

On Novelty Evaluation of Potentially Useful Patterns

Ying Xie, Manmathasivaram Nagarajan, Vijay V. Raghavan, Hisham Haddad

Abstract - As is generally accepted, the most important feature that a Knowledge Discovery in Database (KDD) system must possess is, to be able to discover patterns that are “novel” and “potentially useful”. In order to allow KDD systems to make novelty and potential usefulness judgment, we extend our former work on discovering “potentially useful” patterns by proposing a formal definition of “novelty” based on the same probabilistic logic foundation we used to define “potential usefulness”. Furthermore, a tractable algorithm is proposed that is capable of discovering all novel and potentially useful patterns from databases based upon limited accessible information.

Index Terms— KDD, Data Mining, Logic Foundation, Novel Patterns, Previously Unknown, Surprisingness

I. INTRODUCTION

It is widely accepted that knowledge Discovery in Databases (KDD) is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. We believe that formal definitions of potential usefulness and novelty are necessary in order for the KDD system to make novelty and potential usefulness judgments efficiently and effectively. Therefore, in [21], we utilized Bacchus’ probabilistic logic to categorize KDD and proposed the logical definitions for potential usefulness and previous unknownness of a pattern. Furthermore, we designed a tractable algorithm called DAPUP (Discover All Potentially Useful Patterns) that is capable of discovering all potentially useful patterns with the same time complexity as association mining [22]. In this paper, we continue our work by studying the “novelty” feature of a discovered pattern. In our view, the novelty of a pattern should be evaluated at two different levels. The first level deals with the requirement that the discovered patterns are at least previously unknown, while, the second level of evaluation additionally requires that, the previously unknown patterns have a certain degree of surprisingness or unexpectedness associated with them. According to this view, the word “novel” implies that the KDD system may discover certain types of patterns that go beyond what the user can specify (e.g. association rules, exception rules,

negative rules, and others). We provide a formal definition of a pattern’s degree of surprisingness. Combining the definitions of previous unknownness and surprisingness, we obtain the definition of novelty. It should be noted that all of the definitions that we proposed are completely within the realms of the expressiveness of Bacchus’ probabilistic logic language.

In this paper, we also provided the tractable algorithm to extract novel and potentially useful patterns from databases. Therefore, a complete KDD process can be described as shown in figure 1. Based on the data set itself, the KDD process can extract all potentially useful patterns embedded in the data set by using the DAPUP algorithm. If previously known knowledge base is available, Bacchus’ logic deductive program can be applied to automatically determine previously unknown patterns. Finally, if taxonomy information is provided, the KDD system can automatically conduct novelty evaluation.

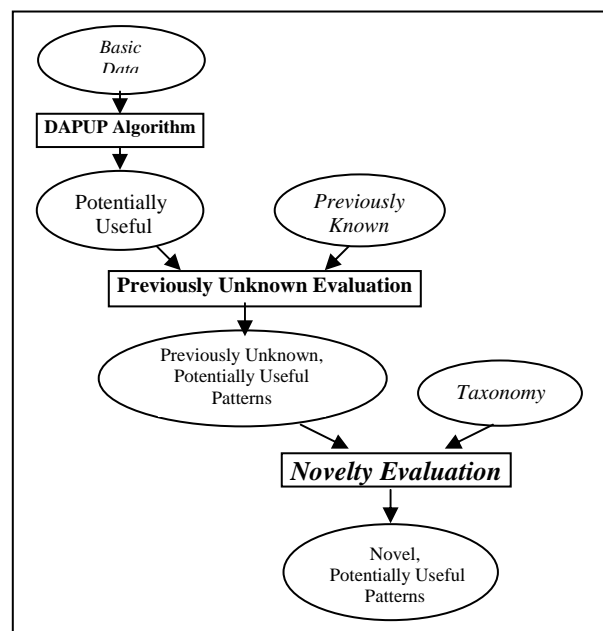


Figure 1: The Proposed KDD Process.

Ying Xie is with the Dept. of Computer Science and Information Systems, Kennesaw State University, GA, USA (e-mail: yxie2@kennesaw.edu); **Manmathasivaram Nagarajan** graduated from the center for advanced computer studies, University of Louisiana, Lafayette, LA, USA (e-mail: mxn7261@cacs.louisiana.edu); **Vijay V. Raghavan** is with the center for advanced computer studies, University of Louisiana, Lafayette, LA, USA (e-mail: raghavan@cacs.louisiana.edu); **Hisham Haddad** is with the Dept. of Computer Science and Information Systems, Kennesaw State University, GA, USA (e-mail: hhaddad@kennesaw.edu).

The rest of the paper is organized as follows. In section 2, we highlight the unique contributions of this work by comparing it with related work. A detailed description of our work is then provided in sections 3 and 4 and 5. First, we illustrate how Bacchus’ probabilistic logic can be used to represent a transaction table in section 3, followed by a briefly introduction of our former work on potential usefulness evaluation in section 4. In section 5, we give a formal definition for “novel pattern” and propose a tractable

procedure to discover novel and potentially useful patterns from database. In section 6, we extend our work to discover “generalized novel and potentially useful rules”. Finally in section 7, we conclude the paper and envision some important future extensions.

II. RELATED WORK

In [6, 8], the unexpectedness of a discovered pattern is evaluated by utilizing certain types of symbolic distances between a discovered pattern and the existing beliefs (or already known patterns). The problem with the two cited approaches is that they only calculate the distance at the syntactic level, but fail to take into consideration the semantic relationship between items or attributes involved in the examined pattern. For instance, the following two patterns, *Mattress* → *Pillows* and *WeddingRing* → *Pillows*, may be deemed to have the same degree of unexpectedness. However, we feel that the latter one should be more surprising than the former one, because it is well known that mattresses and pillows are semantically closer to each other than wedding ring and pillows. Likewise, existing beliefs are utilized to evaluate the unexpectedness of discovered patterns in [5]. However, the manner that existing beliefs are used in [5] is different from that of [6] and [8]. In [6] and [8], a discovered pattern is reported as unexpected if no similar belief is presented, while, in [5], a discovered pattern can be reported as unexpected only if it’s “contradicted” belief is provided. In this sense, the work in [5] can be viewed as a variation of the template-based approach, which conducts the discovery in a “retrieval manner”.

In this paper, we divide the measurement of novelty into two levels. The first level determines if a pattern is previously unknown. The user can model the already known patterns (or beliefs) by using the Bacchus’ probabilistic logic language. As a result, a discovered pattern can be judged as previously unknown if it cannot be logically inferred from the already known patterns. The second level of our measure evaluates a pattern’s degree of surprisingness. We take advantage of the available taxonomy information to calculate the semantic distance between the antecedent and the consequent of the examined pattern. The greater the semantic distance, the more surprising is the corresponding pattern.

III. BACCHUS PROBABILISTIC LOGIC

Given that the scope of our discussion is limited to statistical knowledge discovery, which is one of the most active themes in KDD, we found that Bacchus’s probabilistic logic provides a good foundation for formalizing the KDD process. It augments first order logic with a unified formalism to represent and reason with statistical probability. For example, its first order logic component can be used to model facts and relational structures, while its statistical probability component can be used to represent and reason with patterns. In this section, we briefly illustrate how to use Bacchus’ probabilistic logic to represent facts and patterns in a transaction table (For more information on Bacchus’ probabilistic logic and how it formally represents an information table, please refer to [1, 2]).

The following example of transaction table (Table 1) is used to explain how to model a transaction table in Bacchus’ probabilistic logic. The transaction table includes ten transactions that describe the sales of items namely *Utensils*, *Appliances*, *Couch*, *Table*, *Mattress*, *Pillows*, *WeddingRings*, *FamilyRings*.

We build a semantic structure for this transaction table and denote it as $\mathbf{M} = \langle \mathbf{O}, \mathcal{V}, \mu \rangle$, where

Table1: Transaction Table

TID	Items
T1	Mattress, Utensils, Pillows, Table, Appliance
T2	Table, Appliance, Utensils
T3	Mattress, Table, Couch
T4	Utensils, Table, Couch
T5	Utensils, Couch, Appliance
T6	Mattress, Utensils, Pillows, Couch, Appliance
T7	Mattress, Table, Appliance, Couch,
T8	Mattress, Table, Appliance
T9	Mattress, Utensils, Couch, Appliance
T10	Mattress, Table, Couch
T11	Table, Couch, Utensils
T12	Mattress, Utensils, Pillows, WeddingRing
T13	Mattress, Utensils, Pillows, FamilyRings
T14	Mattress, WeddingRing, FamilyRings
T15	Utensils, WeddingRing, FamilyRings

- $\mathbf{O} = U = \{T1, T2 \dots T15\}$;
- The set of unary object predicate symbols is $\{\text{Utensils}, \text{Appliances}, \text{Couch}, \text{Table}, \text{Mattress}, \text{Pillows}, \text{WeddingRings}, \text{FamilyRings}\}$; and by interpretation function \mathcal{V} , $\text{Utensils}^{\mathcal{V}} = \{T1, T2, T4, T5, T6, T9, T11, T12, T13, T15\}$; $\text{Couch}^{\mathcal{V}} = \{T3, T4, T5, T6, T7, T9, T10, T11\}$; $\text{Mattress}^{\mathcal{V}} = \{T1, T3, T6, T7, T8, T9, T10, T12, T13, T14\}$ and so on; We add two more special predicate symbols \emptyset and F , such that $\emptyset^{\mathcal{V}} = \phi$, and $F^{\mathcal{V}} = \{Ti : Ti \in \mathbf{O}\}$.
- By discrete probability function μ , for every transaction $Ti \in \mathbf{O}$, $\mu(Ti) = 1/15$; and for any subset $A \subseteq \mathbf{O}$, $\mu(A) = \sum_{Ti \in A} \mu(Ti)$ and $\mu(\mathbf{O}) = 1$.
- By interpretation function \mathcal{V} , numeric predicate symbols such as $>$, $=$, $<$, and numeric function symbols such as $+$, $-$, \times , \min , \max get the proper meaning.

Now, based on the semantic structure \mathbf{M} we build, some truth assignments of formulas and interpretations of terms shown as follow:

- 1) $\mathbf{M} \models \text{Utensils}(T1)$, because $T1 \in \text{Utensils}^{\mathcal{V}}$;
- 2) $\mathbf{M} \not\models \text{Utensils}(T3)$, because $T3 \notin \text{Utensils}^{\mathcal{V}}$;
- 3) $\mathbf{M} \not\models \emptyset(Ti)$, for any $Ti \in \mathbf{O}$;
- 4) $\mathbf{M} \models F(Ti)$, for any $Ti \in \mathbf{O}$;
- 5) $[\text{Utensils}] = \mu\{Ti : \mathbf{M} \models \text{Utensils}(Ti)\} = \mu\{T1, T2, T4, T5, T6, T9, T11, T12, T13, T15\} = 10/15$;

- 6) $\mathbf{M} \models [\text{Utensils}] = 10/15;$
- 7) $[\text{Pillows} / \text{Utensils}] = [\text{Pillows} \wedge \text{Utensils}] / [\text{Utensils}] = 4/10;$
- 8) $\mathbf{M} \models [\text{Pillows} / \text{Utensils}] = 4/10;$

The first formula represents one of the facts expressed by the transaction table; while formula 8 represents the logic expression of one of the patterns embedded in the transaction table.

Definition 3.1: Concept

If P is an object predicate symbol, P is a *concept*; it is also called *atomic concept*, if $P \neq F$ and $P \neq \emptyset$.

Both \emptyset and F are called *special concepts*;

If P, Q are concepts, $P \wedge Q$ is also a concept, where binary operator \wedge is defined as follows:

- $(P \wedge Q)^\psi = P^\psi \cap Q^\psi;$
- $P \wedge P = P;$
- $P \wedge Q = Q \wedge P;$
- $P \wedge (Q \wedge R) = (P \wedge Q) \wedge R;$
- $P \wedge F = P;$
- $P \wedge \emptyset = \emptyset.$

If a concept P is neither an atomic concept nor a special concept, it is called a *composite concept*.

Example 3.1

Based on the semantic structure built from table 1, we have:
Atomic concept: Utensils, Appliances, Couch, Table, Mattress, Pillows, WeddingRings, FamilyRings

Composite concept: Utensils \wedge Appliances, Utensils \wedge Appliances \wedge Couch, Mattress \wedge Pillows, ...

Special concepts: \emptyset, F

Now, let $C = \{P_1, P_2, \dots, P_m, F, \emptyset\}$ denote all the atomic concepts and special concepts defined on \mathcal{O} . The \wedge -closure C^* is defined to be the minimum set containing all the concepts in C and is closed under \wedge . Now we define the binary relation \leq on C^* , such that for any pair of concepts $Q_i, Q_j \in C^*$, we have:

$$Q_i \leq Q_j \Leftrightarrow Q_i = Q_j \wedge P,$$

where P can be any concept. The tuple $\langle C^*, \leq \rangle$ is a complete lattice that we call *concept lattice*.

Example 3.2

Let's continue with the transaction table. The set of all atomic concepts is: $C = \{\text{Utensils, Appliances, Couch, Table, Mattress, Pillows, WeddingRings, FamilyRings}\}$. And we have the following concept lattice $\langle C^*, \leq \rangle$ (Figure 2.), each node of which represents a concept.

Definition 3.2: Pattern

Let P and Q be two concepts, and $r \in \mathbf{R}$. We call formula $[P|Q] = r$ a pattern, iff $P = \bigwedge_k a_k$ and $[Q] \geq n$, where $k \geq 1$, each a_k is an atomic concept that does not appear in Q , and parameter n is called *noise controller*.

Example 3.3

If noise controller $n = 3$, then we have the following patterns:

- $[\text{Couch} | \text{Table}] = 4/7,$ $[\text{Pillows} | \text{Utensils}] = 4/10,$ $[\text{Pillows} | \text{Utensils} \wedge \text{Mattress}] = 4/5,$ $[\text{Mattress} | \text{Couch} \wedge \text{Table}] = 3/4 \dots$

Definition 3.3: Validity, Implicitness of Pattern

A pattern \mathcal{P} is *valid* by \mathbf{M} iff $\mathbf{M} \models \mathcal{P}$.

If formula $[P|Q] = r$ is a pattern, it is called *implicit Pattern* by \mathbf{M} iff it is valid by \mathbf{M} and $Q \neq F$.

Intuitively, an implicit pattern cannot be obtained by just querying the database using only one standard SQL statement.

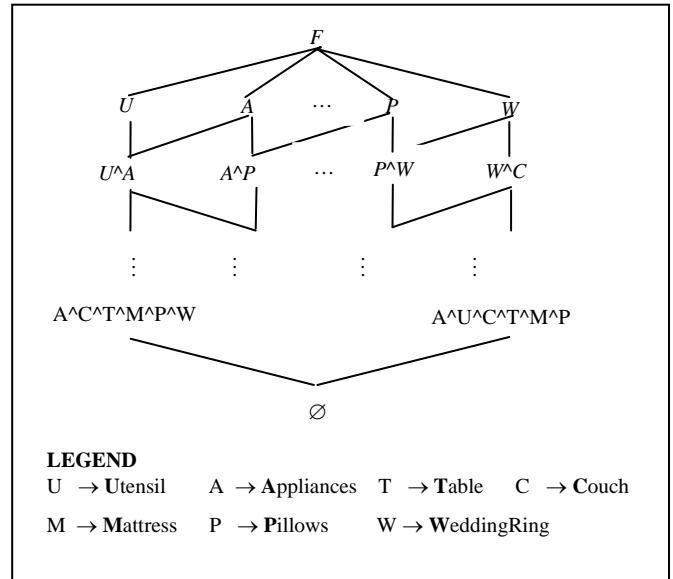


Figure 2. Concept Lattice.

IV. OUR FORMER WORK ON POTENTIAL USEFULNESS EVALUATION

Based on the probabilistic categorization of the database, in [22] we gave the formal definition of potentially useful pattern as follows.

Definition 4.1: Potentially Useful Pattern

Let Q be any concept except F and \emptyset on the concept lattice. Let $A = \bigwedge_k a_k$, where a_k is the atomic concept that does not appear in Q . Now a valid pattern $[A|Q] = r$ is called *positive pattern* by \mathbf{M} , if either of the following two conditions hold:

- 1) $\mathbf{M} \models r - [A] \geq s$, given: Q is an atomic concept; parameter s satisfies $\mathbf{M} \models (0 < s) \wedge (s \leq 1)$.
- 2) $\mathbf{M} \models r - \max_{1 \leq i \leq j} [A | P_i] \geq s$, given: Q is a composite concept; concepts P_1, P_2, \dots, P_j are all the parent concepts of Q ; and parameter s satisfies $\mathbf{M} \models (0 < s) \wedge (s \leq 1)$.

A valid pattern $[A/Q]=r$ is called *negative pattern* by \mathbf{M} , if either of the following two conditions hold:

- 1) $\mathbf{M} \models [A] - r \geq s$, given: Q is an atomic concept; parameter s satisfies $\mathbf{M} \models (0 < s) \wedge (s \leq 1)$.
- 2) $\mathbf{M} \models \min_{1 \leq i \leq j} [A | P_i] - r \geq s$, given: Q is a composite concept; concepts P_1, P_2, \dots, P_j are all the parent concepts of Q ; and parameter s satisfies $\mathbf{M} \models (0 < s) \wedge (s \leq 1)$.

Both positive and negative patterns are *potentially useful patterns*. Parameter s is called *significance controller*.

This definition was given based upon the *direct inference* mechanism of Bacchus's logic, which deals with the inference of propositional probability from statistical assertion in AI. Roughly speaking, *those statistical patterns that affect one's propositional assertion when one faces some particular situation are potentially useful*. (Please refer to [2] for details.)

Furthermore, we derived two interesting lemmas from this formal definition of "potential usefulness". These two lemmas enabled us to design tractable algorithm that is capable of discovering all potentially useful patterns from databases with the same time complexity as association mining. (Please refer to [2] for the details of the DAPUP algorithm.)

Lemma 4.1: If a valid pattern $[Q | P] = r$ is a positive pattern, then $[Q | P] = r$ is an association rule pattern with *confidence* $\geq s$ and *support* $\geq n \cdot s$, where n is the *noise controller* (see definition 3.2) and s is the *significance controller*.

Lemma 4.2: If a valid pattern $[Q | P] = r$ is a negative pattern and P is a composite concept, assuming P_1, P_2, \dots, P_j be all the parent concepts of P , then $[Q | P_1] = r_1, [Q | P_2] = r_2, \dots, [Q | P_j] = r_j$ are all association rule pattern with *confidence* $\geq s$ and *support* $\geq n \cdot s$.

V. NOVEL PATTERNS

Intuitively, if a pattern can be easily deduced from previously known patterns, it should not be classified as previously unknown pattern. The fact that a user's knowledge base can be represented formally using Bacchus' logic formula, allow us to give a formal definition of "previously unknown pattern".

Assume KB_0 is the knowledge base expressed in Bacchus probabilistic logic; and, Pr is an efficient enough deductive program based on the axiom system of Bacchus's probabilistic logic. We now define a previously unknown pattern as follows. For any implicit pattern $[P|Q] = r$, if no formula like $[P|Q] = r_1 \wedge (r - e < r_1) \wedge (r_1 < r + e)$, where r_1 can be any number that belongs to \mathbf{R} , can be deduced from KB_0 by Pr within c steps, then it is called a *previously unknown pattern*, which is denoted as $\text{KB}_0 \not\models_{Pr(c)} [P|Q] = r$. Parameter c is called *complexity controller*, which is utilized to balance the use of deduction and induction processes. Parameter e is

called *error controller*, which is utilized to identify similar or nearly identical knowledge.

The above definition provides a formal way for a KDD system to automatically classify pattern as previously unknown or not. Furthermore, the soundness of this approach is guaranteed by Bacchus's probabilistic logic. However, probabilistic logic views concepts, as, pure symbols without taking into consideration any semantic relation between concepts. Unfortunately, this simplification may sometimes negatively affect the system's judgment. For instance, assuming that $\text{SportWatch} \rightarrow \text{Pillow}$ represents a previously known pattern, and through the KDD process, the user discover the pattern $\text{OutdoorWatch} \rightarrow \text{Pillow}$, the question to ask is whether this discovered pattern is previously unknown? Although SportWatch and OutdoorWatch are logically different antecedents, they are nevertheless semantically close to each other. Therefore, this discovered pattern should not be counted as previously unknown. In order to address this problem, we introduce a specific logic proposition $\text{Sim}(x, y, z)$. Specially, given two concepts P and Q , $\mathbf{M} \models \text{Sim}(P, Q, k)$, if and only if the semantic distance between concepts P and Q is less than k . Now we can revise our previous definition of a previously unknown pattern.

Definition 5.1: Previously Unknown Pattern

Assume KB_0 represents a knowledge base expressed in Bacchus probabilistic logic; and, Pr is an efficient enough deductive program based on the axiom system of Bacchus's probability logic. For any implicit pattern $[P|Q] = r$, if no formula like $[D|C] = r_1 \wedge (r - e < r_1) \wedge (r_1 < r + e) \wedge \text{Sim}(Q, C, k) \wedge \text{Sim}(P, D, k)$ can be deduced from KB_0 by Pr within c steps, then it is called *previously unknown pattern*, which is denoted as $\text{KB}_0 \not\models_{Pr(c)} [P|Q] = r$. The parameter c and e are defined as above. The parameter k is called *similarity controller*, which is utilized to identify semantically similar concepts.

In order to calculate the semantic distance between two concepts, some extra information should be provided. Taxonomy, which groups similar concepts into the same category, is relatively easy to obtain in many situations. In this paper, we propose to calculate the semantic distance between concepts on the basis of a taxonomy tree. Figure 3 shows a taxonomy tree defined in terms of the atomic concepts from table 1. A simple definition of the semantic distance between two atomic concepts is the length of the path between the two concepts. For instance, in figure 3, the semantic distance between *Utensils* and *Couch* is four, while the semantic distance between *Utensils* and *Appliance* is two. If

the concepts P and Q are not atomic, and let $P = \bigwedge_{i=1}^n p_i, Q =$

$\bigwedge_{j=1}^m q_j$, where all p_i and q_j are atomic concepts, the semantic distance between P and Q is defined as $d(P, Q) = \text{avg}(d(p_i, q_j))$, where $d(p_i, q_j)$ denotes the semantic

distance between atomic concepts p_i and q_j .

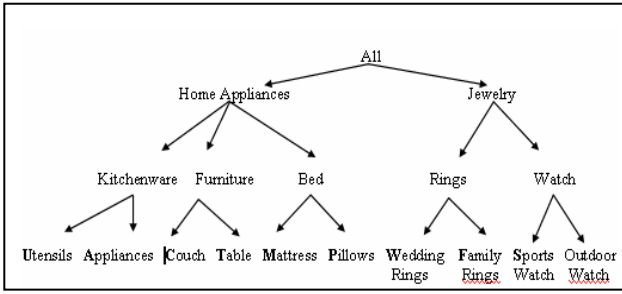


Figure 3: Taxonomy over the Atomic Concepts.

Example 5.1

The semantic distance between composite concepts $Utensils \wedge Couch$ and $Pillows \wedge OutdoorWatch$ is calculated as follows: $d(U \wedge C, P \wedge O) = \text{avg}(d(U, P), d(U, O), d(C, P), d(C, O)) = \text{avg}(4, 6, 4, 6) = 5$.

As mentioned before, two previously unknown patterns may not have the same degree of surprisingness. For example, the positive pattern $WeddingRing \rightarrow Pillows$ may be viewed as a more surprising discovery than the positive pattern $Couch \rightarrow Pillows$ since the semantic distance between $WeddingRing$ and $Pillows$ is greater than the semantic distance between $Couch$ and $Pillows$ (Figure 3). On the contrary, the negative pattern $Mattress \rightarrow \neg Pillows$ may be viewed as a more surprising discovery than the negative pattern $OutdoorWatch \rightarrow \neg Pillows$ since $Pillows$ is semantically closer to $Mattress$ than to $OutdoorWatch$. Hence, the degree of surprisingness can be measured in terms of the semantic distance.

Definition 5.2 Degree of Surprisingness of a Pattern

Given a pattern $[P|Q] = r$, if it is a positive pattern, its degree of surprisingness S is defined as: $S = d(P, Q)$, where $d(P, Q)$ is the semantic distance between concept P and Q ; if it is a negative pattern, its degree of surprisingness S is defined as: $S = 1/d(P, Q)$.

Therefore, novel patterns are previously unknown patterns with a degree of surprisingness greater than a user specified threshold. In the rest of the paper, we use sur_p and sur_n to represent the surprisingness threshold for positive patterns and negative patterns respectively.

Given the formal definition of novelty of a discovered pattern, we extend our **DAPUP** [22] to **DANPUP** (**D**iscovering **A**ll **N**ovel and **P**otentially **U**seful **P**atterns) algorithm. Since the requirement of surprisingness adds more constraints to patterns, DANPUP algorithm can be designed in the manner that is more efficient than DAPUP. Especially, lemma 5.1 can be used to speed up finding surprising negative patterns.

Lemma 5.1: If $d(P, Q) \geq sur$, then there is at least one parent concept of P , denoted as R , such that $d(R, Q) \geq sur$.

Algorithm 4.1 DANPUP (Discovering All Novel and Potentially Useful Patterns)

Some terminology used in the algorithm is explained below,

- $Pa_i = ([Q | P] = r)$ is a pattern, where, $P = \bigwedge_{j=1}^M a_j, a_j \in \{a_1, a_2, \dots, a_M\}$
 $Q = \bigwedge_{k=1}^N b_k, b_k \in \{b_1, b_2, \dots, b_N\}$
antecedent (Pa_i) is P
consequent (Pa_i) is Q .
length (P) = M
length (Q) = N
parents(P) = $\{ \bigwedge_{l=1}^{M-1} c_l : c_l \in \{a_1, a_2, \dots, a_M\} \}$
 if $R \in \text{children}(P)$, then
 $R = \bigwedge_{l=1}^{M+1} c_l, c_l \in \{a_1, a_2, \dots, a_M, b\}$
- FI_n is the set of atomic or composite concepts, such that, for any concept $P \in FI_n \Leftrightarrow [P] \geq n$.
- FI_{ns} is the set of atomic or composite concepts, c that, for any concept $P \in FI_{ns} \Leftrightarrow [P] \geq n \times s$.
- NPR is the set of association rule patterns, such that for any $Pa_i \in NPR$
 $sur_p \geq d(\text{antecedent}(Pa_i), \text{consequent}(Pa_i)) \geq sur_n$
- NPR_{Ql} is the subset of NPR such that for any $Pa_i \in NPR$, if Pa_i satisfies
consequent (Pa_i) = Q , and
length (*antecedent* (Pa_i))) = l , then $Pa_i \in NPR_{Ql}$
- AP is the set of all mined association rule patterns.
- AP_{Ql} is a subset of AP such that, for any $Pa_i \in AP$, if Pa_i satisfies
consequent (Pa_i) = Q , and
length (*antecedent* (Pa_i))) = l , then $Pa_i \in AP_{Ql}$
- NPP is the set of novel positive patterns.
- NNP is the set of novel negative patterns.

Input: significance controller s , noise controller n , surprisingness threshold for positive pattern sur_p and surprisingness threshold for negative pattern sur_n

Output: The set of novel positive patterns NPP and the set of novel negative patterns NNP

- [1] Generate all frequent itemset with support $\geq n \cdot s$ and put them into FI_{ns} . For each atomic or composite concept P in FI_{ns} , if $[P] \geq n$, copy it to set FI_n .
- [2] **for** each concept P in FI_n , if $P \wedge Q$ is in FI_{ns}
- [3] **if** $[Q|P] = r \geq s$ /* $[Q|P] = r$ is association rule with confidence no less than s */
 /* the following finds surprising, positive patterns*/
- [4] **if** $d(P, Q) \geq sur_p$ /*definition of surprising pattern*/
- [5] **if** $(r - \max_{1 \leq i \leq j} [Q | P_i]) \geq s$ for all $P_i \in \text{parents}(P)$ //definition of positive pattern
 add $[Q|P] = r$ to NPP .
- [6] **else if** $d(P, Q) \geq sur_n$
 add $[Q|P] = r$ to NPR /* NPR is used to speed up finding surprising negative patterns by lemma 5.1*/
- [7]
- [8]

[9] add $[Q|P] = r$ to **AP** /*AP stores all association rules that will be used as seeds to generate surprising negative rules*/

//the following finds surprising, negative pattern

[10] Partition **NPR** into multiple subsets **NPR_{XY}**

[11] **for** each subset **NPR_{QI}** /* by lemma 5.1, all these subsets are use to speed up finding surprising negative patterns*/

[12] **if** **APR_{QI}** has size 2^l

[13] **if** there exists a concept **R**, such that for every $Pa_i \in \mathbf{APR}_{QI}$ we have $R \in \text{children}(\text{antecedent}(Pa_i))$

[14] **if** $R \in \mathbf{FI}_n$ /*the above three if conditions make sure the candidate negative pattern exists*/

[15] **if** $d(R, Q) \geq \text{sur}_n$ /*definition of surprising pattern*/

[16] let $[Q/R] = r$

[17] **if** $\min_{1 \leq i \leq j} ([Q | R_i] - r) \geq s$ for

all $R_i \in \text{parents}(R)$
/*definition of negative pattern*/

[18] add $[Q/R] = r$ to **NNP**

//the following conducts previously unknown evaluation

[19] **for** each pattern Pa_i in **NPP** and **NNP**

[20] **if** **NOT** $\text{KB}_0 \not\subseteq_{Pr(c)} Pa_i$. /*definition of previously unknown pattern*/

[21] delete Pa_i

/*finally output both novel positive patterns and novel negative patterns*/

[22] Output **NPP** and **NNP**

VI. DISCOVERING GENERALIZED POTENTIALLY USEFUL PATTERNS

It is possible to discover novel and potentially useful patterns that span different levels of a taxonomy. We can use logic disjunction to represent any category introduced by the taxonomy. For example, in the taxonomy shown in Figure 3, the category *Furniture* can be thought of as a disjunction of the concepts *Couch* and *Table* ($Couch \vee Table$). This representation enables us to integrate taxonomy and the concept lattice into a single structure. Figure 4 illustrates the integration of concept lattice shown in Figure 2 with the taxonomy shown in Figure 3.

Given such an integrated concept lattice with taxonomy, we can discover *generalized* novel and potentially useful rules that span different levels of hierarchy. For example we may obtain a positive pattern like *KitchenWare* \rightarrow *Bed*.

To discover these types of patterns, we simply replace the first step of the DANPUP algorithm with an algorithm for generating frequent itemsets for generalized association rules, as proposed in [20]. To evaluate the degree of surprisingness of a generalized pattern, we need to define the semantic distance between different categories (disjunction

of concepts), or between a category and an atomic concept. One approach is to directly extend the definition of semantic distance to categories. For example, the semantic distance between the categories *Kitchenware* and *Furniture* is two, however, the semantic distance between the concepts *Utensils* and *Appliances* is also two. We content that two concepts *Utensils* and *Appliances* are more closely related to each other than the two categories *Kitchenware* and *Furniture*. To account for this situation, we label the edges from each node with a value calculated using the formula $1/D$, where D is the depth of the node in the taxonomy (depth of the root = 0). This formula ensures that siblings with greater depth are evaluated as being more closely related than siblings at a lesser depth. The semantic distance between two concepts (categories) can now be defined as the weighted sum of values along the path between the two nodes.

VII. CONCLUSION AND FUTURE WORKS

By extending our former work on discovering potentially useful patterns, in this paper, we proposed a formal approach to evaluate the novelty of a discovered pattern. The proposed novelty measure evaluates patterns at two levels. First, a novel pattern should be previously unknown; second, a novel pattern requires certain degree of surprisingness. All these measures, i.e., potential usefulness, previous unknownness and surprisingness are completely staying within the realms of the expressiveness provided by Bacchus' Probabilistic Logic Language. Furthermore, a tractable algorithm called DANPUP is proposed that is capable of discovering all novel and potentially useful patterns from databases.

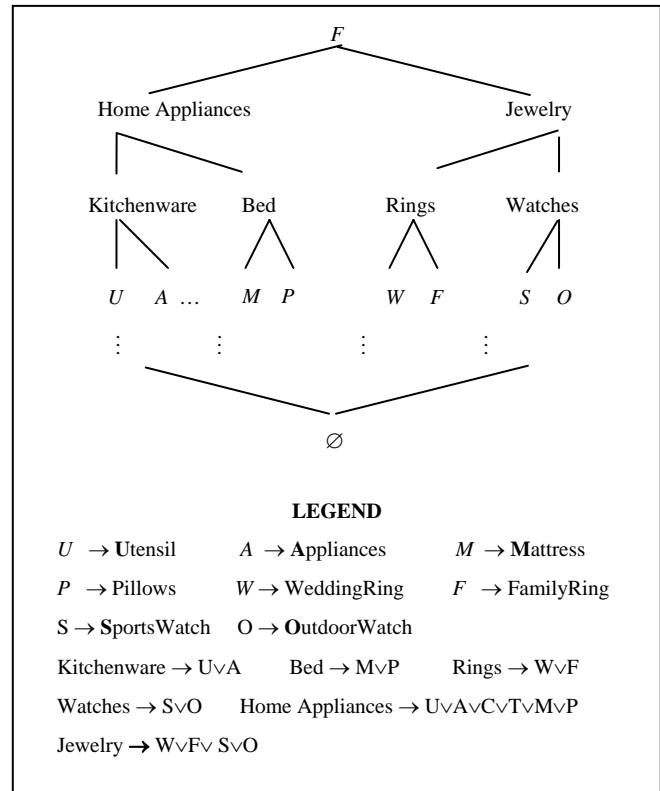


Figure 4: Integrated Concept Lattice with Taxonomy over Atomic concepts

We recognize that a knowledge base of previously known patterns may not always be available. Therefore, as part of our future research, we plan to explore the ways in which a KDD system can conduct previously unknown judgments based on auxiliary knowledge sources. One of our strategies is to utilize the dimensionality of the recorded data. For instance, the transaction table records *sales* for products. There should be other data sets recording *price*, *shelf-location*, *order*, and *promotion* information for products as well. The profitable actions guided by the useful patterns mined from sales dataset will affect price, shelf-location or other datasets with different dimensions. Therefore, it is possible to obtain action patterns by mining data from different dimensions, so as to gain knowledge about the fact that a discovered pattern from sales dataset has already been known or not.

REFERENCE

- [1] Bacchus F, Representing and Reasoning With Probabilistic Knowledge, MIT-Press, Cambridge, MA., 1990.
- [2] Bacchus F, Lp, A Logic for Representing and Reasoning with Statistical Knowledge, Computational Intelligence 6:209-231, 1990.
- [3] Frawley W, Piatetsky-Shapiro G and Matheus C, Knowledge Discovery In Databases: An Overview. Piatetsky-Shapiro G, Frawley W (eds) Knowledge Discovery In Databases, AAAI Press/MIT Press, Cambridge, MA., 1991.
- [4] Boose, J., A survey of knowledge acquisition techniques and tools, In B. Buchanan & D. Wilkins (ed.) Knowledge Acquisition and Learning, 1993, pp. 39-56.
- [5] Padmanabhan, B. and Tuzhilin, A., Unexpectedness as a Measure of Interestingness in Knowledge Discovery, Decision Support Systems, Vol.27 (3), 1999.
- [6] Liu, B. and Hsu, W., 1996, Post-Analysis of Learned Rules, Proc. of the Thirteenth National Conference on Artificial Intelligence (AAAI '96), 1996, pp. 828-834.
- [7] Silberschatz, A., and Tuzhilin, A. What makes patterns interesting in knowledge discovery systems, IEEE Trans. on Knowledge and Data Engineering 8(6), 1996, pp. 970-974.
- [8] Bing Liu, Wynne Hsu, Shu Chen, and Yiming Ma. Analyzing the subjective interestingness of associationrules. IEEE Intelligent Systems, 15(5):47-55, 2000.
- [9] Klemetinen, M., Mannila, H., Ronkainen, P., Toivonen, H. and Verkamo, A.I. 1994, Finding interesting rules from large sets of discovered association rules, Proceedings of the Third International Conference on Information and Knowledge Management, 1994, pp. 401-407.
- [10] Pawlak Z, Rough Sets, Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, 1991.
- [11] Yao Y, On Modeling Data Mining with Granular Computing, Proceeding of the 25th Annual International Computer Software and Applications Conference, Chicago, IL., 2001.
- [12] Lin T, Louie E, Modeling the Real World for Data Mining: Granular Computer Approach, Proceeding of IFSA/NAFIPS, Vancouver, Canada, 2001.
- [13] Louie E, Lin T, Semantics Oriented Association Rules, Proceeding of FUZZ-IEEE Conference IEEE World Congress on Computational Intelligence, Honolulu, HI., 2002.
- [14] Murai T, Murai M, Sato Y , A Note on Conditional Logic and Association Rules, Proceeding of JSAI International Workshop on Rough Set Theory and Granular Computing, Matsue, Japan, 2001.
- [15] Agrawal R, Imielinski T, Swami A, Mining association rules between sets of items in large databases, Proceeding of ACM-SIGMOD International Conference on Management of Data, Washington, DC., 1993.
- [16] Savasere A, Savasere E, Navathe S, Mining for Strong Negative Associations in a Large Database of Customer Transactions, Proceeding of the 14th International Conference on Data Engineering, Orlando, Florida, 1998.
- [17] Suzuki E, tonomous Discovery of Reliable Exception Rules. In: Proceeding of the 3th International Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, California, 1997.
- [18] Zhong N, Yao Y and Ohsuga S, Peculiarity Oriented Multi-database Mining, Proceeding of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases, Freiburg, Germany, 1999.
- [19] Rakesh A., Tomasz I. and Arun S., Mining association rules between sets of items in large databases, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C. (1993) 207-216.
- [20] Ramakrishnan S., Rakesh A., Mining generalized association rules. Future Generation Computer Systems 13(1997) 161-180.
- [21] Y. Xie and V. V. Raghavan, A Probabilistic Logic-based Framework for Characterizing knowledge Discovery in Databases, *Foundations of Data Mining and Knowledge Discovery* (Editors: T. Y. Lin, S. Ohsuga, C. J. Liau, S. Tsumoto), Springer-Verlag Berlin Heidelberg, pp. 87-100, ISBN 3-540-26257-1, 2005.
- [22] Ying Xie, M. Nagarajan, K. Ramachandran, Tom Johnsten, Vijay V. Raghavan, On Discovering "Potentially Useful" Patterns from Databases, 2006 IEEE International Conference on Granular Computing, May 2006, to appear.