Join and Meet Block Based Default Definite Decision Rule Mining from IDT and an Incremental Algorithm

Chen Wu and Jingyu Yang

***Abstract*—**Using maximal consistent blocks of tolerance relation on the universe in incomplete decision table, the concepts of join block and meet block are introduced and studied. Including tolerance class, other blocks such as tolerant kernel and compatible kernel of an object are also discussed at the same time. Upper and lower approximations based on those blocks are also defined. Default definite decision rules acquired from incomplete decision table are proposed in the paper. An incremental algorithm to update default definite decision rules is suggested for effective mining tasks from incomplete decision table into which data is appended. Through an example, we demonstrate how default definite decision rules based on maximal consistent blocks, join blocks and meet blocks are acquired and how optimization is done in support of discernibility matrix and discernibility function in the incomplete decision table.

***Keywords*—**rough set, incomplete decision table, maximal consistent block, default definite decision rule, join and meet block.

I. INTRODUCTION

ROUGH set theory [1,11,12] was put forward by Pawlak in 1982 as a new mathematics tool and has been successively applied in many fields such as decision making, machine learning, automatic control, pattern recognition, data mining and etc[10,13,14]. The original rough set approach is efficient to discover rules from complete decision tables, whereas we always have to face incomplete decision tables (IDT) or incomplete information systems (IIS) in real situations because practical data occurs incomplete to some extent. Currently, people have proposed several rough set approaches to deal with IIS. For example, in [8] a rough set approach to reasoning in IIS is presented and how to derive decision rules directly from incomplete decision table is shown. In [10] a new method of generating all optimal certain rules from IDT is given and proven. [9] proposes a new mining algorithm , which can simultaneously produce rules from incomplete data sets and estimate the missing values in the mining procedure.

However, how to use join and meet of maximal consistent blocks to obtain optimally definite default decision rules from IIS proposed by us is still a new study topic. Tolerance relation is also compatible relation, in discrete mathematics, which is one of the most importantly fundamental and mathematical relations in mathematical formulation. Equivalence relation is compatible and tolerant relation. The collection of all maximal consistent blocks of a compatible relation is also called a complete covering in discrete mathematics and has much usefulness in many scientific fields. Tolerance relation is used to acquire knowledge in incomplete decision table in rough set instead of indiscernibility relation (is really an equivalence relation) in IDT. Compatible relation is of reflexivity and symmetry but not necessary transitivity. A compatible relation can be linked with an undirected graph. The nodes represent objects. Two objects are related if, and only if, there is an edge connecting them. Equivalence relation is nothing but a compatible relation satisfying transitivity. An equivalence relation *R* on *U* determines a partition of *U*, given by the equivalence classes modulo *R*, and this characterizes those coverings of a set that are families of classes of equivalence relation. But to arbitrary compatible relation, things are more complicated in this respect. Corresponding to a compatible relation a set of maximal consistent blocks, also called maximal compatible classes, which form an overlapped covering on the universe, can be uniquely defined. The collection of maximal consistent blocks is also called a complete covering in discrete mathematics. As a compatible relation, tolerance relation produces a set of tolerance classes which are regarded as a collection derived from a generator tolerant with every other element in the collection [8]. Tolerance classes usually construct an overlapped covering on the universe after eliminating repeated ones. But tolerance classes are eventually very different from maximal consistent blocks. In addition to maximal consistent blocks and tolerance classes of an object, join block, a union of some maximal consistent blocks containing the generator, and meet block, an intersection of some maximal consistent blocks containing the generator (also called compatibly kernel block or tolerantly kernel block in some literatures), are introduced by viewing maximal

Manuscript received December 2006. This work was supported in part by Jiangsu Provincial High Educational Natural Science Fund under Grant 02KJA120001.

Chen Wu is with the Computer Department at Nanjing University of Science and Technology, Nanjing Jiangsu Province, 210094 and with the School of Electronics and Information at Jiangsu University of Science and Technology,Zhenjiang,Jiangsu Province 212003, P.R. of China (e-mail: wuchenzj@gmail.com, wuchenzj@sina.com).

Jingyu Yang is with the Computer Department at Nanjing University of Science and Technology, Nanjing Jiangsu Province, 210094, P.R. of China (e-mail:yangjiy@mail.njust.edu.cn).

consistent blocks as primitives. The later two blocks can also build an overlapped covering respectively [14].

Blocks or classes are similar to granules in granulation theory, which is a hot topic nowadays. The results got here can also be considered to be some real implementation of neighborhood system in concrete [4,5,7,15]. Upper and lower approximations and several properties are appropriately discussed.

Through an example, the newly defined concept default definite rules acquired from incomplete decision tables is illustrated by analyzing maximal consistent blocks, join blocks and meet blocks. Optimality is also reached by benefiting from discernibility matrix and discernibility function. The finally obtained formation of default definite decision rules with some restrictions extremely show that they are newly proposed and significant to decision making in environment of incomplete decision table.

In accordance with the rule mining approach, we suggest an incremental learning algorithm to update default definite decision rules. It is effective and does not need to deal with the entire IDT when new objects are appended one by one.

The rest of the paper is organized as follows. Section 2 gives out some preliminary notations about blocks or classes based on maximal consistent blocks. Section 3 introduces upper and lower approximations and discusses some other properties. Section 4 demonstrates how to directly extract default definite rules from an IIS based on maximal consistent blocks. Another concept weak consistence is accordingly defined. Section 5 continuously discovers default definite decision rules with some restrictions from the IDT based on join and meet blocks. The form of the rules represents our certain productive notation. It facilitates knowledge rules clarified. Section 6 reduces the knowledge rules furthermore by using discernibility matrix and discernibility function. Section 7 suggests an incremental algorithm for updating default definite decision rules in data appending to an incomplete decision table, not need to regenerate them again. Section 8 concludes the paper.

II. Preliminaries

Definition 1. An IDT or IIS is a quadruple $S=(U,A,V,f)$, where U is the finite non-empty universe, A is the finite non-empty attribute set. $A=C\cup D$ and $C\cap D=\varnothing$, where C represents condition attribute set, D represents decision attribute set. For any $a\in A$, V_a represents the value domain of attribute a. $V=\cup_{a\in A}V_a$ is called attribute domain. $f=\{f_a \mid \ a\in A\}$ is called information function set. $f_a(x)=v$ means that the value of attribute a on object $x\in U$ is $v\in V_a$. If there at least one attribute $a\in C$ such that V_a contains null value, then S is called an IDT or IIS in general, otherwise complete decision table or complete information system. $f_C(u)=\{f_a(u)\mid a\in C\}$ is also called the information vector of object *u*. Null value (applicable) is denoted by "*". We always assume that $V_d(d\in D)$ does not contain null value "*". Each $d\in D$ is a decision attribute. Without loss of generality, $D=\{d\}$ is always assumed and is called a single incomplete decision table (IDT for short), otherwise, an multiple incomplete decision table. All can be called IIS.

Let $B\subseteq C$. The tolerance relation derived by B is defined by: $SIM(B)=\{(u,w)\in U\times U:\forall a\in B,\ f_a(u) = f_a(w)\vee f_a(u)=*\vee f_a(w)=*\}$.

$S_B(u)=\{\ w\in U\ :\ (u,w)\in SIM(B)\ \}$ is called a tolerance class[8] for $u\in U$ with respect to (w.r.t. for short) B. In $S_B(u)$, each object is tolerant with u and is regarded as indiscernible with u. But two objects other than u in $S_B(u)$ are not ensured to be tolerant mutually. $U/SIM(B)=\{S_B(u)\mid u\in U\}$ denotes the collection of all tolerance classes. Since tolerance relation $SIM(B)$ is reflexive and symmetric, it is also a compatible relation. We use another symbol $TIM(B)$ to substitute for $SIM(B)$, ie, $TIM(B)=SIM(B)$, but its complete covering is defined in this way: $U/TIM(B)=\{W\subseteq U: W^2\subseteq TIM(B),\ \forall u(u\in U\wedge u\notin W\rightarrow (W\cup\{u\})^2\not\subseteq TIM(B))\}$, which is different from $U/SIM(B)$. Elements in $U/TIM(B)$, called blocks or granules , are maximal consistent blocks or maximal compatible classes, not the same as tolerance classes in $U/SIM(B)$. Both may not form different partition of U, but always different overlapped covering. Two objects in a maximal consistent block always are compatible or tolerant w.r.t. $TIM(B)$. Maximal consistent block is a compatible block which could never be included in another consistent block or compatible block. If an object is put into a maximal consistent block to which it does not belong, the maximum of its compatibility or consistence will be broken. Regarding all maximal consistent blocks in $U/TIM(B)$ as atomic blocks, we may define join block $U_B(u)$ and meet block $L_B(u)$ of *u* respectively as follows: $U_B(u)=\cup W\ (W\in U/TIM(B), u\in W)$; $L_B(u)=\cap W(W\in U/TIM(B), u\in W)$.

Meanwhile we directly introduce block[1] $K_B(u)=\cap S_B(w)(u\in S_B(w))$ referring to. $U_B(u)$ is a union of all maximal consistent blocks containing *u*. $L_B(u)$ is an intersection of them. $L_B(u)$ is called the compatible core[1] of *u* w.r.t. $U/TIM(B)$, denoted by $<u>$, while $K_B(u)$, an intersection of all tolerance classes including *u* in $U/SIM(B)$, is called the tolerance core of *u*.

$\{S_B(u): u \in U\}$, $\{U_B(u): u \in U\}$, $U/TIM(B)$, $\{L_B(u): u \in U\}$ and $\{K_B(u): u \in U\}$ form five different overlapped coverings on U respectively. They are called overlapped knowledge representation systems by domain experts. $\{L_B(u) \mid u \in U\}$ is still a overlapped covering but maybe not a partition. Two objects in $L_B(u)$ are still compatible or consistent, but two objects in $U_B(u)$ are not.

III. Relations

Although various blocks are introduced in section 2, we just list some results concerning join blocks, meet blocks and their relations for our main work in the present paper is to mine rules. Assume that $B \subseteq C, u \in U$. Then $U_B(u) = S_B(u)$. $L_B(u) = K_B(u) = <u>$. To the limitation of space, Proofs of them are omitted here. You can also refer to [14]. Now we discuss about upper and lower approximations and their properties under different knowledge representation systems mentioned above. At first, by letting $W \subseteq U$, $B \subseteq C$, several upper and lower approximations are defined respectively as follows:

$\overline{U}_B(W) = \{u \in U : U_B(W) \cap W \neq \varnothing\}$;

$\overline{L}_B(W) = \{u \in U : L_B(W) \cap W \neq \varnothing\}$;

$\overline{K}_B(W) = \{u \in U : K_B(W) \cap W \neq \varnothing\}$;

$\overline{E}_B(W) = \{u \in U : \exists Y \in U/TIM(B)\ (u \in Y \wedge (Y \cap W \neq \varnothing))\}$;

$\overline{A}_B(W) = \{u \in U : \forall Y \in U/TIM(B)\ (u \in Y \rightarrow (Y \cap W \neq \varnothing))\}$;

$\overline{S}_B(W) = \{u \in U : S_B(W) \cap W \neq \varnothing\}$ [8];

$\underline{U}_B(W) = \{u \in U : U_B(u) \subseteq W\}$;

$\underline{L}_B(W) = \{u \in U : L_B(u) \subseteq W\}$;

$\underline{K}_B(W) = \{u \in U : K_B(u) \subseteq W\}$;

$\underline{E}_B(W) = \{u \in U : \exists Y \in U/TIM(B)\ (u \in Y \wedge (Y \subseteq W))\}$;

$\underline{A}_B(W) = \{u \in U : \forall Y \in U/TIM(B)\ (u \in Y \rightarrow (Y \subseteq W))\}$;

$\underline{S}_A(W) = \{u \in U : S_B(u) \subseteq W\}$ [8].

Obviously, $\overline{U}_B(W) = \overline{S}_B(W)$, $\underline{U}_B(W) = \underline{S}_B(W)$, $\overline{L}_B(W) = \overline{K}_B(W)$, $\underline{L}_B(W) = \underline{K}_B(W)$. Other properties like in Pawlak's work are listed below.

Property 1. Let $W \subseteq U$, $B \subseteq C$. Then

i) $\underline{U}_B(W) \subseteq \underline{L}_B(W)$, $\underline{U}_B(W) \subseteq \underline{L}_B(W)$;

ii) $\overline{L}_B(W) \subseteq \overline{U}_B(W)$, $\overline{L}_B(W) \subseteq \overline{U}_B(W)$;

iii) $\underline{S}_B(W) \subseteq \underline{E}_B(W) \subseteq \underline{A}_B(W)$,

$\overline{A}_B(W) \subseteq \overline{E}_B(W) \subseteq \overline{S}_B(W)$.

Property 2. Let $W \subseteq U$ and $P, Q \subseteq C$. Then

i) $\underline{U}_P(W) \subseteq W \subseteq \overline{U}_P(W)$; $P \subset Q \Rightarrow$
$(\underline{U}_P(W) \subseteq \overline{U}_Q(W)) \wedge (\overline{U}_B(W) \supseteq \overline{U}_Q(W))$;

ii) $\overline{L}_P(W) \subseteq W \subseteq \overline{L}_P(W)$; $P \subset Q \Rightarrow$
$(\underline{L}_P(W) \subseteq \underline{L}_Q(W)) \wedge (\overline{L}_P(W) \supseteq \overline{L}_Q(W))$;

iii) $\underline{K}_P(W) \subseteq W \subseteq \overline{K}_P(W)$; $P \subset Q \Rightarrow$
$(\underline{K}_P(W) \subseteq \overline{K}_Q(W)) \wedge (\overline{K}_P(W) \supseteq \overline{K}_Q(W))$;

iv) $\overline{E}_P(W) \subseteq W \subseteq \overline{E}_P(W)$; $P \subset Q \Rightarrow$
$(\underline{E}_P(W) \subseteq \underline{E}_Q(W)) \wedge (\overline{E}_P(W) \supseteq \overline{E}_Q(W))$;

v) $\overline{A}_P(W) \subseteq W \subseteq \overline{A}_P(W)$; $P \subset Q \Rightarrow$
$(\underline{A}_P(W) \subseteq \underline{A}_Q(W)) \wedge (\overline{A}_P(W) \supseteq \overline{A}_Q(W))$.

Proofs of properties 1 and 2 can be also performed similarly in [8,14]. Here we show some important results as theorems and give out their proofs as follows.

Theorem 1. $\underline{A}_B(W) = \underline{S}_B(W)$ for any $W \subseteq U$ and any $B \subseteq C$

Proof. By $\underline{A}_B(W)$, for any $u \in \underline{A}_B(W)$ and arbitrary $Z \in U/TIM(B)$, if $u \in Z$, then $Z \subseteq W$. Thus $\cup_{u \in Z \wedge Z \in U/TIM(B)} Z \subseteq W$. For $\cup_{u \in Z \wedge Z \in U/TIM(B)} Z = U_B(u) = S_B(u)$, we have $S_B(u) \subseteq W$. This implies $u \in \underline{S}_B(W)$. Therefore, $\underline{A}_B(W) \subseteq \underline{S}_B(W)$ because *u* is any given in $\underline{A}_B(W)$. Conversely, we have $S_B(u) \subseteq W$ for any $u \in \underline{S}_B(W)$. Thus $\cup_{u \in Z \wedge Z \in U/TIM(B)} Z \subseteq W$ for $S_B(u) = U_B(u) = \cup_{u \in Z \wedge Z \in U/TIM(B)} Z$. This implies that if $u \in Z$, then $Z \subseteq W$ for any $Z \in U/TIM(B)$. So $u \in \underline{A}_B(W)$. Out of that *u* is an any given element in $\underline{S}_B(W)$, we get $\underline{S}_B(W) \subseteq \underline{A}_B(W)$. Summing up, $\underline{A}_B(W) = \underline{S}_B(W)$.

Theorem 2. $\overline{E}_B(W) = \overline{S}_B(W)$ for $\forall W \subseteq U, \forall B \subseteq C$..

Proof. On one hand, we prove $\overline{S}_B(W) \subseteq \overline{E}_B(W)$. For any $u \in \overline{S}_B(W)$. $S_B(u) \cap W \neq \varnothing$. Because $S_B(u) = U_B(u) = \cup_{u \in Z, Z \in U/TIM(B)} Z$, thus we have $S_B(u) \cap W = (\cup_{u \in Z, Z \in U/TIM(B)} Z) \cap W = \cup_{u \in Z, Z \in U/TIM(B)} (Z \cap W) \neq \varnothing$. Therefore, there must exist a $Z \in U/TIM(B)$ such that $u \in Z$ and $Z \cap W \neq \varnothing$. Thus $u \in \overline{E}_B(W)$. So $\overline{S}_B(W)$

$\subseteq \overline{E}_B(W)$ because u is any given in $\overline{S}_B(W)$. On the other hand, for any given $u \in \overline{E}_B(W)$, there exists a $Z \in U/TIM(B)$ such that $u \in Z$ and $Z \cap W \neq \varnothing$. Furthermore, we easily have $\cup_{u\in Y, Y\in U/TIM(B)}(Y \cap W) = (\cup_{u\in Y, Y\in U/TIM(B)} Y) \cap W = U_B(u) \cap W = S_B(u) \cap W \supseteq Z \cap W \neq \varnothing$. so $S_B(u) \cap W \neq \varnothing$. It means $u \in \overline{S}_B(W)$ and $\overline{E}_B(W) \subseteq \overline{S}_B(W)$ because u is any given in $\overline{E}_B(W)$. Conclusively, $\overline{E}_B(W) = \overline{S}_B(W)$ holds.

Theorem 3. $\overline{A}_B(W) \subseteq \overline{S}_B(W)$ for $\forall W \subseteq U, \forall B \subseteq C$.

Proof. Let $u \in \overline{A}_B(W)$ be any given. For any $Z \in U/TIM(B)$, if $u \in Z$, then $Z \cap W \neq \varnothing$ by the definition of $\overline{A}_B(W)$. Moreover, we have $S_B(u) \cap W = U_B(u) \cap W = (\cup_{u\in Y, Y\in U/TIM(B)} Y) \cap W = \cup_{u\in Y, Y\in U/TIM(B)} (Y \cap W) \supseteq Z \cap W \neq \varnothing$. That is, $S_B(u) \cap W \neq \varnothing$. Thus $u \in \overline{S}_B(W)$. Therefore, $\overline{A}_B(W) \subseteq \overline{S}_B(W)$ because u is any given in $\overline{A}_B(W)$.

Theorem 4. $\underline{A}_B(W) \subseteq \underline{E}_B(W)$ for $\forall W \subseteq U, \forall B \subseteq C$.

Proof. Let $u \in \underline{A}_B(W)$ be any given**.** If $u \in Z$ and $Z \in U/TIM(B)$, then $Z \subseteq W$. This means there exists a $Z \in U/TIM(B)$ such that $Z \subseteq W$ if $Z \in U/TIM(B)$. Thus $u \in \underline{E}_B(W)$**.** Therefore**,** $\underline{A}_B(W) \subseteq \underline{E}_B(W)$ because $u \in \underline{A}_B(W)$ is any given.

Theorem 5. $\overline{A}_B(W) \subseteq \overline{E}_B(W)$ for $\forall W \subseteq U, \forall B \subseteq C$.

Proof. Let $u \in \overline{A}_B(W)$ be any given. For each $Z \in U/TIM(B)$, if $u \in Z$, then $Z \cap W \neq \varnothing$. It implies that there exists a $Z \in U/TIM(B)$ such that if $u \in Z$ then $Z \cap W \neq \varnothing$. Thus $u \in \overline{E}_B(W)$. That is, $\overline{A}_B(W) \subseteq \overline{E}_B(W)$ because u in $\overline{A}_B(W)$ is given arbitrarily**.**

IV. Maximal Consistent Block Based Rule Mining

Assume that $S = (U, A, V, f)$ is an IDT, $A = C \cup \{d\}$, and $B \subseteq C$. Attribute value pair (a, v) $(v \in V_a, a \in B)$ is called a B-atomic property. Any B-atomic property or conjunction of different B-atomic properties is called a B-descriptor. Let t be a B-descriptor. If (a, v) is an atomic property occurring in t, then we denote $(a, v) \in t$. $B(t) = \{a \in B : (a, v_a) \in t\}$ is an attribute set constructed by attributes occurring in t. An object having B-descriptor t is called a support of t .The set of all supports of t is denoted by $\| t \|$, ie, $\| t \| = \{x \in U :\ v \in B(x), \forall (a, v) \in t\}$, where $B(x) = \ \{a(x) : a \in B\}$. Obviously, if t and s are two atomic properties, then $\| t \wedge s \| = \| t \| \cap \| s \|$, and $\| t \vee s \| = \| t \| \cup \| s \|$. Let t and s be two descriptors. If for all $(a, v) \in t$, we have $(a, v) \in s$, that is, t is constructed from a subset of atomic properties occurring in s, then we say t is *coarser* than s or s is finer than t and is denoted by $t \succeq s$ or $s \preceq t$. If t is constructed from a proper subset of atomic properties occurring in s, then we say t is proper coarser than s and is denoted by $t \succ s$ or $s \prec t$. It is easy to prove that $t \succeq s \Rightarrow \| t \| \supseteq \| s \|$; $t \succ s \Leftrightarrow \| t \| \supset \| s \|$;

Let $B \subseteq C$. We denote $DES(B) = \{t : t$ is B-descriptor and $\| t \| \neq \varnothing\}$. For any $t \in DES(B)$, if $B(t) = B$, then t is called a full B-descriptor. When t is a full B-descriptor, we denote $t = \wedge_{a\in B}(a, v_a)(v_a \in V_a)$, $\| t \| = \| \wedge_{a\in B}(a, v_a)(v_a \in V_a) \| = \{u \in U : a(u) = v_a \vee a(u) = *,\ a \in B\}$. For $u \in U$, if for any $a \in B$, $a(u)=v_a$ or $a(u)=*$, then we call u compatible with descriptor $t = \wedge_{a\in B}(a, v_a)(v_a \in V_a)$. $\| t \| = \| \wedge_{a\in B}(a, v_a)(v_a \in V_a) \|$ represents all objects compatible with t or support t. We also denote $FDES(B) = \{t \in DES(B) : t$ is a full B-descriptor$\}$ and $FDES_B(u) = \{t \in FDES(B) : x \in \| t \|\}$ for $B \subseteq C$ and $u \in U$.

For any $t \in DES(C)$, let us define a function $\partial : U \to 2^{V_d}$ as follows: $\partial(u) = \{d(z) :\ z \in \| t \|,\ u \in \| t \|\}$, which is called a generalized decision of u in the incomplete IDT, where 2^{V_d} is the power set of V_d. Any (d, w), $w \in \partial(u)$, will be referred to a generalized decision descriptor of u.

In IDS, $IND(\{d\})=SIM(\{d\})= TIM(\{d\})$ because there is no missing value to decision attribute. Thus, $U/IND(\{d\}) = \{I_{\{d\}}(u) : u \in U\} = U/SIM(\{d\}) = \{S_{\{d\}}(u) \mid u \in U\} = U/TIM(\{d\})$ and they form an identical partition on U, where $I_{\{d\}}(u)$ represent an equivalence class containing u and $S_{\{d\}}(u)$ is a similarity class containing u and $S_{\{d\}}(u) = I_{\{d\}}(u)$ for decision attribute d and $|\{d(z) \mid z \in S_{\{d\}}(u) = I_{\{d\}}(u)\}| = 1$ for each $u \in U$.

Let $K \in U/TIM(C)$. $DESV(K) = (v_1, v_2, \cdots, v_n)$, where $v_i = a_i(u)$ if there exists $u \in K$ such that $a_i(u) \neq *$, or $v_i = *$ if for any $u \in K$, $a_i(u) = * (1 \le i \le n)$. $DESV(K)$ is called a optimal descriptor of K.

Let $C(K) = \{a_i \in C : \exists u \in K, s.t. a_i(u) \neq *\}$. If $a_i \in C(K)$, then $|\{a_i(u) : u \in K \wedge\ a_i(u) \neq *\}| = 1$. So we can denote $a_i(K) = \{a_i(u):\ u \in K \wedge\ a_i(u) \neq *\} = v_i$. $\| \wedge_{a_i \in C(K)}(a_i, v_i) \| = K$. If $B \subseteq C$ and $K \in U/TIM(B)$, then $B(K) = \{a \in B : \exists u \in K(a(u) \neq *)\}$, which is parallel similar to the definition of $C(K)$.

Definition 1. Let $S = (U, C \cup \{d\}, V, f)$ be an IDT, $K \in U/TIM(C)$. If there exists a $D_i \in U/IND(\{d\}) = U/SIM(\{d\}) = U/TIM(\{d\})$ such that $K \subseteq D_i$, then K is consistent, otherwise inconsistent. If $\| t \| = \| \wedge_{a \in B}(a, v_a)(v_a \in V_a) \| \neq \emptyset$ and there exits a $D_j \in U/IND(\{d\})$ such that $\| t \| = \| \wedge_{a \in B}(a, v_a)$ $(v_a \in V_a) \| \subseteq D_j$, then $\wedge_{a \in B}(a, v_a) \to (d, j)$ will be a certain decision rule without contradiction to its description condition and decision.

Definition 2. If K is consistent ($K \subseteq D_i$) and there exists $u \in K$, such that $f_C(u) = C(K)$, then $\wedge_{a_i \in C(K)}(a_i, v_i) \to (d, i)$ is a default definite decision rule.

Definition 3. If K is consistent ($K \subseteq D_i$) and for any $u \in K$, such that $f_C(u) \neq C(K)$, then $\wedge_{a_i \in C(K)}(a_i, v_i) \to (d, i)$ also makes default definite decision rule without violation.

Now that $L_B(u) = K_B(u) = \langle u \rangle$, we can define a weak consistence in IDT as follows.

Definition 4. If for any $u, L_B(u) = K_B(u) \subseteq S_{\{d\}}(u) = I_{\{d\}}(u)$, then the IDT is weakly consistent, otherwise, not weakly consistent.

The meaning of the weak consistence in this way is closely approximate to simulate real world, because we prefer assuming that observed objects having similar attribute values must have unique decision, other interfering attribute values can be regarded as noises. So weak consistence is a fundamental assumption to an IDT. One can also see what it really represents through reading our following example.

Lemma If $S = (U, C \cup \{d\}, V, f)$ is consistent, then S is also weakly consistent. But S may not be consistent if S is weakly consistent.

Example 1. An incomplete decision table is shown in Table 1. According to the discussions in the previous sections in the above, we can first calculate all maximal consistent blocks w.r.t *C* and decision classes w.r.t. *d* as follows.

U/TIM(C)=$\{K_1,K_2,K_3,K_4,K_5,K_6,K_7\}$, where $K_1=\{u_1, u_2\}$,$K_2=\{u_3, u_5\}$, $K_3=\{u_4, u_7, u_9\}$, $K_4=\{u_5, u_6\}$, $K_5=\{u_{11}\}$, $K_6=\{u_8, u_9\}$, $K_7=\{u_4, u_7, u_{10}\}$. *U/IND*($\{d\}$)=$\{D_0, D_1, D_2\}$, where $D_0=\{u_3, u_5, u_6, u_{10}, u_{11}\}$, $D_1=\{u_1, u_2, u_4, u_7\}$, $D_2=\{u_8, u_9\}$.Then we can mine default definite rules as knowledge rules in the following.

TABLE I
AN INCOMPLETE DECISION TABLE

U	a	b	c	e	f	d
u_1	1	0	*	2	0	1
u_2	1	0	1	2	0	1
u_3	0	1	2	*	2	0
u_4	1	2	*	1	1	1
u_5	0	*	2	0	2	0
u_6	0	2	2	0	*	0
u_7	1	*	*	1	*	1
u_8	1	2	1	2	1	2
u_9	*	*	1	*	1	2
u_{10}	*	2	0	*	1	0
u_{11}	2	1	1	0	0	0

(1) Since $K_1 \subseteq D_1$, *DESV*(K_1)=(1,0,1,2,0), $f_C(u_2)$ = *DESV*(K_1),(1,0,1,2,0) $\to$ 1, ie, $(a,1) \wedge (b,0) \wedge$ $(c,1) \wedge (e,2) \wedge (f,0) \to (d,1)$ is default definite decision rule.

(2) For $K_2 = \{u_3,\ u_5\} \subseteq D_0$,*DESV*(K_2)=(0,1,2,0,2), but neither of $f_C(u_3)$ and $f_C(u_5)$ equals to *DESV*(K_2). Thus (0,1,2,0,2) $\to$ 0,ie, $(a,0) \wedge (b,1) \wedge (c,2) \wedge (e,0) \wedge (f,2) \to (d,0)$ is a default definite decision rule without violation.

(3) Because $K_4 \subseteq D_0$,*DESV*(K_4)=(0,2,2,0,2) and neither of $f_C(u_5)$ and $f_C(u_6)$ equals to *DESV*(K_4), Thus (0,2,2,0,2) $\to$ 0,ie, $(a,0) \wedge (b,2) \wedge (c,2) \wedge (e,0) \wedge (f,2) \to (d,0)$ is also a default definite rule without violation.

(4) From $K_5 \subseteq D_0$,*DESV*(K_5)=(2,1,1,0,0) and $f_C(u_{11})$ =*DESV*(K_5), we obtain (2,1,1,0,0) $\to$ 0, ie, $(a,2) \wedge (b,1) \wedge (c,1) \wedge (e,0) \wedge (f,0) \to (d,0)$ is a default definite decision rule.

(5) Because $K_6 \subseteq D_2$,*DESV*(K_6)=(1,2,1,2,1) and $f_C(u_8)$ =*DESV*(K_6), we obtain (1,2,1,2,1)$\to$2, ie, $(a,1) \wedge (b,2) \wedge (c,1) \wedge (e,2) \wedge (f,1) \to (d,2)$, a default definite decision rule.

We call all default definite decision rules certainly or without violation simply default definite rules.

V. JOIN AND MEET BLOCK BASED DEFAULT DEFINITE DECISION RULE ACQUISITION

We are going to keep analyzing Table 1 and intending to find some other rules with the view points of join block and meet block in section 2 and 3.

Example 2 Continue acquiring rules from Table 1.

(1) For any D_i (i=0,1,2), $K_3 \not\subseteq D_i$,$K_7 \not\subseteq D_i$. But $K_3 \cap K_7 = \{u_4, u_7\} \subseteq D_1$. *DESV*($K_3 \cap K_7$) =(1,2,*,1,1), so there exists a *u* in K_3 and a *w* in K_7 such that $c(u) \neq *, c(w) \neq *$, and $c(u) \neq c(w)$ according to that

attribute c takes * in *DESV*($K_3 \cap K_7$), ie, *c*(*DESV*($K_3 \cap K_7$))=*. In this case, we get a strictly definite decision rule: (1, 2, *, 1, 1) $\rightarrow$ 1, ie, $(a,1) \wedge (b,2) \wedge (c,*) \wedge (e,1) \wedge (f,1) \rightarrow (d,1)$. In brief form, it is: $(a,1) \wedge (b,2) \wedge (e,1) \wedge (f,1) \rightarrow (d,1)$, but for $\forall u \in U$, $c(u) \notin \{0,1\}$. Here $\forall u \in U, c(u) \notin \{0,1\}$. It is because $(c,1) \in t' \in FDESV(K_3 \setminus (K_3 \cap K_7))$ and $(c,0) \in t'' \in FDESV(K_7 \setminus (K_3 \cap K_7))$ preclude to have the same decision result. *c*(*DESV*($K_3 \cap K_7$))=* only means that attribute *c* can not take any value in $c((K_3 \setminus (K_3 \cap K_7)) \cup (K_7 \setminus (K_3 \cap K_7))) = \{0,1\}$. Otherwise, decision attribute will not take the same decision value. So we call it a restrict default definite rule. The restrict demands that attribute *c* can not take value 1 or 0 intuitively, otherwise the decision result will be different under the same assumption of other attributes taking similar attribute values to that in *DESV*($K_3 \cap K_7$). We can explain that noise for attribute *c* is sensible and decision is also sensible to attribute *c*. We also think of that this restrict default definite rule is much better than a possible one in practice. Although it did not discover before and is a partly restricted rule, however, it is without dilemma in a definite schema. Possible rule only give a possibility but without clear pattern.

(2) From $K_3 \not\subseteq D_i$, $K_6 \subseteq D_2$, $K_3 \cap K_6 = \{u_9\} \subseteq D_2$, we can obtain a meet block based rule $DESV(K_3 \cap K_6) \rightarrow DESV(D_2)$. That is, $DESV(\{u_9\}) = f_C(u_9) \rightarrow DESV(D_2) = 2$, ie, $(*,*,1,*,1) \rightarrow 2$. $(a,*) \wedge (b,*) \wedge (c,1) \wedge (e,*) \wedge (f,1) \rightarrow 2$. Thus $(c,1) \wedge (f,1) \rightarrow 2$. But we should analyze three *'s in the rule to see whether there is a restriction on the related attribute or not. For $(a,*)$, $a(DESV(K_3)) = a(DESV(K_6)) = 1$, therefore *a* might be omitted because it has no conflict taking values $a(K_3), a(K_6)$ at the same time. Meanwhile, for $(b,*)$, $b(DESV(K_3)) = b(DESV(K_6)) = 2$, therefore *b* might be also omitted. In addition, for $(e,*)$, $e(DESV(K_3)) = 1$, $e(DESV(K_6)) = 2$. When $e(DESV(K_3)) = 1$ in K_3, d=1; when $e(DESV(K_6)) = 2$, d=2, which is compatible to $(c,1) \wedge (f,1) \rightarrow 2$. So from $K_6 = \{u_8, u_9\} \subseteq D_2$, we have that rule $(c,1) \wedge (f,1) \rightarrow 2$ should be imposed a restriction $e(u) \neq 1$ for any u .Thus the final default definite decision rule acquired is: $(c,1) \wedge (f,1) \rightarrow 2$, and $e(u) \neq 1$ for any u .

(3) From $K_2 \cap K_4 = \{u_5\} \subseteq D_0$ and furthermore even $K_2 \cup K_4 = \{u_3, u_5, u_6\} \subseteq D_0$, we see $DESV(K_2)$ =(0,1, 2,0,2), $DESV(K_4)$ =(0,2,2,0,2). $DESV(K_2 \cap K_4) = DESV(\{u_5\})$ =(0,*,2,0,2). According to $DESV(K_2 \cap K_4)$ = (0,*,2,0,2), we may get a default definite decision rule: (0,*,2,0,2) $\rightarrow$ 0, ie, $(a,0) \wedge (b,*) \wedge (c,2) \wedge (e,0) \wedge (f,2) \rightarrow (d,0)$. Then $(a,0) \wedge (c,2) \wedge (e,0) \wedge (f,2) \rightarrow (d,0)$ is the further optimal form. The attribute *b* can take any value including 1 or 2, because $K_2 \cup K_4 = \{u_3, u_5, u_6\} \subseteq D_0$, $K_2 \setminus (K_2 \cap K_4) \subseteq D_0$ and $K_4 \setminus (K_2 \cap K_4) \subseteq D_0$ as well. So the final rule mined is: $(a,0) \wedge (c,2) \wedge (e,0) \wedge (f,2) \rightarrow (d,0)$ .We can say that its decision is stable , not sensible to values of *b*. These are three extreme situations for incomplete decision tables. These three extreme cases also give a heuristics for us to derive decision rules from incomplete decision tables on the base of different levels of granules by granulation theory. Mining rules based on join and meet blocks is just an example. But we obliged to see the related rules have to be dug at a high difficulty on other blocks or granules. That is beyond our discussion scope. We can induce some principles concerning only two maximal consistent blocks.

(1) Let $K_i, K_j \in U/TIM(C)$, $K_i \neq K_j$ and $K_i \cap K_j \neq \varnothing$. If there exists $D_r \in U/IND(\{d\})$ such that $K_i \subseteq D_r$ and $K_j \subseteq D_r$, then we must have an $a \in C$ such that $(a,*) \in DESV(K_i \cap K_j)$, $(a,*) \notin DESV(K_i \setminus K_i \cap K_j)$, $(a,*) \notin DESV(K_j \setminus K_i \cap K_j)$. If $(a,v') \in t' \in C(K_i \setminus K_i \cap K_j)$, $(a,v'') \in t'' \in C(K_j \setminus K_i \cap K_j)$, where $v' \neq *, v'' \neq *$ and $v' \neq v''$, then we can form specialized rules with the same decision *r* : one is with (a,v') in it, another is with (a,v'') in it, and even another is with $(a,*)$ in it. So they can be unified to form a default definite rule based on $DESV(K_i \cap K_j) \rightarrow D_r$ without a in it. That is, a can be omitted from the rule.

(2) Let $K_i, K_j \in U/TIM(C)$, $K_i \neq K_j$ and $K_i \cap K_j \neq \varnothing$. If there exists $D_r \in U/IND(\{d\})$ such that $K_i \subseteq D_r$ but $K_j \not\subseteq D_r$, then we must have some $a \in C$ such that $(a,*) \in DESV(K_i \cap K_j)$, and $(a,*) \notin DESV(K_j \setminus K_i \cap K_j)$. If $(a,*) \in t' \in C(K_i \cap K_j)$, $(a,v'') \in t'' \in C(K_j \setminus K_i \cap K_j)$ and $d(t'') \neq r$, where $v'' \neq *$, then we can form a specialized rule based on

$DESV(K_i \cap K_j) \to D_r$ with a restriction that a can not take value $v"$ and without a in the rule.

(3)Let $K_i, K_j \in U/TIM(C)$ $K_i \neq K_j$ and $K_i \cap K_j \neq \varnothing$. If for any $D_r \in U/IND(\{d\})$ such that $K_i \not\subseteq D_r$ and $K_j \not\subseteq D_r$,but there is a $D_l \in U/IND(\{d\})$ such that $K_i \cap K_j \subseteq D_l$, then we must have some $a \in C$ such that $(a,*) \in DESV(K_i \cap K_j)$, $(a,v') \in t' \in DESV(K_i \setminus K_i \cap K_j)$, $(a,v")$ $\in t"$ $\in DESV(K_j \setminus K_i \cap K_j)$, where $v' \neq *, v" \neq *$ and $v' \neq v"$ such that $\| t' \| \cap D_l = \varnothing$ and $\| t" \| \cap D_l = \varnothing$. In this case, we can obtain also another default definite rule with decision *l* without attribute *a* in it and *a* is forced not to take values in $\{v', v"\}$.

VI. Optimization by Discernibility Matrix and discernibility Function

Definition 6. Let $B \subseteq C$ and $\wedge_{a \in B}(a, v_a) \to (d, j)$ be a certainly existed and definite decision rule. If $B' \subseteq B$ and $\wedge_{a \in B'}(a, v_a) \to (d, j)$ is still a certainly existed and definite decision, then $\wedge_{a \in B'}(a, v_a) \to (d, j)$ is called better than $\wedge_{a \in B}(a, v_a) \to (d, j)$. If for any $B" \subset B'$, $\wedge_{a \in B"}(a, v_a) \to (d, j)$ is not a certainly existed and definite decision, then $\wedge_{a \in B'}(a, v_a) \to (d, j)$ is called a optimal decision rule and B' is called a reduct of $\wedge_{a \in B}(a, v_a) \to (d, j)$.

Definition 7. Let $S = (U, C \cup \{d\}, V, f)$ be an IDT, $K \in U/TIM(C)$, $D_i \in U/IND(\{d\}) = U/TIM(\{d\})$ and $K \subseteq D_i$. If $B \subseteq C$ is a minimal attribute subset and satisfies: $K^B \in U/TIM(B)$, $K^B \supseteq K \Rightarrow K^B \subseteq D_i$, then *B* is called a relative reduction of *C* with respect to *K*.

If *B* is called a relative reduction of *C* with respect to *K*, then $\wedge_{a_i \in B}(a_i, v_i) \to (d, i)$ is an optimal rule to $\wedge_{a_i \in C(K)}(a_i, v_i) \to (d, i)$.

Let $\alpha(u, w) = \{a \mid a \in C, * \neq a(u) \neq a(w) \neq *\}$ .If $\alpha(u, w) = \varnothing$, *u* and *w* are not *c*-discernible; if $c \in \alpha(u, w)$, then *u* and *w* are *c*-discernible, or you can say that one can use *c* to discern *u* and *w*. $M = (\alpha(u, w))_{n \times n}$ is the discernible matrix of system *S*, where $U = \{u_1, u_2, ..., u_n\}$. $\vee \alpha(u, w)$ represents the Boolean disjunction of all attributes in $\alpha(u, w)$. If $\alpha(u, w) = \varnothing$, then $\vee \alpha(u, w) = 1$.

Theorem 6. Let $S = (U, C \cup \{d\}, V, f)$ be an incomplete decision table $K \in U/TIM(C)$, $D_i \in U/IND(\{d\}) = U/TIM(\{d\})$, $K \subseteq D_i$, $B \subseteq C$. Then for $\forall K^B \in U/TIM(B)$, $K^B \supseteq K, K^B \subseteq D_i$ if and only if, for any $u \in U \setminus D_i$, $\exists w \in K$ such that $\alpha(u, w) \cap B \neq \varnothing$.

Proof. Suppose $\forall K^B \in U/TIM(B), K^B \supseteq K$, $K^B \subseteq D_i$. If $\exists u \in U \setminus D_i$ such that for any $w \in K$, $\alpha(u, w) \cap B = \varnothing$, then $(u, w) \in SIM(B)$. Since *w* is arbitrary in *K*, we have $(K \cup \{u\})^2 \subseteq TIM(B)$. Extending $K \cup \{u\}$ to maximal, we get $K^B \in U/TIM(B)$ such that $K \cup \{u\} \subseteq K^B$. From the given condition and $K \subseteq K^B$, we obtain $K^B \subseteq D_i$. Therefore, $u \in D_i$. This contradicts $u \in U \setminus D_i$. So $\alpha(u, w) \cap B = \varnothing$ is void. It proves that for any $u \in U \setminus D_i$, $\exists w \in K$ such that $\alpha(u, w) \cap B \neq \varnothing$.

Conversely, assume that for any $u \in U - D_i$, $\exists w \in K$ such that $\alpha(u, w) \cap B \neq \varnothing$. If there exists some $K^B \in U/TIM(B)$ such that $K \subseteq K^B$ but $K^B \not\subseteq D_i$, then $\exists u \in K^B \setminus D_i$ holds. From $u \in K^B \setminus D_i$ and the given condition, $\exists w \in K$ such that $\alpha(u, w) \cap B \neq \varnothing$. However, we must have $(u, w) \in TIM(B)$ from $u \in K^B \setminus D_i$, $w \in K$ and $K \subseteq K^B$, and moreover $\alpha(u, w) \cap B = \varnothing$. That is a contradiction. So for $\forall K^B \in U/TIM(B)$, $K^B \supseteq K, K^B \subseteq D_i$.

Definition 8. Let $K \in U/TIM(C), K \subseteq D_i$. $\Delta(K, D_i) = \wedge_{u \in U \setminus D_i}(\vee_{w \in K}(\vee \alpha(u, w)))$ is called a discernible function of K with respect to D_i.

Theorem 7. Any attribute subset corresponding to each conjunction in its minimal disjunction normal form of $\Delta(K)$ forms a relative reduct of K .The proof of Theorem 7 can be performed by using Boolean inference theory and the former theorem according to the definition, similar to that in [15].

Example 3. All optimal certain decision rules from incomplete decision table in Table 1.

At first, we form the discernible matrix M as in Fig 1. Then, from $K_1 \subseteq D_1$, $K_2 \subseteq D_0$, $K_4 \subseteq D_0$, $K_5 \subseteq D_0$, $K_6 \subseteq D_2$, we calculate the reduced rules according to discernible matrix respectively as follows.

(1) For $K_1 \subseteq D_1, K_1 = \{u_1, u_2\}, D_1 = \{u_1, u_2, u_4, u_7\}$, we have $U \setminus D_1 = \{u_3, u_5, u_6, u_8, u_9, u_{10}, u_{11}\}$. $\Delta(K_1) =$

$\wedge_{u\in U\setminus D_1}(\vee_{w\in K_1}(\vee\alpha(u,w)))=(a\wedge f)\vee(b\wedge f)\vee(e\wedge f)$. Relative reducts for K_1 are: $\{a,f\},\{b,f\},\{e,f\}$. So, optimal rules to default definite rule (a certain rule) $(a,1)\wedge(b,0)\wedge(c,1)\wedge(e,2)\wedge(f,0)\to(d,1)$ are:

$(a,1)\wedge(f,0)\to(d,1)$; $(b,0)\wedge(f,0)\to(d,1)$;
$(e,2)\wedge(f,0)\to(d,1)$.

$$\begin{bmatrix}
\varnothing \\
\varnothing & \varnothing \\
\{abf\} & \{abcf\} & \varnothing \\
\{bef\} & \{bef\} & \{abf\} & \varnothing \\
\{aef\} & \{aef\} & \varnothing & \{aef\} & \varnothing \\
\{abe\} & \{abce\} & \{b\} & \{ae\} & \varnothing & \varnothing \\
\{e\} & \{e\} & \{a\} & \varnothing & \{ae\} & \{ae\} & \varnothing \\
\{bf\} & \{b\} & \{abcf\} & \{e\} & \{acef\} & \{ace\} & \{e\} & \varnothing \\
\{f\} & \{f\} & \{cf\} & \varnothing & \{cf\} & \{c\} & \varnothing & \varnothing & \varnothing \\
\{bf\} & \{bcf\} & \{bcf\} & \varnothing & \{cf\} & \{c\} & \varnothing & \{c\} & \{c\} & \varnothing \\
\{abe\} & \{abe\} & \{abcf\} & \{abef\} & \{acf\} & \{abc\} & \{ae\} & \{abef\} & \{f\} & \{bcf\} & \varnothing
\end{bmatrix}$$

Fig. 1 Discernibility Matrix of Table 1

(2) For $K_2\subseteq D_0$, $K_2=\{u_3, u_5\}$, $D_0=\{u_3, u_5, u_6, u_{10}, u_{11}\}$, $U\setminus D_0=\{u_1,u_2,u_4,u_7,u_8,u_9\}$. So we have $\Delta(K_2)=\wedge_{u\in U\setminus D_0}(\vee_{w\in K_2}(\vee\alpha(u,w)))=(a\wedge c)\vee(a\wedge f)\vee(c\wedge e)\vee(e\wedge f)$. Relative reducts for K_2 are: $\{a,c\}$, $\{a,f\}$, $\{c,e\}$, $\{e,f\}$. Optimal rules to the default definite rule (a default rule) $(a,0)\wedge(b,1)\wedge(c,2)\wedge(e,0)\wedge(f,2)\to(d,0)$ are:

$(a,0)\wedge(c,2)\to(d,0)$; $(a,0)\wedge(f,2)\to(d,0)$;
$(c,2)\wedge(e,0)\to(d,0)$; $(e,0)\wedge(f,2)\to(d,0)$.

(3) For $K_4\subseteq D_0$, $K_4=\{u_5, u_6\}$, $D_0=\{u_3, u_5, u_6, u_{10}, u_{11}\}$. So $\Delta(K_4)=\wedge_{u\in U\setminus D_0}(\vee_{w\in K_4}(\vee\alpha(u,w)))=(a\wedge c)\vee(a\wedge f)\vee(c\wedge e)\vee(e\wedge f)$. Relative reducts for K_4 are: $\{a,c\}$, $\{a,f\}$, $\{c,e\}$, $\{e,f\}$. Optimal rules to the default definite rule (a default rule) $(a,0)\wedge(b,2)\wedge(c,2)\wedge(e,0)\wedge(f,2)\to(d,0)$ are:

$(a,0)\wedge(c,2)\to(d,0)$; $(a,0)\wedge(f,2)\to(d,0)$;
$(c,2)\wedge(e,0)\to(d,0)$; $(e,0)\wedge(f,2)\to(d,0)$.

(4) For $K_5\subseteq D_0$, $K_5=\{u_{11}\}$, $D_0=\{u_3, u_5, u_6, u_{10}, u_{11}\}$, So $\Delta(K_5)=\wedge_{u\in U\setminus D_0}(\vee_{w\in K_5}(\vee\alpha(u,w)))=(a\wedge f)\vee(e\wedge f)$. Relative reducts are: $\{a,f\}$, $\{e,f\}$. Optimal rules for default definite rule (a certainly existed definite decision rule) $(a,2)\wedge(b,1)\wedge(c,1)\wedge(e,0)\wedge(f,0)\to(d,0)$ are:

$(a,2)\wedge(f,0)\to(d,0)$; $(e,0)\wedge(f,0)\to(d,0)$.

(5) For $K_6\subseteq D_2$, $K_6=\{u_8, u_9\}$, $D_2=\{u_8, u_9\}$, $U\setminus D_2=\{u_1,u_2,u_3,u_4,u_5,u_7,u_{10},u_{11}\}$. Thus we have $\Delta(K_6)=\wedge_{u\in U\setminus D_2}(\vee_{w\in K_6}(\vee\alpha(u,w)))=(b\wedge c)\vee(c\wedge f)=\{b,c\}\vee\{c,f\}$. Relative reduct for K_6 are: $\{b,c\},\{c,f\}$. Optimal rulesto the default definite rule (a certain rule) $(a,1)\wedge(b,2)\wedge(c,1)\wedge(e,2)\wedge(f,1)\to(d,2)$ are:

$(b,2)\wedge(c,1)\to(d,2)$; $(c,1)\wedge(f,1)\to(d,2)$.

Similarly, we can use discernibility function to find relative reducts of rules derived by join and meet blocks.

(6) For $K_3\cap K_7=\{u_4,u_7\}\subseteq D_1$, we can also calculate discernibility function: $\Delta(K_3\cap K_7)=\wedge_{u\in U\setminus D_1}(\vee_{w\in K_3\cap K_7}(\vee\alpha(u,w)))=\{a,e\}\vee\{b,e\}\vee\{e,f\}$. Thus optimal default definite rule (also a certainly decision rule) $(a,1)\wedge(b,2)\wedge(e,1)\wedge(f,1)\to(d,1)$ and $\forall u\in U$, $c(u)\in V_c\setminus\{0,1\}$ are: $(a,1)\wedge(e,1)\to(d,1)$; $(b,2)\wedge(e,1)\to(d,1)$; $(e,1)\wedge(f,1)\to(d,1)$. Note that for $\forall u\in U,\ c(u)\in V_c\setminus\{0,1\}$.

(7) Similarly, $K_3\cap K_6=\{u_9\}\subseteq D_2$ we can compute the discernibility function $\Delta(K_3\cap K_6)=\wedge_{u\in U\setminus D_2}(\vee_{w\in K_3\cap K_6}(\vee\alpha(u,w)))=c\wedge f$. So the rule: $(\mathrm{c},1)\wedge(\mathrm{f},1)\to 2$ with $e(u)\neq 1$ for any u does not required to reduced.

(8) For $K_2\cup K_4=\{u_3,u_5,u_6\}\subseteq D_0$, the discernibility function is computed $\Delta(K_2\cup K_4)=\wedge_{u\in U\setminus D_0}(\vee_{w\in K_2\cup K_4}(\vee\ \alpha(u,w)))=\{a,c\}\vee\{a,f\}\vee\{c,e\}\vee\{e,f\}$. Thus optimal rules for default definite rule (a default rule) $(a,0)\wedge(c,2)\wedge(e,0)\wedge(f,2)\to(d,0)$ are:

$(a,0)\wedge(c,2)\to(d,0)$; $(a,0)\wedge(f,2)\to(d,0)$;
$(c,2)\wedge(e,0)\to(d,0)$; $(e,0)\wedge(f,2)\to(d,0)$.

Summarizing the above optimal default rules, we can eliminate repeat ones and remain useful ones.

VII. An Incremental Algorithm to Mining Default Definite Decision Rules

With new objects appending to the IDT, the former default definite rules have to be updated to suit for a new data circumstance. When new objects are added to an IDT, it is an efficient approach to accept modifying the existing rules or few new rules as current rules, instead of regenerating them wholly. In this section, an incrementally updating present default definite decision rule algorithm is presented.

Let $K_i(i=1,2,\cdots,n)$ be all maximal consistent blocks of incomplete decision table S, $D_j(j=1,2,\cdots,m)$ be all its decision classes. When a new object , at least one of its attribute values is different from the corresponding attribute value of each object, is appended to the IDT, the algorithm is implemented by distinguishing the following cases:

Case 1. $u \notin K_i (i=1,2,\cdots,n)$ and $u \notin D_j (j=1,2,\cdots,m)$. A new maximal consistent block K_{n+1} and a new decision class D_{m+1} are constituted by u. $K_{n+1} = D_{m+1} = \{u\}$. $K_{n+1} \subseteq D_{m+1}$. Create a default definite decision rule $DESV(K_{n+1}) \to DESV(D_{m+1})$, ie, $f_C(u) \to d(u)$. The original set of remains unchangeable.

Case 2. $u \notin K_i (i=1,2,\cdots,n)$ and $u \in D_j (j \in \{1,2,\cdots,m\})$. A new maximal consistent block $K_{n+1} = \{u\}$ is formed. Construct a new default definite decision rule: $DESV(K_{n+1}) \to DESV(D_j)$,ie, $f_C(x) \to DESV(D_j) = d(u)$, ie , $f_C(u) \to d(u)$ because $K_{n+1} = \{u\} \subseteq D_j$ and we assumed that missing value is not included at decision attribute, and $DESV(D_j) = d(u)$.

Case 3. $u \in K_{i_1}, K_{i_2}, \cdots, K_{i_r} (i_1, i_2, \cdots, i_r \in \{1,2,\cdots,n\}, r \geq 1)$ and $u \in D_j (j \in \{1,2,\cdots,m\})$. (Note: $K_1, K_2, \cdots, K_n$ only form a overlapped covering , not a partition; $u \in K_{i_l} (l=1,2,\cdots,r)$ means it is compatible with each object in K_{i_l}). We have to check whether $K_{i_l} \cup \{u\} \subseteq D_j$ $(l=1,2,\cdots,r)$. If it is hold, we build a new rule $DESV(K_{i_l} \cup \{u\}) \to j$, add it to the new rule set. If it is not hold, we do not build such kind of rules. Then consider meet block based default definite decision rule. Obviously, $K_{i_l} \cap K_{i_k} \neq \varnothing (k \neq l, k, l \in \{1,2,\cdots,r\})$, so rules based on meet blocks $K_{i_l} \cap K_{i_k}$ can be created depending on whether or not $K_{i_l} \cap K_{i_k} \subseteq D_j$. Other rules in the original rule set remain unchanged.

Case 4. $u \in K_{i_1}, K_{i_2}, \cdots, K_{i_r} (i_1, i_2, \cdots, i_r \in \{1,2,\cdots,n\}, r \geq 1)$ and $u \notin D_j (j \in \{1,2,\cdots,m\})$. A new decision class D_{m+1} is constituted with only *u* in it. If *r*=1, nothing has to be done because $K_1 \not\subseteq D_{m+1}$ and other else c situations were considered before. If *r*>1, we have to check whether or not for any $i_p, i_q (1 \leq i_p, i_q \leq r)$, $K_{i_p} \cap K_{i_q} \subseteq D_{m+1} = \{u\}$. If not, nothing has to be done. If it holds for example K_{i_p}, K_{i_q} such that $K_{i_p} \cap K_{i_q} \subseteq D_m$, really speaking, $K_{i_p} \cap K_{i_q} = D_{m+1}$ for D_{m+1} is a singleton subset, then we have to add a new default definite decision rule $DESV(K_{i_p} \cap K_{i_q}) = f_C(u) \to DESV(D_{m+1})$, ie, $f_C(u) \to d(u)$. All other default definite rules in the original rule set need not updated.

After appending an object into the IDT and finishing rules updating, we need to add a new row to the discernibility matrix because a lower triangle matrix is constructed by us and it is for continuing appending objects later.

TABLE II
A NEW OBJECT TO BE INSERTED

	a	b	c	e	f	d
u	*	2	1	*	1	1

Example 4. Let a new object u shown in Table 2 is appended to the IDT in Table 1. We can see, u_4 and u are compatible, u_9 and u are also compatible, u_7 and u are also compatible, u_8 and u are also compatible. So $K_3 = \{u_4, u_7, u_9\}$ has to be updated to $K_3' = K_3 \cup \{u\} = \{u_4, u_7, u_9, u\}$, $K_6 = \{u_8, u_9\}$ is updated by $K_6' = K_6 \cup \{u\} = \{u_8, u_9, u\}$. $u \in D_1$ because $d(u)=1$. So D_1 has to be updated by $D_1' = D_1 \cup \{u\} = \{u_1, u_2, u_4, u_7, u\}$. It belongs to the Case 3. $DESV(K_3') = DESV(K_3)$ =(1,2,1,1,1), $DESV(K_6') = DESV(K_6)$ = (1,2,1,2,1).

TABLE III
COMPATIBLE OBJECTS

U	a	b	c	e	f	d
u	*	2	1	*	1	1
u_4	1	2	*	1	1	1
u_7	1	*	*	1	*	1
u_8	1	2	1	2	1	2
u_9	*	*	1	*	1	2

In the following, we consider which rules have to be changed. According to meet block and $K_3 \cap K_7 \subseteq D_1$ in the original, we got a rule. Now $K_3' \cap K_7 = K_3 \cap K_7 = \{u_4, u_7\} \subseteq D_1$ is still hold and $DESV(K_3' \cap K_7) = DESV(K_3 \cap K_7)$. So the rule by meet block $K_3' \cap K_7$ needs not to be updated. In the original rule generation, $K_6 \subseteq D_2 = \{u_8, u_9\}$, but now $K_6' = \{u_8, u_9, u\} \not\subseteq D_2$. Therefore the original default definite rule $DESV(K_6) \to DESV(D_2) = 2$ has to be removed because there is a conflict between $DESV(D_2) = (d, 2)$ and $d(u)=1$. On the other hand, $K_6' \cap K_3' = \{u, u_9\} \neq \varnothing$, but $\{u, u_9\} \not\subseteq D_1', D_0, D_2$. So we do not need to use it to produce a meet block based rule. In addition, the last thing for us to do is to add a new row to the discernibility matrix with $\alpha(u, u_i)(i=1,2,\cdots,11)$ and $\alpha(u,u) = \varnothing$). To save space, it is omitted here.

VIII. CONCLUSIONS

After assuming maximal consistent blocks to be basic blocks and acting some set operations on some of them, other kinds of blocks such as tolerance class, join and meet blocks or compatible kernels or tolerance kernels may be produced. They provide us with possibilities to acquire some other kinds of finer and precision domain knowledge from incomplete decision table different from that in [3,5,7,14] .

Laying on maximal consistent blocks and join and meet blocks, some default definite decision rules are generated by our approach. Our deep knowledge is logically similar to the inference rule in default logic. That is the reason why we call them default definite rules. So they are meaningful in facing complicated incomplete decision tables.

Incremental algorithm proposed here can improve efficiency of mining knowledge from IDT when training examples are appended one by one, without performing regeneration process on the whole IDT. The future work or glorious task may be to acquire other rules based on other granules different from join and meet blocks here.

REFERENCES

[1] W.L. Chen, J.X. Cheng, C.J. Zhang, "A Generalized Model of Rough Set Theory Based on Compatibility Relation", Computer Engineering and Applications", Vol.40,No. 4, 2004,pp.26-28

[2] T.P. Hong, L.H. Tseng, S.L Wang, "Learning rules from incomplete training examples by rough sets", Expert Systems with Applications Vol. 22,Issue 4,2002, pp.285-293

[3] T.R. Li, N. Yang, Y. Xu, J. Ma, "An Incremental Algorithm for Mining Classification Rules in Incomplete Information System", Fuzzy Information, Processing NAFIPS '04. Vol. 1,2004, pp. 446- 449

[4] Y. Li, S. C.K. Shiu, S.K. Pal, J.N.K. Liu, "A Fuzzy-Rough Mothod for Concept Based Document Expansion", in Tsumoto,S. Slowinski,R. Komorowski,J. Grzymala-Busse,J.W.(Ed.), Rough Sets and Current Trends in Computing, Springer-Verlag Berlin Heidelberg,LNAI 3066,2004,pp.699-707

[5] G. Li, X. Zhang, "Decomposition of Rough Set Based on Similarity Relation", Computer Engineering and Applications, Vol.40,No.2,2004, pp.85-86 and pp.119

[6] Y. Leung, D. Li , "Maximal Consistent Block Technique for Rule Acquisition in Incomplete Information Systems" Information Sciences, Vol.153, No.1, 2003, pp: 85-106

[7] K. Funakoshi, T. B. Ho,"Information Retrieval by Rough Tolerance Relation", Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery, November 6-8, Tokyo, Japan, 1996,pp.31-35

[8] M. Kryszkiewicz, "Rough Set Approach to Incomplete Information Systems", J. of Information Sciences,Vol.112, No.1,1998, pp.39-49

[9] M. Kryszkiewicz, "Rules in Incomplete Information Systems",J. of information Sciences,Vol.113,1999, pp.271-292

[10] J.S. Mi, W.Z. Wu, W.X. Zhang, "Approaches to Knowledge Reduction Based on Variable Precision Rough Set Model", Information Sciences, Vol.159,No.3-4,2004,pp.255-272

[11] Z. Pawlak, "Rough sets and intelligent data analysis",Information Sciences. Vol.147,Issue 1-4, 2002 ,pp.1-12

[12] J. Stefanowski, A. Tsoukiàs, "Incomplete Information Tables and Rough Classification", J. Computational Intelligence, Vol. 11, No.3,2001,pp.545-566

[13] C. Wu,, X.B. Yang, "Information Granules in General and Complete Covering",Proceedings of 2005 IEEE International Conference on Granular Computing, Vol.2 , 2005,pp.675-678

[14] C. Wu,, X.H. Hu,J.Y Yang, X.B. Yang ,"Expanding Tolerance RST Models Based on Cores of Maximal Compatible Blocks", Rough Sets and Current Trends in Computing, Springer-Verlag Berlin Heidelberg,LNAI 4259, 2006,pp.235-243

[15] Y.Y. Yao, "Neigborhood systems and approximate retrieval",J. of Information Sciences,Vol.116,No.23,2006,pp.3431-3452

Chen Wu, male ,born in 1962, received the B.S. degree in Numerical Computing from Wuhan University in 1982 and M.S. degrees in Applied Mathematics in Huazhong University of Science and Technology in 1985 both in P.R.China. During 1988-1990, he stayed in Artificial Intelligent Laboratory at High Mechanical and Electrical Institute in Bulgaria to study Artificial Intelligence and Parallel Processing. He now is with Computer Department at Nanjing University of Science and Technology as a candidate for Ph.D in part-time and with School of Electronics and Information at Jiangsu University of Science and Technology as a professor. His interest direction is rough set theory and data processing including data mining and knowledge discovery. E-mail: wuchenzj@sina.com or wuchenzj@gmail.com.

Jingyu Yang,, male, born in 1941, a supervisor for candidates for Ph.D , working as a a senior professor at the Department of Computer Science in Nanjing University of Science and Technology. His main study fields are: theory and application of pattern recognition, computer application, intelligent robot technology, data fusion. E-mail: yangjy@mail.njust.edu.cn.