subsets whose union is the universe. It is a special type of granulation of the universe. By treating equivalence classes as basic granules, one can derive rough set approximations of an arbitrary subset of the universe.[16]

Consider now the problem of retrieval in an information table without vertical semantics. For clarity, we choose a class of simple queries formed by the equality sign and logical connectives $\wedge$ (and) and $\vee$ (or). An atomic query is of the form, *attribute_name* $=$ *attribute_value*. If q_1 and q_2 are two queries, both $(q_1 \wedge q_2)$ and $(q_1 \vee q_2)$ are queries. For an atomic query $q : a = v$, where $a \in At$ and $v \in V_a$, the following set of objects:

$$ret(a = v) = \{x \in U \mid I_a(x) = v\}, \tag{8}$$

satisfy the query. The set $ret(q)$ is called the retrieved set of objects of query q. Let $ret(q_1)$ and $ret(q_2)$ be the retrieved sets of objects of queries q_1 and q_2. The retrieved sets of $q_1 \wedge q_2$ and $q_1 \vee q_2$ are given by:

$$\begin{aligned} ret(q_1 \wedge q_2) &= \{x \in U \mid x \text{ satisfies } q_1 \text{ and } x \text{ satisfies } q_2\} \\ &= \{x \in U \mid x \in ret(q_1) \text{ and } x \in ret(q_2)\} \\ &= ret(q_1) \cap ret(q_2), \end{aligned}$$

$$\begin{aligned} ret(q_1 \vee q_2) &= \{x \in U \mid x \text{ satisfies } q_1 \text{ or } x \text{ satisfies } q_2\} \\ &= \{x \in U \mid x \in ret(q_1) \text{ or } x \in ret(q_2)\} \\ &= ret(q_1) \cup ret(q_2). \end{aligned} \tag{9}$$

Queries, represented as logical expressions, are therefore interpreted in set-theoretic terms.[32] In general, one can obtain the retrieved set of any query. For example, the retrieved set of the query $q_1 \wedge (q_2 \vee q_3)$ can be obtained by $ret(q_1 \wedge (q_2 \vee q_3)) = ret(q_1) \cap (ret(q_2) \cup ret(q_3))$. By the distributive properties of $\wedge$ and $\vee$, and $\cap$ and $\cup$, $q_1 \wedge (q_2 \vee q_3)$ and $q_1 \wedge q_2 \vee q_1 \wedge q_3$ are equivalent queries. They produce the same set of retrieved objects. Similarly, q, $q \wedge q$ and $q \vee q$ are equivalent queries.

3.2. Information Tables with Binary Relation Semantics

Many proposals have been made to avoid limitations imposed by an equivalence relation in the theory of rough sets. Zakowski suggested using a compatibility relation, i.e., a reflexive and symmetric relation, on the universe.[33] Wybraniec-Skardowska introduced different rough set models based on various types of binary relations.[22] Yao and Lin examined non-standard rough set models induced by various binary relations, based on the results from modal logic.[27] Those studies assumed that a binary relation is defined on the universe of objects U. Yao *et al.* investigated various binary relations on attributes values in information tables for the development of a generalized theory of rough sets.[29,30] A binary relation on objects is defined using arbitrary binary relations on their attribute values.

Formally, a constrained information table with binary relation semantics can be described by:

$$CBT = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\}, C_r, \{R_a \mid a \in At\}), \tag{10}$$

where R_a is a binary relation on the values of an attribute $a \in At$. We can define a binary relation on U:

$$x \Re_a y \Longleftrightarrow I_a(x) R_a I_a(y). \tag{11}$$

An object x is related to another object y, based on only attribute a, if and only if their values on a are related. With respect to a subset A of attributes, x is related to y if and only if their values are related for every attribute in A. Hence, we have the definition:

$$\begin{aligned} x \Re_A y &\Longleftrightarrow (\forall a \in A) I_a(x) R_a I_a(y) \\ &\Longleftrightarrow (\forall a \in A) x \Re_a y. \end{aligned} \tag{12}$$

When all relations R_a are chosen to be $=$, we obtain the original definition. Relation $\Re_A$ has the common properties of relations $R_a, a \in A$. For instance, if every relation R_a is reflexive, symmetric and transitive, respectively, then $\Re_A$ is reflexive, symmetric and transitive, respectively. We assume that R_a is at least reflexive so that it may be interpreted as a similarity relation. For two values $v, v' \in V_a$, if $v R_a v'$ we say that v is similar to v'. For two objects $x, y \in U$, if $x \Re_A y$, we say that x is similar to y with respect to the set of attributes A.

From the reflexive relation $\Re_a$, for an object $x \in U$ we associate the following predecessor neighborhood of x:

$$\begin{aligned} (\Re_a)_p(x) &= \{y \in U \mid y \Re_a x\} \\ &= \{y \in U \mid I_a(y) R_a I_a(x)\} \\ &= \{y \in U \mid I_a(y) \in (R_a)_p(I_a(x))\}. \end{aligned} \tag{13}$$

For a subset of attributes $A \subseteq At$, we have:

$$\begin{aligned} (\Re_A)_p(x) &= \{y \in U \mid y \Re_A x\} \\ &= \bigcap_{a \in A} (\Re_a)_p(x) \\ &= \bigcap_{a \in A} \{y \in U \mid I_a(y) \in (R_a)_p(I_a(x))\}. \end{aligned} \tag{14}$$

This definition is clearly consistent with the definition given in terms of the equality relation $=$. By the reflexivity of $\Re_A$, the family $\{(\Re_A)_p(x) \mid x \in U\}$ is a covering of the universe of objects, consisting of possibly overlap subsets whose union is the universe. It is a generalization of a partition. By treating $(\Re_A)_p(x)$'s as basic granules, one can derive generalized rough set approximations.[16]

Approximate retrieval in an information table with binary relation semantics involves a query relaxation process. Consider an atomic query $q : a = v$, where $a \in At$ and $v \in V_a$. The value v is associated with a predecessor neighborhood:

$$(R_a)_p(v) = \{v' \in V_A \mid v' R_a v\}. \tag{15}$$

We may define a relaxed query:

$$q_r = \bigvee_{v' \in (R_a)_p(v)} a = v'. \tag{16}$$

The retrieved set of objects by q_r is given by:

$$\begin{aligned} ret(q_r) &= \bigcup_{v' \in (R_a)_p(v)} ret(a = v') \\ &= \{x \in U \mid I_a(x) R_a v\}. \end{aligned} \tag{17}$$

That is, using the relaxed query, an object is retrieved if its value on a is similar to v, instead of equaling to v. By combining $ret(q_r)$ with the retrieved set of q, we have the following family of retrieved sets:

$$aret(q) = \{ret(q), ret(q_r)\}. \tag{18}$$

Let q_1 and q_2 be two queries with retrieved family of sets $aret(q_1)$ and $aret(q_2)$, the retrieved families of sets of $q_1 \wedge q_2$ and $q_1 \vee q_2$ are given by:

$$\begin{aligned} aret(q_1 \wedge q_2) &= aret(q_1) \sqcap aret(q_2), \\ aret(q_1 \vee q_2) &= aret(q_1) \sqcup aR(q_2). \end{aligned} \tag{19}$$

An atomic query is interpreted by a retrieved family of sets, and logical connectives are interpreted by neighborhood systems operations. Since $\sqcap$ and $\sqcup$ are not distributive over each other, two logically equivalent expressions $q_1 \wedge (q_2 \vee q_3)$ and $q_1 \wedge q_2 \vee q_1 \wedge q_3$ may produce different results. Similarly, q, $q \wedge q$, and $q \vee q$ may also produce different results.

3.3. Information Tables with Neighborhood Semantics

By combining a binary relation and the equality relation $=$ on values of an attribute, we essentially using a neighborhood systems containing two neighborhoods $\{\{v\}, (R_a)_p(v)\}$. By extending the same formulation, we can define a constrained information table with neighborhood semantics as follows:

$$CNT = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\}, C_r, \{\mathrm{NS}_a \mid a \in At\}), \tag{20}$$

where $\mathrm{NS}_a : V_a \longrightarrow \mathcal{P}(\mathcal{P}(V_a))$ defines a neighborhood system for each value $v \in V_a$ of an attribute $a \in At$. The use neighborhood systems provides more flexibility in describe semantic closeness of attribute values.

Consider an object $x \in U$ whose value on an attribute $a \in At$ is $I_a(x) \in V_a$. Let $I_a(x) = v \in V_a$ and let the neighborhood system of v be given by:

$$\mathrm{NS}_a(v) = \{n_a^1(v), \ldots, n_a^{K_v}(v)\}. \tag{21}$$

With respect to a neighborhood $n_a^i(v), 1 \le i \le K_v$, we define a neighborhood of x by:

$$\begin{aligned} n_a^i(x) &= \{y \in U \mid I_a(y) \in n_a^i(v)\} \\ &= \{y \in U \mid I_a(y) \in n_a^i(I_a(x))\}. \end{aligned} \tag{22}$$

The same symbols n_a^i's are used for both neighborhoods of objects and neighborhoods of attribute values, in order to show their connections. An object y is in a neighborhood of x if its value on attribute a is in the corresponding neighborhood of the value of x. The neighborhood system $\mathrm{NS}_a(v)$ defines a neighborhood system of x:

$$\mathrm{NS}_a(x) = \{n_a^1(x), \ldots, n_a^{K_v}(x)\}. \tag{23}$$

Again, the same set of symbols are used. An 1-neighborhood system on V_a can be defined based on a binary relation R_a on V_a:

$$\mathrm{NS}_a(v) = \{(R_a)_p(v) = \{v' \in V_a \mid v' \Re v\}\}, \tag{24}$$

which produces an 1-neighborhood system on U:

$$\mathrm{NS}_a(x) = \{(\Re_a)_p(x) = \{y \in U \mid y \Re_a x\}\}. \tag{25}$$

From this connection, one can say that neighborhood systems defined on U are consistent to, and generalizes, the ones given by a binary relation. In general, each attribute value induces a neighborhood system which is not necessarily an 1-neighborhood system as in the case of binary relation. We must use neighborhood system intersection to define a neighborhood system with respect to a subset of attributes $A \subseteq At$:

$$\mathrm{NS}_A(x) = \sqcap_{a \in A}\ \mathrm{NS}_a(x). \tag{26}$$

When reflexive 1-neighborhood systems on attribute values are used, as the case of binary relation semantics, one obtains a covering of the universe. This produces a single-layered granulation of the universe. In general, a reflexive neighborhood systems on attributes values may produce multi-layered granulations of the universe.

Approximate retrieval in information tables with neighborhood semantics is similar to that of using binary relations.[25] For an atomic query $q : a = v$, where $a \in At$ and $v \in V_a$, suppose a neighborhood system of v is given by:

$$\mathrm{NS}_a(v) = \{n_a^1(v), \ldots, n_a^{K_v}(v)\}. \tag{27}$$

With respect to a neighborhood $n_a^i(v)$, $1 \leq i \leq K_v$, we construct a query:

$$q^i : \bigvee_{v' \in n_a^i(v)} a = v', \tag{28}$$

which can be considered as a relaxed version of q by the neighborhood $n_a^i(v)$. The retrieved set of objects by q^i is given by:

$$ret(q^i) = \bigcup_{v' \in n_a^i(v)} ret(a = v'). \tag{29}$$

From the neighborhood system $\mathrm{NS}_a(v)$, we have the following set of queries:

$$\mathrm{NS}(q) = \{q, q^1, \ldots, q^{K_v}\}. \tag{30}$$

It may be interpreted as a neighborhood system of q, if one extends the notion of neighborhoods to the set of all queries (logical expressions). The corresponding retrieved family of sets is given by:

$$aret(q) = \{ret(q), ret(q^1), \ldots, ret(q^{K_v})\}. \tag{31}$$

The family $aret(q)$ may be interpreted as a neighborhood system of an element that satisfies the query q. Based on the results of atomic queries, one may define the results of any query recursively. Let q_1 and q_2 be two queries with retrieved family of sets $aret(q_1)$ and $aret(q_2)$, the retrieved families of sets of $q_1 \wedge q_2$ and $q_1 \vee q_2$ are given by:

$$\begin{aligned} aret(q_1 \wedge q_2) &= aret(q_1) \sqcap AR(q_2), \\ aret(q_1 \vee q_2) &= aret(q_1) \sqcup AR(q_2). \end{aligned} \tag{32}$$

Based on the information provided by the retrieved family of subsets of objects, one may design a systematic method to rank objects in the universe with respect to a query.[25] An advantage of approximate retrieval based on neighborhood semantics is that a non-linear ordering of objects may be produced.

4. CONCLUSION

The use of information tables as a knowledge representation method involves horizontal semantics characterized by dependency of attributes and vertical semantics characterized by relationships of attribute values. While there have been extensive studies on the horizontal analysis of information tables in databases, machine learning, and data mining, much less attention has paid to the vertical analysis of information tables. There is a need for systematic studies on the vertical analysis of relationships between attribute values. In this paper, we have introduced neighborhood systems on attribute values. Binary relations on attribute values are described by 1-neighborhood systems. Other commonly used methods, such concept hierarchies, fuzzy binary relations, similarity measures, and distance functions on attribute values, can be interpreted in terms neighborhood systems.[25] With the introduction of the neighborhood semantics, one brings more flexibility and expression power to information tables.

Two specific problems, information granulation and information retrieval, have been examined. Without vertical semantics, the only available tool is the trivial equality relation $=$ on attribute values. This leads to a simple single-layered granulation (a partition) of the universe. The standard theory of rough sets is developed based on this framework.[16] Without considering semantic closeness between attribute values, only exact retrieval is possible. With the introduction of binary similarity relations on attribute values, one is able to obtain generalized granulation (a covering) of the universe and two-layered approximate retrieval. Consequently, generalized rough set theories may be developed. Neighborhood systems can describe various degrees and types of, and multi-layered, similarity of attribute values. One may therefore derive multi-layered granulation of the universe, and multi-layered approximate retrieval.

This paper presents some preliminary results on the use of neighborhood semantics for information tables. Although there is no constraints on neighborhood systems, in real applications one may in fact use very simple or special types of neighborhood systems. For instance, a neighborhood system may only contain a few, rather than a very large number of, neighborhoods. In this paper, we only examined some formal aspects of neighborhood semantics. The computational issues need further study. The implications and applications of neighborhood semantics in other problems, such as machine learning and data mining, should also be further investigated.

REFERENCES

1. A. Bell, "Discovery and maintenance of functional dependencies by independencies," *Proceedings of KDD-95*, pp. 27-32, 1995.
2. C. Brink, "Second-order Boolean algebras," *Quaestiones Mathematicae*, **7**, pp. 93-100, 1984.
3. C. Brink, "Power structures," *Algebra Universalis*, **30**, pp. 177-216, 1993.
4. C.J. Butz, S.K.M. Wong, and Y.Y. Yao, "On data and probabilistic dependencies," *Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 1692-1697, 1999.
5. J. Han, Y. Cai, and N. Cercone, "Data-driven discovery of quantitative rules in data bases," *IEEE Transactions on Knowledge and Data Engineering*, **5**, pp. 29-40, 1993.
6. T.B. Iwinski, "Ordinal information systems I," *Bulletin of the Polish Academy of Sciences, Mathematics*, **36**, pp. 467-475, 1988.
7. G.J. Klir, and B. Yuan, *Fuzzy Sets and Fuzzy Logic, Theory and Applications*, Prentice Hall, New Jersey, 1995.
8. T.Y. Lin, "Neighborhood systems and approximation in relational databases and knowledge bases," *Proceedings of the 4th International Symposium on Methodologies of Intelligent Systems*, 1988.
9. T.Y. Lin, "Granular computing on binary relations I: data mining and neighborhood systems," in: L. Polkowski and A. Skowron (Eds.), *Rough Sets in Knowledge Discovery 1, Methodology and Applications*, Physica-Verlag, Heidelberg, pp. 286-318, 1998.
10. T.Y. Lin, and N. Cercone (Eds.), *Rough Sets and Data Mining: Analysis for Imprecise Data*, Academic Publishers, Boston, 1997.
11. W. Jr. Lipski, "On databases with incomplete information," *Journal of the ACM*, **28**, pp. 41-70, 1981.
12. J.B. Michael, T.Y. Lin, "Neighborhoods, rough sets, and query relaxation in cooperative answering," in: T.Y. Lin and N. Cercone (Eds.), *Rough Sets and Data Mining: Analysis of Imprecise Data*, Kluwer Academic Publishers, Boston, pp. 229-238, 1997,
13. R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (Eds.), *Machine Learning*, Tioga, 1983.
14. E. Orlowska, "Logic of indiscernibility relations," *Lectures Notes in Computer Science, vol. 208*, Springer-Verlag, Berlin, pp. 177-186, 1985.

15. Z. Pawlak, "Information systems – theoretical foundations," *Information Systems*, **6**, pp. 205-218, 1981.
16. Z. Pawlak, "Rough sets," *International Journal of Computer and Information Sciences*, **11**, pp. 341-356, 1982.
17. Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*, Klumer Academic Publishers, Boston, 1991.
18. L. Polkowski, and A. Skowron (Eds.), *Rough Sets in Knowledge Discovery 1,2*, Physica-Verlag, Heidelberg, 286-318, 1998.
19. W. Sierpiński, C. Krieger, *General Topology*, University of Toronto, Toronto, 1956.
20. D. Vakarelov, "A modal logic for similarity relations in Pawlak knowledge representation systems," *Fundamenta Informaticae*, **XV**, pp. 61-79, 1991.
21. A. Wasilewska, "Conditional knowledge representation system – model for an implementation," *Bulletin of Polish Academy of Science, Mathematics*, **37**, pp. 63-69, 1990.
22. U. Wybraniec-Skardowska, "On a generalization of approximation space," *Bulletin of the Polish Academy of Sciences, Mathematics*, **37**, pp. 51-61, 1989.
23. Y.Y. Yao, "Relational interpretations of neighborhood operators and rough set approximation operators," *Information Sciences*, **111**, pp. 239-259, 1998.
24. Y.Y. Yao, Granular computing using neighborhood systems, in: R. Roy, T. Furuhashi, and P.K. Chawdhry (Eds.), *Advances in Soft Computing - Engineering Design and Manufacturing*, Springer-Verlag, London, pp. 539-553, 1999.
25. Y.Y. Yao, Neighborhood systems and approximate retrieval, manuscript, 2000.
26. Y.Y. Yao, Information granulation and rough set approximation, manuscript, 2000.
27. Y.Y. Yao, and T.Y. Lin, "Generalization of rough sets using modal logic," *Intelligent Automation and Soft Computing, An International Journal*, **2**, pp. 103-120, 1996.
28. Y.Y. Yao, and N. Noroozi, "A unified model for set-based computations," *Soft Computing: 3rd International Workshop on Rough Sets and Soft Computing*, pp. 252-255, 1994.
29. Y.Y. Yao, and S.K.M. Wong, "Generalization of rough sets using relationships between attribute values," *Proceedings of the 2nd Annual Joint Conference on Information Sciences*, pp. 30-33, 1995.
30. Y.Y. Yao, S.K.M. Wong, and T.Y. Lin, "A review of rough set models," in: Lin, T.Y. and Cercone, N. (Eds.), *Rough Sets and Data Mining: Analysis for Imprecise Data*, Academic Publishers, Boston, pp. 47-75, 1997.
31. Y.Y. Yao, and N. Zhong, "On association, similarity and dependency of attributes," *Proceedings of PAKDD'00*, to appear, 2000.
32. Y.Y. Yao, and N. Zhong, Granular computing using information tables, manuscript, 2000.
33. W. Zakowski, "Approximations in the space (U, Π)," *Demonstratio Mathematica*, **XVI**, pp. 761-769, 1983.