A Feature/Attribute Theory for Association Mining and Constructing the Complete Feature Set

Tsau Young Lin

Department of Computer Science
San Jose State University, San Jose, California 95192-0103
tylin@cs.sjsu.edu

Summary. A correct selection of features (attributes) is vital in data mining. For this aim, the complete set of features is constructed. Here are some important results: (1) Isomorphic relational tables have isomorphic patterns. Such an isomorphism classifies relational tables into isomorphic classes. (2) A unique canonical model for each isomorphic class is constructed; the canonical model is the bitmap indexes or its variants. (3) All possible features (attributes) is generated in the canonical model. (4) Through isomorphism theorem, all un-interpreted features of any table can be obtained.

Keywords: attributes, feature, data mining, granular, data model

1 Introduction

Traditional data mining algorithms search for patterns only in the given set of attributes. Unfortunately, in a typical database environment, the attributes are selected primarily for record-keeping, not for understanding of real world. Hence, it is highly possible that there are no visible patterns in the given set of attributes; see Sect. 2.2. The fundamental question is: Is there a suitable transformation of features/attributes so that

- The "invisible" patterns become visible in this new set?

Fortunately, the answer is yes. To answer this question, we critically analyze the essence of association mining. Based on it, we are able

- To construct the *complete* set of features for a given relational table.

Many applications will be in the forth coming volumes [10]. Here are some important results:(Continue the count from the abstract) (5) all high frequency

T.Y. Lin: *A Feature/Attribute Theory for Association Mining and Constructing the Complete Feature Set*, StudFuzz **180**, 63–78 (2005)
www.springerlink.com

patterns (generalized association rules) of the canonical model can be generated by a finite set of linear inequalities within polynomial time. (6) Through isomorphism theorem, all high frequency patterns of any relational table can be obtained.

1.1 Basics Terms in Association Mining (AM)

First, we recall (in fact, formalize) some basic terms. In traditional association rule mining, two measures, called the support and confidence, are the main criteria. Among the two, support is the essential measure. In this paper, we will consider **the support only**. In other words, we will be interested in the high frequency patterns that are not necessary in the form of rules. They could be viewed as *undirected association rules*, or just *associations*.

Association mining is originated from the market basket data [1]. However, in many software systems, the data mining tools are added to general DBMS. So we will be interested in data mining on *relational tables*. To be definitive, we have the following translations:

1. a relational table is a bag relation (i.e., repeated tuples are permitted [8])
2. an item is an attribute value,
3. a q-itemset is a subtuple of length q, or simply q-subtuple,
4. A q-subtuple is a q-association or (high frequency) q-pattern, if its occurrences are greater than or equal to a given threshold.

2 Background and Scope

2.1 Scope – A Feature Theory Based on the Finite Data

A feature is also called an attribute; the two terms have been used interchangeably. In the classical data model, an attribute is a representation of property, characteristic, and so forth [17]. It represents a human view of the universe (a slice of real world) – an intension view [5]. On the other hand, in modern data mining (DM), we are extracting information from the data. So in principle, the real world, including features (attributes), is encoded by and only by a finite set of data. This is an extension view or data view of the universe.

However, we should caution that each techniques of data mining often use some information (background knowledge) other than data [6]. So the encoding of the universe is different for different techniques. For examples association mining (AM) (Sect. 3) uses only the relational table, while clustering techniques utilize not only the table (of points), but also the geometry of the ambient space. So the respective feature theories will be different. In this paper, we will focus on Association Mining.

Next, we will show some peculiar phenomena of the finite encoding. Let Table 1θ and 2θ be the new tables derived from the tables in Sect. 2.2 by

Table 1. Ten point in (X,Y)-coordinate and Rotated coordinate

Segment#		Y	X'	Y'
S_1	2	0	1.99	0.17
S_2	$\sqrt{3} = 1.73$	1	1.64	1.15
S_3	$\sqrt{2} = 1.41$	$\sqrt{2}$	1.29	1.53
S_4	1	$\sqrt{3}$	0.85	1.81
S_5	0	2	-0.17	1.99
	X-Y coordinate Table 1A		Rotates -5 degree Table 1B	

rotating the coordinate systems θ degree. It should be easy to verify that (see Sect. 4.1 for the notion of isomorphism).

Proposition 2.1.1. *Table 1A, 1B and 1θ are isomorphic, so are the Table 2A, 2B and 2θ.*

This proposition says even though the rotations of the coordinate system generate infinitely many distinct features/attributes, they reduce to the same feature/attribute if the universe is encoded by a relational table. The main result of this paper is to determine all possible features of the encoded world.

2.2 Background – Mining Invisible Patterns

Let us consider a table of 5 points in X-Y-plane, as shown in Table 1A. The first column is the universe of the geometric objects. It has two attributes, which are the "X-Y coordinates." This table has no association rule of length 2. By transforming, the "X-Y coordinates" to "Polar coordinate system" (Table 2A), interestingly

Associations of length 2 appear .

The key question is how can we find such appropriate new features (polar coordinates).

Table 2. Ten points in polar coordinate and rotated coordinate

Segment#	Length	Direction	Length	Direction
S_1	2.0	0	2.0	5
S_2	2.0	30	2.0	35
S_3	2.0	45	2.0	45
S_4	2.0	60	2.0	65
S_5	2.0	90	2.0	95
	X-Y coordinate Table 2A		Rotates -5 degree Table 2B	

3 Formalizing Association Mining

In this section, we will critically analyze the association (rule) mining. Let us start with a general question: What is data mining? There is no universally accepted formal definition of data mining, however the following informal description (paraphrase from [7]) is rather universal:

- Deriving useful patterns from data.

This "definition" points out key ingredients: data, patterns, methodology of derivations and the real world meaning of patterns (useful-ness). We will analyze each of them.

3.1 Key Terms "Word" and "Symbol"

First we need to precisely define some key terms.

A *symbol* is a string of "bit and bytes." It has no *formal* real world meaning, more precisely, any real world interpretation (if there is one) does *not* participate in formal processing or computing. Mathematicians (in group theory, more specifically) use the term "word" for such purpose. However, in this paper, a "word" will be more than a symbol. A symbol is termed a *word*, if the intended real world meaning *does participate* in the formal processing or computing. In AI, there is a similar term, semantic primitive [2]; it is a symbol whose real world interpretation is not implemented. So in automated computing, a semantic primitive is a symbol.

3.2 What are Data? – A Table of Symbols

To understand the nature of the data, we will examine how the data is created: In traditional data processing, (1) we select a set of attributes, called relational schema. Then (2) we (knowledge) represent a set of real world entities by a table of words.

$$K_{map} : V \ \rightarrow \ K_{word} \ ; \quad v \ \longrightarrow \ k$$

where K_{word} is a table of words (this is actually the usual relational table). Each word, called an attribute value, represents a real world fact (to human); however the real world semantic is not implemented. Since K_{word} is a bag relation [8], it is more convenient to use the graph $K_{graph} = \{(v, K(v) \mid v \in V\}$. If the context is clear, we may drop the subscript, so K is a map, an image or a graph.

Next, how is the data processed? In traditional data processing environment, for example, the attribute name COLOR means exactly what a human thinks. Therefore its possible values are yellow, blue, and etc. More importantly,

- DBMS processes these data under *human commands*, and carries out the human perceived-semantics. Such processing is called Computing with Words.

However, in the system, COLOR, yellow, blue, and etc are "bits and bytes" without any meaning, they are pure symbols.

The same relational table is used by Association Mining (AM). But, the data are processed without human interventions, so the table of words K_{word} is processed as a table K_{symbol} of symbols.

$$DM: \; K_{word} \quad \Rightarrow \quad K_{symbol}$$

In summary,

- *The data (relational table) in AM is a table of symbols.*

3.3 What are Patterns? and Computing with Symbols

What are the possible patterns? The notion depends on the methodology. So we will examine the algorithms first. A typical AM algorithm treats words as symbols. It just *counts* and does not consult human for any possible real world meaning of any symbol. As we have observed in previous section no real world meaning of any symbols is stored in the system. So an AM algorithm is merely a computing of pure symbols. AM transforms a table K_{symbol} of symbols into a set A_{symbol} of association (rules)s of symbols. These associations are "expressions" of symbols. Therefore,

- All possible patterns of AM are expressions of the symbols of the relational table.

3.4 Interpretation and Realization of Patterns

The output of an AM algorithm is examined by human. So each symbol is alive again. Its interpretation (to human only) is assigned at the data creation time. So the patterns are interpreted by these interpretations of symbols.

1. Interpretation: A pattern, an expression of symbols, is an expression of words (to human). So a pattern is a mathematical expression of real world facts.
2. Realization: A mathematical expression of real world facts may or may not correspond to a real world phenomenon.

4 Understanding the Data – A Table of Symbols

In the previous section, we have concluded that the input data to AM is a table of symbols. In this section, we will explore the nature of such a table.

4.1 Isomorphism – Syntactic Nature of AM

We have explained how data is processed in (automated) data mining: The algorithms "forget" the real world meaning of each word, and regard the input data as pure symbols. Since no real world meaning of each symbol participates in the computing process if we replace the given set of symbols by a new set, then we can derive new patterns by simply replacing the symbols in "old" patterns. Formally, we have (Theorem 4.1. of [12])

Theorem 4.1.1. *Isomorphic relational tables have isomorphic patterns.*

Though this is a very important theorem, its proof does not increase the understanding. Its proof is in the appendix. Isomorphism is an equivalence relation defined on the family of all relational tables, so it classifies the tables into isomorphic classes.

Corollary 4.1.2. *A pattern is a property of an isomorphic class.*

The impacts of this simple theorem are rather far reaching. It essentially declares that patterns are syntactic in nature. They are patterns of the whole isomorphic class, even though many somorphic relations may have very different semantics.

Corollary 4.1.3. *The probability theory based on the item counting is a property of isomorphic class.*

We will illustrate the idea by an example. The following example is adopted from ([8], pp 702):

Example 4.1.4. In this example, we will illustrate the notion of isomorphism of tables and patterns. In Table 3, we present two "copies" of relational tables; they are obviously isomorphic (by adding prime' to one table you will get the other one). For patterns (support = 2), we have the following:

Isomorphic tables K and K' have isomorphic q-associations:

1. 1-association in K: 30, 40, bar, baz,
2. 1-association in K′: 30′, 40′, bar', baz',
3. 2-association in K: (30, bar) and (40, baz),
4. 2-association in K': (30′, bar') and (40′, baz').

Two sets of q-association (q = 1,2) are obviously isomorphic in the sense that adding prime ′ to associations in K become associations in K'.

Table 3. A Relational Table K and its Isomorphic Copy K′

V	$\to$	F	G	V	$\to$	F'	G'
e_1	$\to$	30	foo	e_1	$\to$	$30'$	foo'
e_2	$\to$	30	bar	e_2	$\to$	$30'$	bar'
e_3	$\to$	40	baz	e_3	$\to$	$40'$	baz'
e_4	$\to$	50	foo	e_4	$\to$	$50'$	foo'
e_5	$\to$	40	bar	e_5	$\to$	$40'$	bar'
e_6	$\to$	40	bar	e_6	$\to$	$40'$	bar'
e_7	$\to$	30	bar	e_7	$\to$	$30'$	bar'
e_8	$\to$	40	baz	e_8	$\to$	$40'$	baz'

4.2 Bitmaps and Granules – Intrinsic Representations

Due to the syntactic nature, as we have observed in last section, we can have a more intrinsic representation, that is a representation in which only the internal structure of the table is important, the real world meaning of each attribute value can be ignored.

We will continue to use the same example. The following discussions essential excerpt from ([8], pp 702). Let us consider the bitmap indexes for K (see Table 3) the first attributes, F, would have three bit-vectors. The first, for value 30, is 11000110, because the first, second, sixth, and seventh tuple have $F = 30$. The other two, for 40 and 50, respectively, are 00101001 and 00010000. A bitmap index for G would also have three bit-vectors: 10010000, 01001010, and 00100101. It should be obvious that we will have the exact same bitmap table for K'.

Next, we note that a bit vector can be interpreted as a subset of V, called an elementary granule. For example, the bit vector, 11000110, of $F = 30$ represents the subset $\{e_1, e_2, e_6, e_7\}$. Similarly, 00101001, of $F = 40$ represents the subset $\{e_3, e_5, e_8\}$, and etc. Let us summarize the discussions in the following proposition:

Proposition 4.2.1. *Using Table 4 as a translation table, we transform a table of symbols (Table 3) into its respective*

1. a bitmap table, and Table 5.
2. a granular table, Table 6.

Conversely,

Proposition 4.2.2. *Using Table 4 as an interpretation table that interpret*

1. Table 5 and Table 6 into Table 3, where (to human) each symbol corresponds to a real world fact.
2. Note that F-granules (and G-granules too) are mutually disjoints and form a covering of V. So the granules of each attribute induces a partition on V (an equivalence relation).

Table 4. Translation Table

F-Value	Bit-Vectors	Granules
30	$= 11000110$	$(\{e1, e2, e6, e7\})$
40	$= 00101001$	$(\{e3, e5, e8\})$
50	$= 00010000$	$(\{e4\})$
G-Value	$=$ Bit-Vectors	Granules
Foo	$= 10010000$	$(\{e1, e4\})$
Bar	$= 01001010$	$(\{e2, e5, e7\})$
Baz	$= 00100101$	$(\{e3, e6, e8\})$

Table 5. Contrasting Tables of Symbols and Bitmaps

Table K					Bitmap Table B_K	
V	$\longrightarrow$	F	G		F-bit	G-bit
e_1	$\longrightarrow$	30	foo		11000110	10010000
e_2	$\longrightarrow$	30	bar		11000110	01001010
e_3	$\longrightarrow$	40	baz		00101001	00100101
e_4	$\longrightarrow$	50	foo		00010000	10010000
e_5	$\longrightarrow$	40	bar		00101001	01001010
e_6	$\longrightarrow$	30	baz		11000110	00100101
e_7	$\longrightarrow$	30	bar		11000110	01001010
e_8	$\longrightarrow$	40	baz		00101001	00100101

Table 6. Contrasting Tables of Symbols and Granules

Table K					Granular Table G_K	
U	$\longrightarrow$	F	G		E_F	E_G
$v1$	$\longrightarrow$	30	foo		$\{e1, e2, e6, e7\}$	$\{e1, e4\}$
$v2$	$\longrightarrow$	30	bar		$\{e1, e2, e6, e7\}$	$\{e2, e5, e7\}$
$v3$	$\longrightarrow$	40	baz		$\{e3, e5, e8\}$	$\{e3, e6, e8\}$
$v4$	$\longrightarrow$	50	foo		$\{e4\}$	$\{e1, e4\}$
$v5$	$\longrightarrow$	40	bar		$\{e3, e5, e8\}$	$\{e2, e5, e7\}$
$v6$	$\longrightarrow$	30	baz		$\{e1, e2, e6, e7\}$	$\{e3, e6, e8\}$
e_7	$\longrightarrow$	30	bar		$\{e1, e2, e6, e7\}$	$\{e2, e5, e7\}$
e_8	$\longrightarrow$	40	baz		$\{e3, e5, e8\}$	$\{e3, e6, e8\}$

3. Each elementary granule, for example, the elementary granule $\{e_1, e_2, e_6, e_7\}$ of $F = 30$, consists of all entities that have (are mapped to) the same attribute value, in this case, F-value 30. In other words, F-granule $\{e_1, e_2, e_6, e_7\}$ is the inverse of the value $F = 30$.

It should be obvious that these discussions can be generalized: They are summarize in *Proposition* 5.1.1.

5 The Model and Language of High Frequency Patterns

As we have observed in Sect. 3.3, informally patterns are expressions (subtuples) of the symbols of the relational table. Traditional association mining considers only the "conjunction of symbols." Are there other possible expressions or formulas? A big Yes, if we look at a relational table as a logic system. There are many such logic views, for example, deductive database systems, Datalog [21], and Decision Logic [19] among others. For our purpose, such views are too "heavy", instead, we will take an algebraic approach. The idea is stated in [13] informally. There, the notion of "logic language" was introduced informally by considering the "logical formulas" of the names of elementary granules. Each "logical formula" (of names) corresponds to a set theoretical formula of elementary granules. In this section, we shall re-visit the idea more formally.

5.1 Granular Data Model (GDM) – Extending the Expressive Power

Based on example, we have discussed granular data model in Sect. 4.2. Now we will discuss the general case.

Let V be set of real world entities, $A = \{A^1, A^2, \ldots, A^n\}$ be a set of attributes. Let their (active) attribute domains be $C = \{C^1, C^2, \ldots, C^n\}$, where active is a database term to emphasize the fact that C^j is the set of distinct values that occur in the current representation. Each C^j, often denoted by $Dom(A^j)$, is a Cantor set.

A relational table K can be regarded as a map (knowledge representation)

$$K_{map} : V \longrightarrow Dom(A) = Dom(A^1) \times \ldots Dom(A^n)$$

Similarly, an attribute is also a map

$$A^j : V \longrightarrow Dom(A^j)\ ; \quad v \longrightarrow c\ .$$

The inverse of such an attribute map defines a partition on V (hence an equivalence relation); we will denote it by Q^j and list some of its properties in:

Proposition 5.1.1

1. The inverse image $S = (A^j)^{(-1)}(c)$ is an equivalence class of Q^j. We say S is elementary granule, and c is the name of it.
2. For a fixed order of V, S can be represented by a bit-vector. We also say c is the name of the bit vector.
3. By replacing each attribute value of the table K_{symbol} by its bit-vector or elementary granule (equivalence class), we have the bitmap table B_K or granular table G_K respectively.
4. The equivalence relations, $Q = \{Q^1, Q^2, \ldots, Q^n\}$, play the role of attributes in Table G_K and B_K.

5. For uniformity, we write $V/Q^j = Dom(Q^j)$, namely, we regard the quotient set as the attribute domain.
6. Theoretically, G_K and B_K conceptually represent the same granular data model; the difference is only in representations and is an internal matter.
7. We will regard the table K as an interpretation of G_K and B_K. The interpretation is an isomorphism (via a table similar to Table 4) By Theorem 4.1.1., the patterns in K, G_K, B_K are isomorphic and hence is the same (identified via interpretation).

- It is adequate to do the AM in G_K.

The canonical model G_K is uniquely determined by its universe V, and the family Q of equivalence relations. In other words, the pair (V, Q) determines and is determined by G_K.

Definition 5.1.1. *The pair (V, Q) is called granular data model (GDM).*

(V, Q) is a model of some rather simple kind of logic, where the only predicates are equivalence predicates (predicates that satisfy the reflexive, symmetric and transitive properties). It was considered by both Pawlak and Tony Lee and has been called knowledge base, relation lattice, granular strucutre [9, 13, 19].

Note that the set of all elementary granule in (V, Q) generate a sub-Boolean algebra of the power set of V. By abuse of notation, we will use (V, Q) to denote this algebra. Since G_K is a table format of (V, Q), we need to describe how G_K is "embedded" into the Boolean algebra. We will extend *Proposition* 5.1.1, Item 7 into

Proposition 5.1.2. *An attribute value of G_K, which is an elementary granule, is mapped to the same granule in (V, Q). A subtuple of G_K, consisting of a set of elementary granules is mapped into the granule that is the intersection of those elementary granules; note two subtuples may be mapped to the same granule.*

5.2 Algebraic Language and Granular Boolean Algebra

The attribute values in K are pure symbols. Now we will introduce a new Boolean algebra L_K as follows: We will use $\cup$ and $\cap$ as the join and meet of this Boolean algebra. L_K is a free Boolean algebra subject to the following conditions:

The $\cap$ between symbols in the same columns are

$$B_i^j \cap B_k^j = \quad \forall i \neq k \ \forall j$$

This condition reflects the fact that the elementary granules of the same column are mutually disjoint.

We can give a more algebraic description [3]. Let F be the free Boolean Algebra generated by the symbols in K. Let I be the ideal generated by

$$B_i^j \cap B_k^j \ \ \forall\ i, k, j$$

Then the quotient algebra $F/I \ = \ L_K$.

We will regard this Boolean algebra as a language and call it

$$\text{Granular algebraic language}\ .$$

An attribute value in K can be regarded as the name of the corresponding elementary granule in G_K and the elementary granule is the meaning set of the name. Recall that GDM (V, Q) can be regarded as Boolean algebra of elementary granules, and G_K is "embedded" in (V, Q) (*Proposition* 5.1.2.) So the name-to-meaning set assignment, $K \rightarrow G_K$, can be extended to a homomorphism of Boolean algebras:

$$\text{name-to-meaning: } L_K \longrightarrow (V, Q); \quad \text{formula} \ \rightarrow \ \text{meaning set}\ .$$

- High frequency patterns of AM are formulas with large meaning set (the cardinality is large).

6 The Formal Theory of Features in AM

The theory developed here is heavily depended on the nature of association mining (AM) that are formalized in Sect. 3.

6.1 Feature Extractions and Constructions

Let us examine some informal assertions, e.g., [18]: "All new constructed features are defined in terms of original features,. . . ." and "Feature extraction is a process that extracts a set of new features from the original features through some functional mapping." In summary the new feature is derived (by construction or extraction) from the given set of attributes. We will formalize the idea of features in association mining (AM). Perhaps, we should re-iterate that we are *not* formalizing the general notion of features that involves human view.

Let K be the given relational table that has attributes $A = \{A^1, \ldots A^n\}$. Next, let $A^{n+1} \ldots A^{n+m}$ be the *new* attributes that are constructed or extracted. As we remark in Sect. 5.1, an attribute is a mapping from the universe to a domain, so we have the following new mappings.

$$A^{n+k} : V \longrightarrow Dom(A^{n+k})\ .$$

Now, let us consider the extended table, K_a, that includes both old and additional new attributes $\{A^1, \ldots A^n \ldots A^{n+m}\}$. In this extended table, by the meaning of feature construction, A^{n+k}, should be (extension) functionally dependent (EFD) on A. This fact implies, by definition of EFD, there is a mapping

$$f^{n+k} : Dom(A^1) \times \ldots \times Dom(A^n) \longrightarrow Dom(A^{n+k}) .$$

such that $A^{n+k} = f^{n+k} \circ (A^1 \times \ldots \times A^n$.

Those new extracted or constructed features, such as f^{n+k} is called derived feature.

6.2 Derived Features in GDM

Now we will consider the situation in G_K, the granular table of K. In this section, we will express EFD f^{n+k} in granular format, in other words, the granular form of f^{n+k} is:

$$V/(Q^1 \cap \ldots \cap Q^n) = V/Q^1 \times \ldots \times V/Q^n \longrightarrow V/Q^{n+k}$$

The first equality is a simple property of quotient sets. The second map is f^{n+k} in its granular form. The granular form of f^{n+k} implies that Q^{n+k} is a coarsening of $(Q^1 \cap \ldots \cap Q^k)$. So we have the following

Proposition 6.2. *Q^{n+k} is a derived feature of G_K if and only if Q^{n+k} is a coarsening of $(Q^1 \cap \ldots \cap Q^n)$.*

Let the original Table K have attributes $A = \{A^1, \ldots A^n\}$. Let $B \subseteq A$ and $Y \in A$ (e.g., $Y = A^{n+k}$ and $Y_E = Q^{n+k}$).

Proposition 6.3. *Y is a feature constructed from B if and only if the induced equivalence relation Y_E is a coarsening of the induced equivalence relation $B_E = (Q^{j_1} \cap \ldots \cap Q^{j_m})$, where $Y \in A$ and $B \subseteq A$*

The proposition says all the new constructed features are coarsening of the intersection of the original features.

7 Universal Model – Capture the Invisibles

Let $\Delta(V)$ be the set of all partitions on V (equivalence relations); $\Delta(V)$ forms a lattice, where meet is the intersection of equivalence relations and join is the "union," where the "union," denoted by $\cup_j Q^j$, is the smallest coarsening of all $Q^j, j = 1, 2, \ldots$ $\Delta(V)$ is called the partition lattice.

Let $(V, Q = \{Q^1, \ldots, Q^n\})$ be a GDM. Let $L(Q)$ be the smallest sublattice of $\Delta(V)$ that contains Q, and $L^*(Q)$ be the set of all possible coarsenings of $(Q^1 \cap \ldots \cap Q^n)$. $L^*(Q)$ obviously forms a sublattice of $\Delta(V)$; the intersection and "union" of two coarsenings is a coarsening. From *Proposition* 6.2., we can easily establish

Theorem 7.1. *Let G_K be a granular table; its GDM is (V, Q). Then $(V, L^*(Q))$ is a GDM that consists of all possible features for G_K.*

The set of all possible features of G_K is the set D of all those derived features. By *Proposition* 6.2., D is the set of all those coarsenings of $(Q^1 \cap \ldots \cap Q^n)$. SO $(V, L^*(Q))$ is the desirable one.

Definition 7.2. *The* $(V, L^*(Q))$ *is the completion of* (V, Q) *and is called the universal model of* K*.*

We should point out that the cardinal number of $L^*(Q))$ is enormous; it is bounded by the Bell number B_n, where n is the cardinality of the smallest partition in $L^*(Q)$ [4].

8 Conclusions

1. A feature/attribute, from human view, is a characteristic or property of the universe (a set of entities). Traditional data processing takes such a view and use them to represent the universe (knowledge representation).
2. A feature/attribute, in data mining, is defined and encoded by data. So a feature in association mining is a partition of the universe. Under such a view, we have shown that a set of infinite many distinct human-view-features (rotations of coordinate systems) is reduced to a single data-encoded-feature (Sect. 2.1).
3. Such views are shared by those techniques, such as classification, that utilize only the relational table of symbols in their algorithms. The other techniques, such as clustering and neural network, that utilize additional background knowledge, do not share the same view.
4. In association mining, we have the following applications [10, 11]: All generalized associations can be generated by a finite set of integral linear inequalities within polynomial time.
5. Finally, we would like to note that by the isomorphism theorem, two isomorphic relations may have totally distinct semantics. So relations with additional structures that capture some semantics may be worthwhile to be explored; see [13, 15].

9 Appendix

9.1 General Isomorphism

Attributes A^i and A^j are isomorphic if and only if there is a one-to-one and onto map, $s : Dom(A^i) \longrightarrow Dom(A^j)$ such that $A^j(v) = s(A^i(v)) \; \forall \; v \in V$. The map s is called an isomorphism. Intuitively, two attributes (columns) are isomorphic if and only if one column turns into another one by properly renaming its attribute values.

Let $K = (V, A)$ and $H = (V, B)$ be two information tables, where $A = \{A^1, A^2, \ldots A^n\}$ and $B = \{B^1, B^2, \ldots B^m\}$. Then, K and H are said to be isomorphic if every A^i is isomorphic to some B^j, and vice versa. The isomorphism of relations is reflexive, symmetric, and transitive, so it classifies all relations into equivalence classes; we call them isomorphic classes.

Definition 9.1.1. *H is a simplified relational table of K, if H is isomorphic to K and only has non-isomorphic attributes.*

Theorem 9.1.2. *Let H be the simplified relational table of K. Then the patterns (large itemsets) of K can be obtained from those of H by elementary operations that will be defined below.*

To prove the Theorem, we will set up a lemma, in which we assume there are two isomorphic attributes B and B' in K, that is, degree K – degree $H = 1$. Let $s : Dom(B) \longrightarrow Dom(B')$ be the isomorphism and $b' = s(b)$. Let H be the new table in which B' has been removed.

Lemma 9.1.3. *The patterns of K can be generated from those of H by elementary operations, namely,*

1. If b is a large itemset in H, then b' and (b, b') are large in K.
2. If (a. ., b, c. . .) is a large itemset in H, then (a. ., b', c. . .) and (a. ., b, b', c,. . .) are large in K.
3. These are the only large itemsets in K.

The validity of this lemma is rather straightforward; and it provides the critical inductive step for Theorem; we ill skip the proof.

9.2 Semantics Issues

The two relations, Tables 7 and 8, are isomorphic, but their semantics are completely different. One table is about part, the other is about suppliers. These two relations have Isomorphic association rules;

1. Length one: TEN, TWENTY, March, SJ, LA in Table 7 and

Table 7. An Relational Table K

V	K	($S\#$	Business Amount (in m.)	Birth Day	CITY)
v_1	$\longrightarrow$	(S_1	TWENTY	MAR	NY
v_2	$\longrightarrow$	(S_2	TEN	MAR	SJ
v_3	$\longrightarrow$	(S_3	TEN	FEB	NY
v_4	$\longrightarrow$	(S_4	TEN	FEB	LA
v_5	$\longrightarrow$	(S_5	TWENTY	MAR	SJ
v_6	$\longrightarrow$	(S_6	TWENTY	MAR	SJ
v_7	$\longrightarrow$	(S_7	TWENTY	APR	SJ
v_8	$\longrightarrow$	(S_8	THIRTY	JAN	LA
v_9	$\longrightarrow$	(S_9	THIRTY	JAN	LA

Table 8. An Relational Table K′

V	K	($S\#$	Weight	Part Name	Material
v_1	$\longrightarrow$	(P_1	20	SCREW	STEEL
v_2	$\longrightarrow$	(P_2	10	SCREW	BRASS
v_3	$\longrightarrow$	(P_3	10	NAIL	STEEL
v_4	$\longrightarrow$	(P_4	10	NAIL	ALLOY
v_5	$\longrightarrow$	(P_5	20	SCREW	BRASS
v_6	$\longrightarrow$	(P_6	20	SCREW	BRASS
v_7	$\longrightarrow$	(P_7	20	PIN	BRASS
v_8	$\longrightarrow$	(P_8	30	HAMMER	ALLOY
v_9	$\longrightarrow$	(P_9	30	HAMMER	ALLOY

2. Length one: 10, 20, Screw, Brass, Alloy in Table 8
3. Length two: (TWENTY, MAR), (Mar, SJ), (TWENTY, SJ)in one Table 7,
4. Length two: (20, Screw), (screw, Brass), (20, Brass), Table 8

However, they have very non-isomorphic semantics:

1. Table 7: (TWENTY, SJ), that is, the business amount at San Jose is likely 20 millions; it is isomorphic to (20, Brass), which is not interesting.
2. Table 8: (SCREW, BRASS), that is, the screw is most likely made from Brass; it is isomorphic to (Mar, SJ), which is not interesting.

References

1. R. Agrawal, T. Imielinski, and A. Swami, “Mining Association Rules Between Sets of Items in Large Databases,” in Proceeding of ACM-SIGMOD international Conference on Management of Data, pp. 207–216, Washington, DC, June, 1993 64
2. A. Barr and E. A. Feigenbaum, The handbook of Artificial Intelligence, Willam Kaufmann 1981. 66
3. G. Birkhoff and S. MacLane, A Survey of Modern Algebra, Macmillan, 1977 72
4. Richard A. Brualdi, Introductory Combinatorics, Prentice Hall, 1992. 75
5. C. J. Date, C. Date, An Introduction to Database Systems, 7th ed., Addison-Wesley, 2000. 64
6. Margaret H. Dunham, Data Mining Introduction and Advanced Topics Prentice Hall, 2003, ISBN 0-13-088892-3 64
7. U. M. Fayad, G. Piatetsky-Sjapiro, and P. Smyth, “From Data Mining to Knowledge Discovery: An overview.” In Fayard, Piatetsky-Sjapiro, Smyth, and Uthurusamy eds., Knowledge Discovery in Databases, AAAI/MIT Press, 1996. 66
8. H Gracia-Molina, J. Ullman, and J. Windin, J, Database Systems The Complete Book, Prentice Hall, 2002. 64, 66, 68, 69
9. T. T. Lee, “Algebraic Theory of Relational Databases,” The Bell System Technical Journal Vol 62, No 10, December, 1983, pp. 3159–3204. 72

10. T. Y. Lin, "A mathematical Theory of Association Mining" In: Foundation and Novel Approach in Data Mining (Lin & et al), Spriner-Verlag, 2005, to appear. 63, 75
11. T. Y. Lin, "Mining Associations by Solving Integral Linear Inequalities," in: the Proceedings of International Conference on Data Mining, Breighton, England, Nov 1–4, 2004. 75
12. T. Y. Lin "Attribute (Feature) Completion – The Theory of Attributes from Data Mining Prospect," in: the Proceedings of International Conference on Data Mining, Maebashi, Japan, Dec 9–12, 2002, pp. 282–289. 68
13. T. Y. Lin, "Data Mining and Machine Oriented Modeling: A Granular Computing Approach," Journal of Applied Intelligence, Kluwer, Vol. 13, No 2, September/October, 2000, pp. 113–124. 71, 72, 75
14. T. Y. Lin and M. Hadjimichael, "Non-Classificatory Generalization in Data Mining," in Proceedings of the 4th Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery, November 6–8, Tokyo, Japan, 1996, 404–411.
15. E. Louie, T. Y. Lin, "Semantics Oriented Association Rules," In: 2002 World Congress of Computational Intelligence, Honolulu, Hawaii, May 12–17, 2002, 956–961 (paper # 5702) 75
16. E. Louie and T. Y. Lin, "Finding Association Rules using Fast Bit Computation: Machine-Oriented Modeling," in: Foundations of Intelligent Systems, Z. Ras and S. Ohsuga (eds), Lecture Notes in Artificial Intelligence 1932, Springer-Verlag, 2000, pp. 486–494. (ISMIS'00, Charlotte, NC, Oct 11–14, 2000)
17. H. Liu and H. Motoda, "Feature Transformation and Subset Selection," IEEE Intelligent Systems, Vol. 13, No. 2, March/April, pp. 26–28 (1998) 64
18. Hiroshi Motoda and Huan Liu "Feature Selection, Extraction and Construction," Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol 5, No. 2, May 2002, pp. 67–72. (proceeding for the workshop "Toward the Foundation on Data Mining" in PAKDD2002, May 6, 2002. 73
19. Z. Pawlak, Rough sets. Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, 1991 71, 72
20. Z. Pawlak, Rough sets. International Journal of Information and Computer Science **11**, 1982, pp. 341–356.
21. J. Ullman, Principles of Database and Knowledge-Base Systes, Vol 1, II, 1988, 1989, Computer Science Press. 71