KNOWLEDGE DISCOVERY FROM DATABASES: THE NYU PROJECT

James Clifford
Vasant Dhar
Alex Tuzhilin

Stern #IS-95-12

KNOWLEDGE DISCOVERY FROM DATABASES: THE NYU PROJECT

James Clifford
Vasant Dhar
Alex Tuzhilin

Department of Information Systems
Leonard N. Stern School of Business
New York University
44 West 4th Street, Suite 9-170
New York, NY 10012-1126

(212) 998-0800

Working Paper Series
Stern #IS-95-12

Knowledge Discovery from Databases: The NYU Project

James Clifford
Vasant Dhar
Alex Tuzhilin
Information Systems Department
Stern School of Business
New York University

February 15, 1995

Abstract

More and more application domains, from financial market analysis to weather prediction, from monitoring supermarket purchases to monitoring satellite images, are becomingly increasingly data-intensive. The result is massive databases that are growing at a rapid rate — it has been estimated that the world's electronic data almost doubles every year. With this rate of data explosion, there is a pressing need for computers to play an increasing role in analyzing these huge data repositories which are impossible to penetrate manually. The challenge is to ferret out the regularities in the data that will prove to be interesting to the user.

A group in the Information Systems department at the NYU Business School has been working in this area for a number of years. The focus of our project is now on the discovery of patterns from time series data. In this paper we give an overview of the kinds of databases we are "mining" and the kinds of temporal patterns and rules which we are attempting to discover. In the first phase of this research, we have developed a taxonomy of patterns as a way to organize our research agenda. We wish to share the taxonomy with the research community in the "knowledge discovery in databases" area since we have found it useful in classifying the universe of regularities or patterns into distinct types, that is, patterns which differ in terms of their structure and the amount of search effort required to find them. Although the primary focus of our project is on time series data, and the examples we will present are chosen from this arena, the taxonomy is general enough to apply to any type of data.

1 Motivation and Goals

More and more application domains, from financial market analysis to weather prediction, from monitoring supermarket purchases to monitoring satellite images, are becomingly increasingly data-intensive. The result is massive databases that are growing at a rapid rate —it has been estimated that the world's electronic data doubles every 20 months [FPSM91]. With this rate of data explosion, there is a pressing need for computers to play an increasing role in analyzing these huge data repositories which are impossible to penetrate manually. NASA observation satellites, for example, send back a terabyte of telemetry data daily [FPSM91]. Large financial institutions collect hundreds of gigabytes of data every day. Likewise there is an abundance of medical data on a variety of virologic, cancer, and other kinds of diseases that is growing rapidly. Ferreting out interesting patterns from large amounts of this data *must* be done by the computer.

The focus of our group is on time series data. This is because much of the data that are flowing into large databases, whether they be business, medical, or scientific, are inherently temporal. Not surprisingly then, interesting patterns often tend to be temporal. Consider, for example, the set of time-series plots shown in Figure 1. Many interesting patterns lie hidden in such temporally rich databases, such as the fact that interest rates and the large capitalization stock index tend to move in the same direction, or that humidity tends to become very volatile after periods of low rainfall.

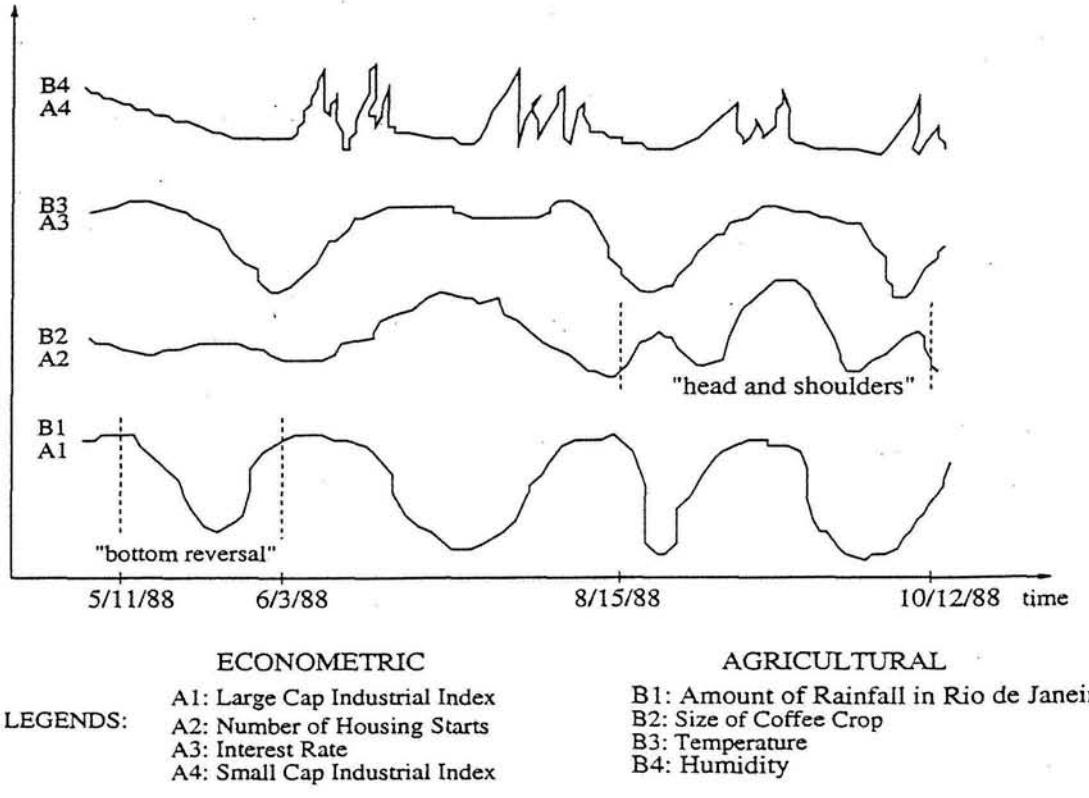

Figure 1: Some Typical Time Series Functions

Although our focus in this research project is on time series data, our framework for

classifying patterns is general enough to apply to data with any kind of ordering (other than time) and also to cross-sectional data.

2 The Type of Data to Be Explored

Let us consider the line graphs in Figure 1, which show four variables plotted over time. The legends (A and B) show two possible sets of labels for the graphs, one in an econometric and the other in an agricultural domain. In the agricultural application, the items of interest are the amount of rainfall in some area (say, Rio de Janeiro), along with its temperature, coffee crop, and humidity, all plotted over time. Appropriate scales can be chosen on the vertical axis. The second labeling, from the econometric domain, has as its corresponding data a large capitalization industrial index (a weighted average of large capitalization stocks), interest rate, the number of housing starts, and a small capitalization industrial index. We can envision various other types of data, similarly plotted, for other application areas.

How many different patterns can be extracted from the data corresponding to Figure 1? A few patterns seem fairly apparent. For example, "temperature (B3) and rainfall (B1) often seem to move in the same direction." Also, "the coffee crop (B2) seems to rise after a "cool dry spell" (B3 & B4)." Likewise, for legend A, the corresponding patterns are that "interest rate (A3) and the large capitalization index (A1) often seem to move in the same direction," and that "housing starts (A2) seem to jump after a period of lower interest rates (A3)." However, there are also some more subtle patterns in the data. For example, "the small capitalization index (A4) seems to become very volatile after a drop in the large capitalization index (A1)."

In describing the trends in the graphs, we made use of several kinds of terms, which can be viewed as part of an application-specific *vocabulary*. For example, a "bottom reversal" is a term used by technical analysts of economic indicators [LR78] to express concisely an interesting (to them) progression of data values plotted in time. Similarly, economists refer to "seasonal unemployment" or "peaks" as part of their language. The same is true in most application areas, e.g. medicine ("night fevers"), biology ("hibernation periods"), and other areas of specialization involving temporally rich data. Such terms, which we call *temporal predicates*, make explicit reference to the temporal dimension. One issue that we must address is that there is frequently some degree of fuzziness in these terms, in the sense that there can be a certain tolerance for variations from the "typical" trend.

We also noted that housing starts jump *after* a period where the average interest rates are low. Accordingly, our vocabulary must also consist of a rich set of purely *temporal predicates* such as "*after*", "*immediately after*", "*before*", etc. These predicates refer exclusively to time. Although there is a well-known *complete* set of these predicates [All84], we must be able to deal with imprecision in matching them with the data. For example, $immediately\_after(t3, t2)$ is true if the value of $t3$ is greater than that of $t2$ by less than some specified "small" amount.

In addition, we need *temporal aggregate functions*, which specify how the raw data can be aggregated over time. For example, averaging, summation, maximizing, minimizing, and counting are examples of aggregates which can be applied to raw data. For example, $average\_housing\_starts(housing\_starts, t1, t2)$ would return the average number of housing

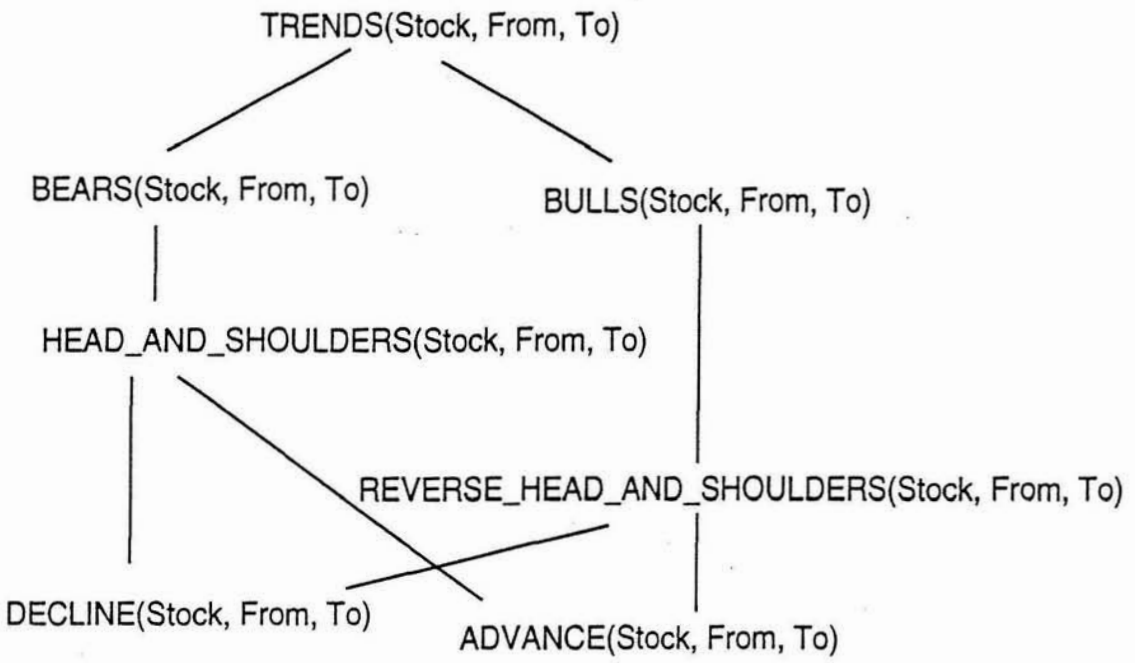

Figure 2: Example of a Classification Hierarchy of Technical Analysis Patterns.

starts in the interval $(t1, t2)$. Likewise $average\_volatility * (stock, t1, t2)$ would return the standard deviation of averaged daily returns of a stock in the interval $(t1, t2)$.

Note that terms such as "bottom reversal" are user-defined predicates built out of a set of more primitive predicates. These user-defined predicates essentially provide *relational views* [Ull88] on the original data and are useful since patterns are often expressed in terms of such views on the original data. More precisely, a user-defined predicate over some database is a disjunction of conjunctive clauses, i.e., is of the form $\vee_{i=1}^{m}(\wedge_{j=1}^{n} P_{ij})$, where each atomic formula P_{ij} is either (i) a database relation, (ii) relational operator, such as $=$, $<$, $\leq$, defined either on temporal values or on the domain values (including attributes from database relations), (iii) another user-defined predicate.

Since a user-defined predicate can be defined in terms of other user-defined predicates, we can represent this relationship in terms of a classification hierarchy, such as the one in Figure 2. This hierarchy is useful for providing a vocabulary for expressing and searching for patterns.

Finally, we note that we have all along assumed an underlying data type called a time series, a function from some bounded time interval into some underlying domain of values. Formally, let DOM be a domain (any set) and T be a *bounded* time interval ($T = [L, R]$). Then a *time series* is a function $TS : T \rightarrow DOM$. We assume that this data type is supported in some way — either directly or indirectly — by the underlying DBMS. Time series can be specified with various representation methods. For example, it can be represented with a Fourier transform. In this case, the frequency plot will be a "description" of the time series. Alternatively, the time series can be approximated with a set of polynomials based on the approaches used in the wavelet theory [Chu92]. Finally, the extensional representation of the function itself as a set (typically, very large) of $< time, value >$ pairs might sometimes be appropriate.

To summarize, in addition to the database, we make use of what can be termed a *data dictionary* which can be viewed as a layer that enables a pattern discovery system to focus its attention on things that will be meaningful to the user. This vocabulary consists of temporal predicates, some of which can be defined by the user in terms of other temporal predicates, temporal aggregate functions, and time series.

3 The Type of Knowledge to Be Discovered

There is no standard definition of the term "pattern" in the literature. In trying to draw a common thread through a collection of papers on "Knowledge Discovery in Databases," Frawley et.al. [FPSM91] define patterns as follows:

> Given a set of facts (data) F, a language L, and some measure of certainty C, a pattern S is a statement S in L that describes relationships among a subset F_S of F with certainty C, such that S is simpler (in some sense) than the enumeration of all facts in F_S.

Their definition is intentionally vague to cover a wide variety of approaches.

The temporal domain brings additional considerations to the concept of a pattern. For example, in Figure 1 the number of housing starts (A2) from 8/15/88 to 10/12/88 exhibits a pattern known as *head-and-shoulders* [LR78]. As another example of a temporal pattern consider the statement "if a price correction to a stock is seen immediately preceding the announcement of big news, then insider trading in that stock is likely during that time." This pattern can be expressed as a temporal rule described below.

The first type of pattern mentioned above, which we call *predicate-based* patterns, maps a time series of data into the interval [0,1]. Let **TS** be a set of all time series TS defined over the time interval T, and let **I** be a set of all possible time subintervals over T. Then a *predicate-based* pattern is a function:

$$PAT \ : \ \mathbf{TS} \times \mathbf{I} \rightarrow [0,1]$$

The interval [0,1] captures the "fuzziness" associated with the patterns. A certain pattern, for example, might be the "prototypical" head-and-shoulders in which case a value of 1 would be assigned to the pattern. Lower values would indicate deviations from the prototypical pattern, with 0 denoting no match at all. A special case of the above are patterns that assign only 1 and 0 (or TRUE and FALSE) to time series, are therefore like standard predicates in logic.

We will denote a pattern over the time interval $[t_1, t_2]$ for the time series TS as $PAT(TS, t_1, t_2)$. For example, $bottom\_reversal(daily\_rainfall, 5/11/88, 6/3/88)$ means that the daily rainfall in Rio de Janeiro exhibits the *bottom_reversal* pattern between 5/11/88 and 6/3/88.

Predicate-based patterns can be defined in many ways. For example, the *bottom_reversal* pattern can be defined in terms of the first-order logic as:

$$bottom\_reversal(x, t_1, t_2) \ = \ (\exists t)(t_1 < t < t_2 \ \wedge \ monotone\_decrease(x, t_1, t) \ \wedge \ monotone\_increase(x, t, t_2))$$

where $monotone\_increase(x, t_1, t_2)$ and $monotone\_decrease(x, t_1, t_2)$ are user-defined predicates saying that "by and large" x is monotonically increasing (decreasing) between times t_1 and t_2. These predicates can be defined in an obvious way in first-order terms. Alternatively, *bottom_reversal* can be defined by taking a Fourier transform of a pattern and then identifying certain types of frequencies in the transformed pattern. Another way to define *bottom_reversal* pattern is to use the pattern recognition approach by identifying a set of features, dividing the space of all features into different regions and then computing into which region a certain pattern falls.

The second type of pattern can be defined as a *rule*, following the approach proposed in [DT93], i.e. as an expression of the following form:

$$p_1 \wedge \ldots \wedge p_n \rightarrow q\ [c] \tag{1}$$

where p_i, $i = 1, \ldots, n$ and q are any of the predicates discussed in Section 2 (temporal or non-temporal database relations, user-defined predicates and predicate-based patterns) or negations of these predicates, or relational operators $(=, <, \leq, \geq, >)$ defined on constants, variables or user-defined functions, and c is a number (for instance, a conditional probability) in the interval [0,1] that indicates the degree to which the rule holds. Some of the predicates and functions in the rules are temporal and others are non-temporal. We make the usual assumptions about our rules, i.e, there are no negations in the head of a rule, and the patterns are *safe* [Ull88], so there is a finite number of tuples satisfying the body of a rule.

We provide examples of some patterns expressed as rules below.

> If a price correction in a stock is seen before the announcement of big news about the company, then insider trading is likely

$$correction(Stock, t1, t2) \wedge big\_news(Stock, t3) \wedge immediately\_after(t3, t2) \rightarrow insider\_trading(Stock, t1, t2)\ [75\%] \tag{2}$$

where *correction, big_news, immediately_after, insider_trading* are user-defined temporal predicates. The next pattern provides an example of a predicate-based pattern and a user-defined function appearing in a rule.

> If a stock shows a head and shoulders pattern, and investor cash levels are low, then a bearish period is likely to follow

$$head\_and\_shoulders(stock, t1, t2) \wedge avg\_investor\_cash * (cash\_index, t1, t2) < low\_cash \wedge immediately\_after(t, t1, t2) \rightarrow bearish(stock, t)\ [62\%] \tag{3}$$

where *head_and_shoulders* is a predicate-based pattern and *avg_investor_cash** is the user-defined aggregate function returning the average levels of investor cash between times $t1$ and $t2$.

4 A Categorization of Knowledge Discovery Tasks

How should we categorize the different types of patterns that can be discovered from data? To appreciate what we mean by "different types of patterns, consider the following example.

	Validation	Generation
Predicates	I	III
Rules	II	IV

Table 1: Types of Knowledge Discovery Tasks.

Suppose a financial analyst is perusing a time series of prices for a particular bond, and finds an interesting pattern, where its price underwent a sudden price reversal. He wants the system to find all similar cases. Contrast this with a case where the analyst to know all cases where the price went through "unusual swings". Intuitively, cases of the second type seem like they would be harder for a system to discover. The goal is less precise, making the search a lot harder.

We believe that a reasonable and useful way to categorize pattern discovery tasks is in terms of the representation scheme for knowledge that the system utilizes. As described in the previous section, our representation of "knowledge" is in the form of patterns represented either as predicates, such as *head_and_shoulders(Stock, From, To)*, or as rules containing these predicates, such as the "insider trading" rule 2 presented in Section 3. Thus, one dimension along which we can distinguish the activities of the system is whether the *unit of discovery* is an individual predicate or a complete rule.

The second dimension for distinguishing between different types of discovery is whether the discovery is of *new* pieces of "knowledge" — i.e., the generation of new predicates or rules previously unknown to the system — or the *validation* or search for existing templatess — i.e., discerning the occurrence of a known pattern at a particular point or over a particular interval in time, or adjusting the certainty of a known rule (note that in both of these cases of "validation", the "existing" predicates or rules may have been asserted by the user of the system, or may themselves have been "discovered" previously by the system itself).

Categorizing patterns in terms of the above two dimensions leads to a two-by-two classification of the knowledge discovery tasks to be explored, as shown in Table 1. The two dimensions of this matrix represent orthogonal dimensions of the discovery activity. The *predicates/rules* dimension refers to the unit of the discovery task, and since *predicates* are the components of *rules* it is natural to consider them in this order. The *validation/generation* dimension refers to the input to the discovery task. In *validation* the system focuses on a particular pattern and determines whether it holds in the data, whereas in *generation* the system attempts to discover new patterns on its own.

The following sections analyze the four types of discovery shown in Table 1. Discovery tasks of Class I involve the validation of previously defined facts or concepts, expressed in the form of predicates. Discovery tasks of Class II involve the validation of previously asserted rules. Discovery tasks of Class III involve the discovery of new, interesting predicate-based patterns that occur in the database. Discovery tasks of Class IV involve the discovery of new rules, which consist of interesting relationships among predicates.

4.1 Class I Tasks

Discovery tasks of Class I involve the validation of previously defined facts or concepts, expressed in the form of predicates. This activity therefore involves the evaluation of user-defined predicates or views over the underlying database.

Suppose, for example, that the user selects the time series $A1$ in Figure 1, indicates that the segment from 5/11/1988 through 6/3/1988 is "interesting" and asks the system to determine if this is a "bottom reversal" pattern, as defined in Section 3. This discovery process can have different levels of focus of the search. In our context of temporal databases and predicates, therefore, we consider the following types of discovery of predicate-based patterns:

1. *bottom-reversal*$(IBM, 5/11/1988, 6/3/1988)$

   This type discovers if the time series for the IBM stock price exhibits the *bottom-reversal* pattern from 5/11/1988 to 6/3/1988.

2. *bottom-reversal*(IBM, t_1, t_2)

   This type discovers all times t_1 and t_2 such that the IBM stock price exhibits the *bottom-reversal* pattern on the time interval $[t_1, t_2]$.

3. *bottom-reversal*$(X, 5/11/1988, 6/3/1988)$

   This type discovers all the time series X that exhibit the *bottom-reversal* pattern between 5/11/1988 and 6/3/1988.

What would such a discovery process require? First, it requires an appropriate representation for describing segments of temporal functions, as discussed in Section 2. Second, the matching of a pattern with a time series can be either exact or approximate ("fuzzy.") As an example of the exact matching, the interest rate might exactly match the user-defined bottom reversal pattern between 5/11/1988 and 6/3/1988. However, most likely, the matching would only be approximate. In this case, we have to use some measure of "fuzziness" to see if a time series matches a pattern. In general, approximate matching can be stated as the following optimization problem. Define some distance between a pattern and a time series that depends on some "matching" function. Then find a matching function that minimizes the distance between the pattern and the time series, subject to the user-specified constraints on the matching function. We say that the pattern matches the time series if the minimal distance is "small enough."

We have already gone some way in developing and testing some techniques that appear to hold promise for tasks in this class. In particular, we have found that the signal processing technique of dynamic time warping is quite useful in detecting near matches of temporal patterns in large time series data [BC94, BC95]. We have also begun exploring the application of neural net and fuzzy set technologies to tasks of this class.

4.2 Class II Tasks

Discovery tasks of Class II involve the validation of previously asserted *rules*. Typically, rules involve multiple predicates, and therefore generally multiple relations or views in the

database. When the database is rich in temporal data, such rules frequently express correlations between occurrences of different patterns, such as that "X typically follows Y", where "typically" might mean with a certainty factor above a certain threshold. Looking again at the data shown in Figure 1, for example, a domain expert might discern certain patterns that could be expressed in the form of rules. In the econometric domain, a rule might assert that "a sharp rise in housing starts (A2) typically follows a bottom reversal in the large capitalization index (A1)." In the agricultural domain, it appears that "a sustained drop in the temperature (B3) and a sustained drop in the amount of rainfall (B1) typically occur simultaneously." Both of these proposed rules demonstrate the need to accommodate fuzziness in expressing such patterns, for instance in such terms as "typically follows" and "sustained."

The task at this level can be viewed more generally as follows. Given a rule of the form as defined in expression (1), evaluate (if it is a newly proposed rule) or re-evaluate (if the rule was previously asserted) the certainty factor c.

We have already touched above upon one of the research question which bears on this type of discovery, namely, the treatment of "fuzzy" temporal relationships such as *follows*, *follows soon after*, etc. and the treatment of "fuzzy" matches with other temporal patterns such as *bottom-reversal*. As we discussed above, this fuzziness could be treated either at the language level, with an explicit representation of the fuzziness, or left to the algorithmic level, i.e., incorporated into the pattern matching. In either case, the degree of matching of the predicates in a rule will influence the overall confidence — expressed in terms of c — in the strength of the entire rule.

In summary, classes I and II are both validation tasks, where the system is presented with a given pattern (predicate- or rule-based) and is asked to "evaluate" it with respect to its knowledge base and the underlying database. More difficult than this type of discovery is the task of generation. In the next two sections we discuss the issues involved in the generation of new predicates (Class III) and new rules (Class IV).

4.3 Class III Tasks

In order to *discover* an interesting pattern (in contrast to *validating* one), a system must know *where* to focus its search. It must also know *what* to look for. Both these pieces of information can be based on the interests of the user. For example, patterns of interest could be "bottom reversals" in interest rates over some specified time period, "periodic spiking" of rainfall in a region, "volatile yields" of coffee crops, and so on. These patterns are newly discovered predicates, which could later become part of new rules which we consider in Class IV patterns.

Let us consider a specific database on schema $stocks(Date, Stock, Volume, P/E, High, Low, Close)$, from which we want to discover "interesting" predicate-based patterns. If we are interested in finding patterns on time series, we should first specify the *space* of time series to consider and how it can be determined from the database. This space of time series can either be specified by the *user* or be discovered by the *system* as being interesting in some sense. Initially, we assume that the user specifies this space of time series. For example, he or she may be interested in patterns of daily closing prices for the small capitalization stocks traded on the NASDAQ Stock exchange between January 1990 and January 1993.

Once the space of time series is defined, the system should look for "interesting" patterns in it. Where should a system start looking for "interesting" patterns in this set of time series? Should it look for periodic spikes or drops? For shapes that seem to occur several times over the time series? What durations should it pick for these shapes? Should it look for the most frequently occurring patterns of closing prices where the pattern length is between 10 and 30 days? The search here must be based either on what is of interest to the user or "anomalous" data values, that is, values or distributions that deviate significantly from what they are "expected" to be.

There are a number of measures of interestingness that the user can specify to the discovery system, such as

- volatility, standard deviation, and change in amplitude; for example the user may want to find patterns with highest standard deviation over some time interval
- frequency of occurrence; for example, the user may want to find the most frequently occurring patterns of daily closing stock prices for small capitalization stocks
- periodicity; for example, the user may want to find the most periodic patterns in the amounts of rainfall in Rio de Janeiro

In addition, the system should have parameters for the above measures of interestingness. For example, it makes little sense to simply ask a system to look for the most frequently occurring patterns since shorter duration patterns would occur far more frequently than longer ones. Rather, what might be more interesting are the most frequently occurring patterns over periods of time ranging, say, between 10 and 30 days with the maximal standard deviation of 3.75. In effect, we must specify *constraints* which enable a system to find patterns that will turn out to be meaningful to the user.

We have begun exploring the application of genetic algorithms and neural net technologies to tasks of this class. We also plan to explore clustering, factor, and discriminant analysis techniques for generating interesting predicates.

4.4 Class IV Tasks

Discovery tasks of Class IV involve the discovery of new rules of the form $p_1 \wedge \ldots \wedge p_n \rightarrow q$, as defined in Section 3. For example, the system can discover the following rule that might be of interest to the economists:

> a bottom reversal in the large capitalization index, when accompanied by a bottom reversal in the interest rate, is typically followed shortly by a spike in the number of housing starts

This rule can be formally expressed as

$$bottom\_reversal(largecap\_index, t1, t2) \wedge bottom\_reversal(int\_rate, t3, t4) \wedge same\_time(t1, t2, t3, t4) \wedge soon\_after(t, t2) \rightarrow spike(housing\_starts, t) \quad [c]$$

where *same_time(t1,t2,t3,t4)* is a temporal correlation predicate specifying that the time intervals $[t1, t2]$ and $[t3, t4]$ occur "roughly" at the same time, and $soon\_after(t, t')$ is another temporal correlation predicate specifying that $t' < t$ and that t and t' are "close enough."

The problem of discovering new interesting temporal rules of the general form $p_1 \wedge \ldots \wedge p_n \rightarrow q$ [c] is an open-ended problem and can be focused by specifying the following types of constraints:

1. Constraints on the structure of a rule, i.e., what types of predicates p_i and q, $i = 1, \ldots, n$, can appear in the rule, and constraints on the value of n. At one extreme, we can impose no constraints on the rules to be discovered. This means that we allow n to be any integer, and p_i and q be arbitrary predicates defined in Section 3. As another extreme, we assume that $n = 1$, and that we want to discover the rules of the form $p_1 \rightarrow q$, where p_1 and q are predicate-based patterns. Clearly, we can consider rules with other types of constraints that form a spectrum of options between the two extremes discussed above.

2. Constraints on the size of the set of ordered data (such as a time series) on which rules are discovered. We can assume either that each time series comes from a predefined *set* of ordered data (e.g. the set of daily temperatures taken in the major US cities over the past 3 years) or that it is a *single* set of data (daily temperatures in New York City taken over the past 3 years).

3. Constraints on the number of patterns associated with set of ordered data. For example, we can look either at a *single* pattern on a time series or on *multiple* patterns.

Depending on the selection of these constraints, we obtain different types of pattern discovery problems. We present some of the important problems for three specific choices of sets of constraints described above.

1. Discovering *correlation* rules among patterns such that each pattern is defined on a *single* data series (the rule $p_1 \wedge \ldots \wedge p_n \rightarrow q$ is a *correlation* rule if $p_1, \ldots, p_n$ and q are predicate-based patterns). Consider the ordered data set TS_i for $i = 1, \ldots, k$ and assume that a predicate-based pattern PAT_i is associated with each TS_i. Then we want to find any interesting correlations among different patterns PAT_i for $i = 1, \ldots, k$. For example, we can try to find out if there is a correlation between *head_and_shoulders* and *decline* patterns in interest rates considered over the past 10 years.

2. Discovering *correlation* rules among *multiple* patterns on *multiple* series of data. Consider an index $I = \{1, \ldots, k\}$ and a set of data series $\mathbf{TS}_i = \{TS_{ij}\}$ for each $i \in I$. For each series TS_{ij}, there is a set of patterns $\{PAT_{ijm}\}$ associated with it. Then we want to find any interesting correlations among different patterns $PAT_{ij_im_i}$ for $i = 1, \ldots, k$, such that pattern $PAT_{ij_im_i}$ is defined on TS_{ij_i}.

3. Discovering any *arbitrary* interesting rules $p_1 \wedge \ldots \wedge p_n \rightarrow q$ without any constraints on their structure. For example, we may want to discover interesting rules involving known technical analysis patterns [LR78] for the stock prices. These rules can involve stock prices, interest rates, and various other important statistics. They can also involve various groups of stocks, such as "transportation," or "small capitalization," or "growth" stocks.

The major research issue is how to discover new rule-based patterns for each of the three problems presented above. The special case of Problem 1, when we want to discover correlations between two known patterns on a single time series, can be solved by using the solutions to the pattern discovery problems in Class I. For example, if we want to find correlations between *head-and-shoulders* and *decline* patterns in interest rates over the past 10 years, then we can determine all the cases of *head-and-shoulders* and of *decline* patterns over the past 10 years using the discovery techniques for Class I. After that, we can detect correlations among instances of these patterns. On the other extreme, Problem 3 in its general formulation requires that a system decide for itself what metrics (frequencies, standard deviations, etc) it will use for detecting anomalies, and direct the search based on these metrics.

We believe that the discovery tasks in this quadrant are inherently generate and test kinds of tasks for which intelligent hypothesis generators are essential. We have already developed some discovery techniques for the non-temporal domain [DT93] which we plan to extend to the temporal case. We have also plan to explore applications of genetic algorithms, neural nets, decision trees [Qui86], and statistical classification methods [BFOS84] to the discovery tasks in this quadrant.

5 Concluding Remarks

The task of discovering knowledge from databases is a challenging one, While considering temporal databases adds to the complexity of both the database and the knowledge structures, we believe that the payoff will be great because in many applications temporal associations, correlations, and patterns ware a rich source of domain knowledge. In this paper we have provided an overview of the characteristics of both the application areas and the knowledge structures which we are analyzing.

In the organization of our research agenda, we have found that having a simple but informative classification scheme for the units and the tasks of discovery is an extremely useful in developing a plan of attack for the many problems such a rich domain presents. It provides a way of categorizing tasks according to their level of difficulty, and it also suggests what techniques might work for discovering the different kinds of patterns.

We are currently exploring a number of techniques for handling these different types of discovery in a variety of application areas, and are developing a prototype discovery system structured as a loosely-coupled toolkit of tasks. Each task is a specialist in solving a particular subproblem of the overall goal of discovering useful knowledge.

ACKNOWLEDGMENTS

This research was supported by the NSF under grant IRI-9318773. The authors would like to acknowledge the contribution to this research of Mr. Donald Berndt, who has been an active collaborator and participant on this project since its inception.

References

[All84] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[BC94] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings KDD-94: AAAI Workshop on Knowledge Discovery in Databases*, Seattle, Washington, July 1994. AAAi.

[BC95] D. Berndt and J. Clifford. Finding patterns in time series: A dynamic programming approach. In U.M. Fayyad and G. Piatetsky-Shapiro, editors, *nowledge Discovery in Databases, volume 2.* AAI/MIT Press, Cambridge,, 1995. (to appear).

[BFOS84] L. Breiman, J. H. Friedman, R.A. Olshen, and C.J. Stone. Classification and regression trees. 1984.

[Chu92] C. K. Chui. *An Introduction to Wavelets.* Academic Press, 1992.

[DT93] V. Dhar and A. Tuzhilin. Abstract-driven pattern discovery in databases. *IEEE Transactions on Knowledge and Data Engineering*, 5(6), 1993.

[FPSM91] W.J. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus. Knowledge discovery in databases: an overview. In G. Piatetsky-Shapiro and W.J. Frawley, editors, *Knowledge Discovery in Databases.* AAAI / MIT Press, 1991.

[LR78] J. B. Little and L. Rhodes. *Understanding Wall Street.* Liberty Publishing Company, Cockeysville, Maryland, 1978.

[Qui86] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[Ull88] J. Ullman. *Principles of Database and Knowledge-Base Systems*, volume 1. Computer Science Press, 1988.