Representing Discovered Patterns Using Attributed Hypergraph

Yang Wang and Andrew K. C. Wong
Pattern Analysis and Machine Intelligence Group
Department of Systems Design Engineering
University of Waterloo
Waterloo, Ontario N2L 3G1, CANADA
{wang, akcwong}@watnow.uwaterloo.ca

Abstract

One of the fundamental problems in knowledge discovery in databases and other applications of AI is how to represent knowledge and patterns. Existing representation schemes have various shortcomings. In this paper, we propose a new knowledge representation scheme using *attributed hypergraph* (*AHG*), which is simple yet general enough to directly encode different order patterns discovered from large databases. In *AHG*, both the qualitative and quantitative relations are represented as attributed hyperedges. Such representation is lucid and transparent for visualization. Besides, patterns in *AHG* are easy to understand. In the discussion, some basic manipulations of *AHG* for data mining tasks are briefly addressed. The paper ends with examples of pattern representation using *AHG*.

Introduction

For most applications of AI, including machine learning and KDD, the choice of knowledge representation is a difficult task. Woods (Woods 1983) suggests that two measurements, *expressive adequacy* and *notational efficiency*, should be used to evaluate the performance of a knowledge representation.

By knowledge discovery in databases, or Data Mining, we mean automatically process from databases large quantities of data; identify the significant and meaningful patterns; and represent them in a form suitable for achieving the user's goal (Matheus & Piatetsky-Shapiro 1993). Since the goals of such a system are often vaguely defined and change with time, knowledge representation tends to be more important for a KDD system than a conventional classification system. In addition to the requirements proposed by Woods, several other aspects should be considered. First, the representation scheme should offer a mechanism for easy knowledge re-organization or focus on a certain portion of the knowledge to meet the changing goal. Secondly, the represented knowledge should be transparent, easy to be visualized and understood. Since data in real world databases usually contains noise and uncertainty, patterns extracted by a KDD system are generally probabilistic. It is required that numerical inferences be supported by the representation in addition to logical inference. Finally, since the patterns detected from large databases could be of different orders, and since high order patterns cannot be induced by lower order relations (Wong & Wang 1995), different order patterns should be explicitly represented.

In this paper, after a brief review of popular representation, we propose a new knowledge representation based on *attributed hypergraph* (*AHG*), which is simple yet general enough to encode different order patterns. With such representation, both the qualitative and the quantitative relations are explicitly represented and are easy to understand.

Representation Schemes for KDD

Over the years, numerous knowledge representation schemes have been reported. The most popular ones are decision tree, networks, production rule and logic.

Decision tree is a simple representation popularized by Quinlan's ID3 and successfully applied to inductive learning. Decision tree based systems are found in a wide range of application domains, mostly in the classification-oriented areas. A disadvantage of decision tree is its difficulty for humans to interpret, especially from the viewpoint of expert systems (Smyth & Goodman 1992) and KDD systems (Holsheimer & Siebes 1995). Also, trees are not designed to deal with missing attribute information (Smyth & Goodman 1992). Moreover, since decision trees are mainly designed for classification purposes, they are not suitable for multi-attribute prediction (Fisher 1987).

Trees can be considered as a special case of graphs. Graph representations, such as Bayesian and Markov networks, usually provide more general methods to represent patterns. They directly represent the first order associations between two nodes by links. How-

ever, as observed by Pearl (Pearl 1988), graph-based representation, including trees and networks, cannot distinguish between set connectivity and connectivity among their elements. Hence, they are not general enough for representing different order patterns.

Production (if-then) rule is another scheme widely used in expert systems and classification oriented tasks. It explicitly presents the association between a set of observations (left-hand antecedent) and one attribute value (right-hand consequent). Rules are considered easier to understand than trees. However, in KDD applications, with each changing interest, the values of different attributes have to be predicted. Besides, a huge number of rules have to be obtained. This is sometimes impractical in the real world (Wong & Wang 1996). In this case, we need a scheme which can easily re-organize the represented knowledge for different goals of the system.

In addition to attribute (proposition) based representations, relational representations such as Horn clause (see (Kowalski 1979) for an overview) and First Order Logic (see (Muggleton 1992) for an overview) are used in learning systems. They are very powerful and expressive formalisms. Since they are originally designed to formalize mathematical reasoning and later used in logic programming, patterns in them are deterministic rather than probabilistic. To do probabilistic reasoning, special adoptions have to be done. This problem also exists in the structured representations such as semantic networks. Besides, logic based representations are considered less comprehensible and harder to visualize than graph based representations.

The AHG Representation

To overcome the shortcomings of the traditional representations, we here propose an *attributed hypergraph* representation to depict the associations of patterns in a data set. *AHG* is a direct, simple and efficient representation for describing the information at different and/or mixed levels of abstraction. It has been successfully used in 3D scene interpretation and object recognition (Wong & Rioux 1990). In *AHG*, both the qualitative relations (the structure of the hypergraph) and the quantitative relations (the attribute values of vertices and hyperedges) are encoded. Since *AHG* representation is lucid and transparent for visualization, interpretation of different order patterns can be easily achieved. A good number of mature graph algorithms can be adopted to implement various operations for pattern retrieval and re-organization. The computational complexity of this representation will be related to the complexity of the algorithms performing graph operations. Before proposing the *AHG* representation, we first formalize the definition of a pattern.

Pattern as Event Association in Database

Consider that we have a database D containing M instances. Every instance is described in terms of N fields, $\mathbf{X} = \{X_1, \cdots, X_N\}$. Then each field, X_i, $1 \leq i \leq N$, can be seen as a random variable taking on values from its domain $Dom(X_i)$. In this manner, each instance in D is a realization of $\mathbf{X}$, denoted as $\mathbf{x}_j = \{x_{1j}, \cdots, x_{Nj}\}$, where x_{ij} can assume any value in $Dom(X_i)$.

A *component* of D is either a field or any possible value (range) of a field. Any field X_i, $1 \leq i \leq N$ is a component. *True* can be a component if it is a possible value of a field. An interval $(25, 50)$ can also be a component if it belongs to a domain. An *atomic event*, or *event* for short, is defined as the relationship between two components. Thus, any realizations of the fields, such as $X_1 = True$ and $X_2 \in (25, 50)$ are atomic events. The relationships between two fields such as $X_1 < X_2$, $X_1 \neq X_2$ and $X_1/X_2 = 2.5$ are also events if they are meaningful. A *compound event*, or *composite* for short, is a set of atomic events and/or compound events. The *order* of a composite is its cardinality. Any first order composite is an atomic event. Thus, $[X_1 = True, X_2 \in (25, 50)]$ is a second order composite. A *sub-composite* of a composite is a subset of the composite. Let T be a statistical significance test. If a composite c passes the test, we say that c is a *significant pattern*, or simply a *pattern*, of *order* $|c|$. The elements of c are said to have a *statistically significant association according to* T or simply they are *associated.*

We argue that most patterns in a database can always be described as event associations. An if-then rule can be seen as an association between its left-hand composite and its right-hand event. Due to the noise in a database, patterns are probabilistic rather than deterministic. In a real world database, the existence of higher order patterns does not guarantee the existence of lower order patterns and *vice versa* (Wong & Wang 1995). Hence, whether or not a composite is a pattern cannot be determined by examining its sub-composites and *vice versa.* This implies that, in general, higher order patterns cannot be synthesized from the lower order ones (Wong & Wang 1995). It requires that different order patterns be represented explicitly.

Representing Patterns in AHG

Let us first give a formal definition of hypergraph.

Def. 1. (Berge 1989) Let $Y = \{y_1, y_2, \cdots, y_n\}$ be a finite set. A *hypergraph* on Y is a family $H = (E_1, E_2, \cdots, E_m)$ of subsets of Y such that

1. $E_i \neq \phi \quad (i = 1, 2, \cdots, m)$, and
2. $\bigcup_{i=1}^{m} E_i = Y$.

The elements $y_1, y_2, \cdots, y_n$ of Y are called *vertices*, and the sets $E_1, E_2, \cdots, E_m$ are the *edges* of the hypergraph, or simply, *hyperedges*.

Def. 2. A *simple hypergraph* is a hypergraph H with hyperedges $(E_1, E_2, \cdots, E_m)$ such that

$$E_i = E_j \Rightarrow i = j.$$

Unless otherwise indicated, we refer to *hypergraph* as *simple hypergraph*.

Def. 3. An *attribute* of a hypergraph is a data structure associated with a hyperedge or a vertex.

Def. 4. An *attributed hypergraph* is a hypergraph such that each of its hyperedges and vertices has an attribute.

In *AHG* representation, each *vertex* represents an atomic event. Each pattern or statistically significant association is represented by a *hyperedge*. The *rank* (*anti-rank*) of a hypergraph is the highest (lowest) order of the patterns detected from the database. For an event e, the *star* $H(e)$ of hypergraph H with center e represents all the patterns related to the event e. Let A be a subset of all atomic events, the *sub-hypergraph* of hypergraph H induced by A represents the event associations in A.

The attributes of both the vertices and the hyperedges depend on the application and the pattern discovery algorithm applied. In (Wong & Wang 1996), we proposed a statistical pattern discovery method based on adjusted residual analysis. In such a case, the attribute of each vertex is the marginal probability of the corresponding atomic event. The attribute of each hyperedge contains the probability of the compound event, the expected probability of the compound event, and the probabilities of sub-compound events one order lower. All of these attributes will be useful for the inference process. Therefore, hyperedges depict the qualitative relations among their elementary vertices, while the attributes associated with the hyperedges and the vertices quantify these relations.

Fig. 1 shows some generalized cases of different order significant associations. The upper part of each case in this figure depicts the event occurrences and their pairwise associations, while the lower part furnishes the hypergraph representation of the associations (attributes not shown). This figure also illustrates that the existence of higher order patterns does not guarantee the existence of lower order patterns and *vice versa*. For instance, Case 3 shows a situation where third order pattern $[A, B, C]$ exists, but there is no second order association between A, B and C. Case 4 depicts a contrary instance such that all of the three second order patterns exist but not the third order pattern.

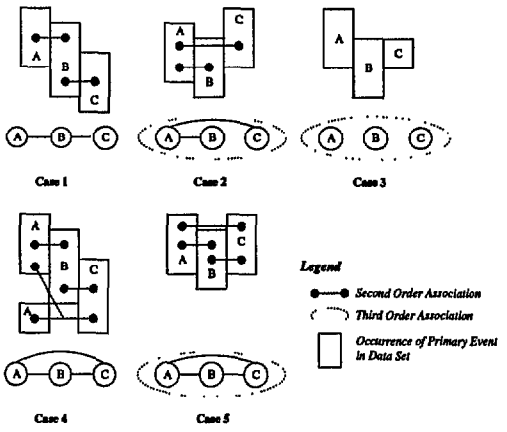

Figure 1: Different Order Significant Associations

Within the *AHG* framework, to manipulate patterns is to operate on the hyperedges, vertices and their attributes. To re-organize knowledge is to select sub-hypergraphs according to the current system goal. If we are classifying a new instance against a field X_1, only the hyperedges containing an event of X_1 are interesting. If the system is later asked to find the patterns related to event $X_2 = True$, only the hyperedges containing this event are focused on. Thanks to a good number of mature algorithms on graphs, these kinds of operations are expected to be computationally efficient. Most database mining problems can be classified into three categories: association, classification, and sequence (Agrawal & Swami 1993). In the *AHG* framework, associations among events are represented as hyperedges. When we consider class labels as a special field, classification can always be treated as using patterns related to this special field to predict the class of a new object. The sequential problem is just a special case of association with a time tag attached.

How to operate on an *AHG* is also application dependent. Basic operators include *Construct()* which constructs an attributed hypergraph from a database, *HighestOrder()* and *LowestOrder()* which find the highest (lowest) order of detected relationships, *FindRelation()* which extracts all the patterns related to a specified event, and *FindSubEvent()* which extracts all patterns that contain a given composite or its non-empty sub-composites. The last one is to find all the compound events which are considered relevant to the inference process from a set of facts.

Examples of AHG Representation

XOR is a typical high order problem. Case 1 of Fig. 2 shows all the patterns found by applying the algorithm in (Wong & Wang 1995). We note right away that there are only third order patterns. If we are inter-

ested in only the patterns related to $C = F$, then a sub-hypergraph shown by Case 2 is extracted. This hypergraph is equivalent to the rule: $(A = T \wedge B = T) \vee (A = F \wedge B = F) \Rightarrow C = F$.

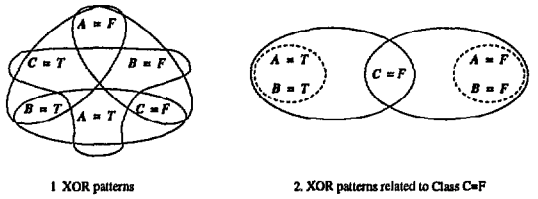

Figure 2: AHG Representation of XOR Patterns

In the breast cancer database (Wolberg & Mangasarian 1990), each sample is described by 10 attributes and classified into one of the two classes. Fig. 3 shows part of the patterns detected by applying the algorithm proposed in (Wong & Wang 1995) and (Wong & Wang 1996). Here, the values of hypergraph attributes are not shown. To make explicit the class and attribute association, we single out the classes (*benign* and *malignant*). Their associations with other atomic or compound events are shown by the solid lines. Significant compound events associated with a class are enclosed by dotted curves. This *AHG* shows that: 1) any composite c can only be associated with only one of the two classes; and 2) if c is associated with one class, none of the sub-composite of c would appear in hyperedges related to the other class (i.e. the two classes are totally separated). It implies that, theoretically, we can achieve 100% classification accuracy.

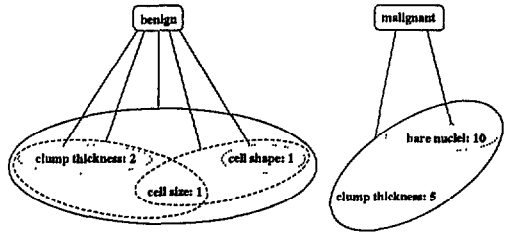

Figure 3: Part of AHG for Breast-Cancer Database

Fig. 4 is the *AHG* of all the second order patterns related to the field *Severity* discovered from a company's injury database. The number on the line indicates the significant level of the pattern. A dash line shows that the pattern is negative, which means that the two connected events are unlikely to happen together. From the figure, for example, we can see that two fields, *Injury_Type* and *Department* have relations with *Severity*. If *Injury_Type* is 1, *Severity* will be higher than 1. The most probable *Severity* level will be 2, since this pattern has the highest significant level. On the other hand, only one event of *Department* is related to *Severity*. It depicts that workers in *Department* 1 normally do not have injuries of *Severity* level 2.

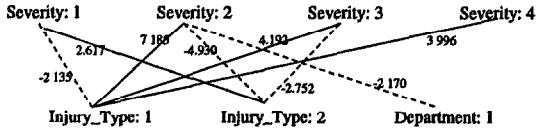

Figure 4: Second Order Patterns in an Injury Database

Summary

This paper presents a new pattern representation for KDD. Here, different order patterns are explicitly represented in the form of *AHG* which allows the user to analyze the data at different levels of abstraction. In this *AHG* framework, to re-organize knowledge, relations can be used to induce new hyperedge. Such representation encodes both qualitative and quantitative patterns. Since the framework is transparent to the user, knowledge can be visualized and interpreted by humans without difficulty. Current work concentrates on the inference processes using *AHG* knowledge representation for various data mining tasks.

References

Agrawal, R.; Imielinski, T., and Swami, A. 1993. Database mining: A performance perspective. *IEEE Trans. on KDE* 5(6):914–925.

Berge, C. 1989. *Hypergraph: Combinatorics of Finite Sets.* North Holland.

Fisher, D. H. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine Learning* 2(2):139–172.

Holsheimer, M., and Siebes, A. 1995. Data mining: The research for knowledge in databases. Technical Report CS-R9406, CWI.

Kowalski, R. 1979. *Logic for Problem Solving.* North Holland.

Matheus, C. J.; Chan, P. K., and Piatetsky-Shapiro, G. 1993. Systems for knowledge discovery in databases. *IEEE Trans. on KDE* 5(6):903–913.

Muggleton, S. 1992. *Inductive Logic Programming.* Academic Press.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann.

Smyth, P., and Goodman, R. M. 1992. Information theoretic approach to rule induction from database. *IEEE Trans. on KDE* 4(4):301–316.

Wolberg, W. H., and Mangasarian, O. L. 1990. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In *Proc. of the Nat. Aca. of Sci.*, volume 87.

Wong, A. K. C.; Lu, S. W., and Rioux, M. 1990. Recognition and shape synthesis of 3-d object based on attributed hypergraph. *IEEE Trans. on PAMI* 11(3):279–290.

Wong, A. K. C., and Wang, Y. 1995. Discovery of high order patterns. In *Proc. of IEEE Int'l Conf. on SMC*, 1142–1148.

Wong, A. K. C., and Wang, Y. 1996. High order pattern discovery from discrete-valued data. *IEEE Trans. on KDE.* accepted.

Woods, W. A. 1983. What's important about knowledge representation. *Computer* 16(10).