A Survey of Temporal Knowledge Discovery Paradigms and Methods

John F. Roddick, *Member*, *IEEE Computer Society*, and
Myra Spiliopoulou, *Member*, *IEEE Computer Society*

Abstract—With the increase in the size of data sets, data mining has recently become an important research topic and is receiving substantial interest from both academia and industry. At the same time, interest in temporal databases has been increasing and a growing number of both prototype and implemented systems are using an enhanced temporal understanding to explain aspects of behavior associated with the implicit time-varying nature of the universe. This paper investigates the confluence of these two areas, surveys the work to date, and explores the issues involved and the outstanding problems in temporal data mining.

Index Terms—Temporal data mining, time sequence mining, trend analysis, temporal rules, semantics of mined rules.

1 Introduction

The increase in the affordability of storage capacity, the associated growth in the volumes of data being stored and the mounting recognition in the value of temporal data (as well as the usefulness of temporal databases and temporal data modeling) has resulted in the prospect of mining temporal rules from both static and longitudinal/temporal data. Data mining can itself be viewed as the application of artificial intelligence and statistical techniques to the increasing quantities of data held in large, more or less structured data sets, such as databases [1], [2], [3], and temporal data mining is an extension of this work.

Temporal data mining is an important extension as it has the capability of mining activity rather than just states and, thus, infering relationships of contextual and temporal proximity, some of which may also indicate a cause-effect association. In particular, the accommodation of time into mining techniques provides a window into the temporal arrangement of events and, thus, an ability to suggest cause and effect that are overlooked when the temporal component is ignored or treated as a simple numerical attribute. Moreover, temporal data mining has the ability to mine the behavioral aspects of (communities of) objects as opposed to simply mining rules that describe their states at a point in time—i.e., there is the promise of understanding *why* rather than merely *what*.

Temporal mining covers a wide spectrum of paradigms for knowledge modeling and discovery. For example, consider a rule stating that, when the stock price of company A shows a steep increase, the stock price of company B shows a similar trend within the next 30 minutes. This is a rule from the domain of time series analysis—it exposes the similarities between the two time series and can be the output of a trend discovery algorithm.

As another example, consider an association rule[1] stating that potato chips and hot chili sauce are purchased together during winter. This is a temporal association rule—the static equivalent would simply associate the two products. The temporal aspect "during winter" is essential in two respects. First, the association may be rare during the rest of the year, so it may go undetected if the analysis concentrates on static association rules over the whole data set. Second, since the association holds only in winter, marketing campaigns combining the two products should also take place in winter—a joint offering during summertime might fail to bring the expected return of investment.

The discovery of frequent sequences (we use the terminology of [4]) is another domain of temporal mining. Here, correlations are discovered among events ordered on the time axis. In many applications, from car manufacturing to signal jam detection in networks, sequence mining is used to assess after which events an interesting event (e.g., an error) is most expected to occur. In web analysis, the same paradigm is used to predict and prefetch the URLs to be requested by a user on the basis of already visited pages.

Finally, consider an association between the marital status and the voting behavior of people. An observation stating that this association is declining, i.e., that its confidence is decreasing with time, is surely as important for the prediction of voting results as the association itself. Here, temporal knowledge discovery is not applied on the data but on the rules extracted from the data at various time points.

The above examples indicate the importance and usefulness of capturing and analyzing the temporal aspects of the data and of the rules discovered over the data as part of the knowledge discovery process. In this paper, we investigate the confluence of data mining and temporal semantics.

- *J. Roddick is with the School of Informatics and Engineering, Flinders University of South Australia, PO Box 2100, Adelaide 5001, South Australia. E-mail: roddick@cs.flinders.edu.au.*
- *M. Spiliopoulou is with the Leipzig Graduate School, Jahnallee 59, D-04109 Leipzig, Germany. E-mail: myra@ebusiness.hhl.de.*

Manuscript received 21 May 1999; revised 5 June 2000; accepted 22 Jan. 2001; posted to Digital Library 7 Sept. 2001.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 109934.

[1] All examples in this paper should be assumed to be fictitious.

1041-4347/02/$17.00 © 2002 IEEE

To this end, we first provide an overarching conceptual framework, into which studies in temporal knowledge discovery are organized according to the type of temporal objects to which they are applied and on the mining paradigm they use. We then survey the research contributions to each part of this framework.

Our objective is less the comparison of studies—an issue that is largely dependent on the specific application problem—than the establishment of a framework of reference and an intuitive insight into the strengths and particularities of each contribution.

In the next section, we first provide an overview of temporal semantics as seen from the viewpoint of temporal data and temporal knowledge and propose the framework for the categorization of studies on temporal data mining. The survey of research contributions and the discussion of open issues are organized according to this framework. Section 3 concentrates on the discovery of temporal association rules. In Section 4, the focus is on algorithms used to discover patterns in time series and generic temporal sequences with methods other than classification and clustering. The latter are the subject of Section 5, which discusses supervised classification and unsupervised clustering. Section 6 investigates research concerned with determining what is meant by *useful* or *interesting* mining results in the context of temporal data mining. Section 7 discusses temporal mining requirements and the environments within which temporal mining occurs. Section 8 concludes this study and presents a number of areas for future research.

2 The Semantics of Temporal Data and Temporal Knowledge

Before surveying the domain of temporal data mining, we need a framework that catagorizes the large corpus of existing literature, albeit that it is difficult to assign a unique label to each research contribution in this wide and active domain. Our framework is based on the types of temporal data being analyzed, the mining paradigm being applied, and on the goal of the knowledge discovery process.

2.1 Types of Temporal Data

The construction of temporal data sets and the development of temporal databases has enjoyed substantial interest for several years [5], [6] and a number of bibliographies of research in the field have already been published [7], [8], [9], [10], [11].[2] It is nevertheless important to note that a fully temporal database is not essential for temporal knowledge discovery and that temporal rules can also be derived from sequences of static data sets.

Hoschka and Klösgen [14], for example, discuss the potential for a limited form of temporal mining within the *Explora* system using multiple static snapshots. Temporal reasoning is added by storing separate snapshots of the rule set over time that are then compared to draw conclusions regarding the change in data over time. This technique could be applied to any nontemporal database to facilitate some temporal reasoning. However, because data are not stored within a temporal database, rules describing the change in the data over time can only be derived indirectly from changes in the stored rule set.

It should also be noted that the existence of some temporal knowledge can be used to make mining easier. For example, in [15] the existence of calendars is used to segment a pattern, thus making the problem more tractable.

Four broad categories of temporality within data can be determined:

- *Static:* No temporal context is included and none can be inferred. Occasionally, some temporal inference can be made through reference to transaction-time by referring to audit trails or transaction logs.
- *Sequences:* Ordered lists of events. This category includes ordered, but not timestamped collections of events. Many marketbasket data sets are held (or are interpreted) as sequences. While most collections are often limited to the sequence relationships *before* and *after*, this category also allows for the richer relationships described by Allen and others [16], [17], [18] such as *meets*, *overlaps*, *contemporary of*, etc.
- *Timestamped:* A timed sequence of static data taken at more or less regular intervals. Examples include census and satellite meteorological data and, in some cases, time-stamped marketbasket transactions or web-based activity.
- *Fully temporal:* Each tuple in a time-varying relation in the database may have one or more dimensions of time, such as either or both of a transaction-time or valid-time history. The multidimensional nature of time as it applies to information systems (and its associated terminology) has been explored in some depth in other publications ([5], [6], [19], [20]). Most of the research discussed in this work considers a single temporal dimension, usually transaction-time.

2.2 Architectural Frameworks for Temporal Knowledge Discovery

Data Mining was itself developed from the confluence of research in artificial intelligence (particularly machine learning), statistics, and database systems. A significant quantity of artificial intelligence work in temporal reasoning exists and this has guided the development of many of the techniques. Notable work includes that of Dean and McDermott [21], [22] who developed *inter alia*, a comprehensive system for reasoning about time, and Allen [16], Vilain [18], [23], and Freksa [17], who refined the temporal relationships used in much of the temporal data mining research. The development of various forms of temporal logic has also played a part in some mining research, particularly temporal pattern matching and sequence mining. This paper does not attempt to cover the areas of temporal logic and reasoning and they will be referred to only when appropriate.

A theoretical framework for temporal knowledge discovery was proposed in the early work of Al-Naemi [24] and a fuller discussion of some issues involved with temporal knowledge discovery appeared in [25]. In addition, within

[2] Interestingly, the most recent of these [11] provides a short list of temporal data mining references. This list is expanded and available in [12], [13].

the context of temporal relational databases, a formal method of defining temporal induced dependencies was proposed in [26].

Clifford et al., provide a classification of the types of temporal patterns and rules that can be discovered [27]. They categorize possible regularities into classes of patterns that are differentiated by pattern structure and search effort/methods. Although the development of the taxonomy was motivated primarily by their work in time series analysis (see later), the taxonomy can be applied more widely.

More recently, Fawcett and Provost introduced the concept of "Activity Monitoring" and mapped a category of problems from the domain of temporal mining to it [28]. In particular, they defined as activity monitoring the task of analyzing sequences of events in order to detect the occurrence of interesting behavior, which they term "positive activity," where an activity may be an event or a combination of events. The origins of the approach are in the domain of fraud detection, where it is important to identify as early as possible that a sequence will exhibit positive activity in the future and issue an alarm. Nevertheless, the authors show that, with an appropriate specification of the notion of positive activity, activity monitoring can be applied in other domains as well, such as in the monitoring of correlated stock prices.

2.3 A Taxonomy for Temporal Knowledge Discovery

The frameworks discussed above do not encompass the complete area of research related to knowledge discovery over temporal data. In this study, we provide a framework that covers all types of objects with temporal properties, over which knowledge discovery is applied—that is, we consider in this study both data *and* patterns on that data.

Many studies in conventional data mining distinguish two *strategic goals* for the discovery process: 1) the *description* of the characteristics of a population and 2) the *prediction* of its evolution in the future. In temporal data mining, this distinction is less appropriate because the evolution of the population is already incorporated in the temporal properties of the data being analyzed.

In our study, we distinguish two *tactical approaches*, as opposed to *strategic goals* for temporal data mining, namely, 1) prediction of the values of a population's characteristics and 2) identification of similarities among members of a population. The usefulness of this approach becomes most apparent when we observe time series data in which research related to time series analysis focuses either: 1) on the *prediction of the curve* to be followed by one time series in the future or 2) on the *discovery of similarities* among multiple time series. It should be noted, however, that the two motivations overlap because one approach to predicting a time series curve is based on observing the curve of a time series similar to it. We anticipate that, in this case, the major challenges still lie in comparing time subseries and discovering similarities among them.

In this survey, we concentrate on research related to the discovery of similarities among temporal data. The reader is referred to [29] for a collection of articles on prediction in time series analysis and for a thorough description of the issues related to the prediction-of-the-curve problem.

In the context of the discovery of similarities in temporal data, we categorize data mining research across three dimensions:

- *Datatype:* The data subject to the knowledge discovery process can be conventional scalar values, such as stock prices, or events that cannot be ordered, such as telecommunication signals. We also consider one further datatype, the one describing the mining results themselves, so that we can observe pattern evolution in time.
- *Mining paradigm:* With respect to the process employed in the discovery of similarities, we can identify different approaches. They include methods of supervised and unsupervised classification, methods for the discovery of association rules, and frequent sequences, as well as the mining languages for the specification of templates to be detected on the temporal sequences.
- *Ordering:* The data subject to the knowledge discovery process can be temporally ordered, perhaps by timestamping. Some of the research contributions utilize the temporal order placed over the data (e.g., time series analysis), while others mine the full temporal characteristics of the data (e.g., some temporal association rule routines).

In Table 1, we thus propose a taxonomy for temporal mining by considering each mining paradigm for each datatype of our feature space. We further distinguish between mining methods that are applied on sequences and those applied on unordered itemsets. Some combinations of values in this feature space are already well-known, such as "sequence mining." Others are intuitive, such as the discovery of "temporal association rules." For some concepts, we have introduced a term, such as "pattern evolution" although there is existing research in this area, no generally agreed term has yet emerged. It should be noted that we use the term "classification" both for supervised classification via training and for unsupervised classification, also known as clustering.

A special remark is necessary for the group of methods characterized as performing an "Apriori-like" discovery. In 1993, Agrawal et al. proposed the concept of "association rule discovery" together with the "Apriori" algorithm, which discovered associations between items by "mining the data" [30]. While association rules ignore ordering among the items, an Apriori variation respecting (temporal) ordering emerged as early as 1995 under the name "sequence mining" [4]. There is, therefore, justification in saying that association rules' discovery and sequence mining, while different in many respects, are based on the same mining concept. To the best of our knowledge, this mining concept does not have a name. In our taxonomy, we need a common name for this concept to which temporal association rules' discovery, sequence mining, and some further methods are subordinate. Notwithstanding the fact that many of the research contributions in these areas are *not* variations of the Apriori method, we opted for the name "Apriori-like" to stress the common historical origin.

TABLE 1
A Taxonomy of Temporal Mining Concepts

Datatype →	*Timestamped objects*		
Mining paradigm ↓	*Value*	*Event*	*Mining result*
Apriori-like discovery	· Template-based mining languages	· Sequence mining · Discovery of temporal association rules	· Pattern maintenance · Pattern evolution
Classification	· Discovery of common trends	· Sequence classification · Temporal extensions to classification	×

In the first column of this taxonomy, we consider the analysis of timestamped scalar values such as stock prices, animal populations, etc. Knowledge discovery for this type of data encompasses the analysis of time series by describing patterns (often as templates) and identifying the time series where they appear, as well as the discovery of common trends among multiple time series. Although the two paradigms are semantically close, we distinguish between them as the former uses conventional pattern matching techniques while the latter exploits classification mechanisms.

Example 1. Consider a time series describing stock prices in the stock exchange. A rule stating that "whenever the stock price for company A shows a steep increase, the stock price for company B shows a similar increase after 20 or 30 minutes" may be interesting for potential buyers of company B stocks.

This rule belongs to the first column of our taxonomy. It can be discovered by an Apriori-like algorithm taking as input a template (in this case a "steep increase") and a time window (in this case, 30 minutes) and returning all time series that contain a subseries satisfying this template. This method of discovering similarities is depicted in the upper cell of the first column.

Classification algorithms, as depicted in the lower cell of the first column, group time series showing similar trends together. Thus, the time series of companies A and B would be assigned to the same group by the classifier. The inspection of the group would reveal the correlation between their stock prices.

The second column of our taxonomy addresses the analysis of events along a specified time axis. Events are ordered in time, but their contents have no ordering and, thus, cannot be compared to identify increasing or decreasing trends.

Sequence mining is the appropriate paradigm used in this case, being successfully applied in the analysis of telecommunication signals and the discovery of web usage patterns. Temporal association rules are the result of a discovery process focusing on the temporal relationships between a collection of events. Finally, classification and clustering methodologies are applied on sequences of timestamped events to form groups composed of sequences with similar subsequences.

Example 2. In Fig. 1, the first rule states that a rainfall is preceded by a drop in atmospheric pressure in 60 percent of the cases. This rule is actually the sequence composed of the events `Pressure-drop` and `Rainfall` in the order imposed by the temporal dimension. Sequence miners are developed to process sequences of events and identify frequently occuring subsequences in them.

The same rule would be discovered by a classification algorithm grouping sequences of events (in this case, meteorological phenomena). Sequences containing the two events `Pressure-drop` and `Rainfall` in the same order would be placed in the same group. The inspection of the group would reveal the correlation between the two events.

Example 3. In Fig. 1, the fourth and the sixth rule depict associations across the temporal dimension. Conventional mechanisms for association rule discovery would identify the correlation between the events, but overlook the time parameter, which is important in both cases. The goal of temporal association rule discovery is to discover such rules.

The fifth rule in Fig. 1 is a sophisticated variation of the first rule type, which was shown to be a sequence of events. The fifth rule states that a sequence of events is more frequent during specific time intervals than it is other times.

The third column in Table 1 addresses the fact that the mining results have themselves temporal properties. Studies on the observation of changes in discovered patterns are assigned to this column under the term "pattern evolution." Research on the maintenance and updating of rules, whenever the data set is modified, are also related to this subject. These methods focus mostly on the mining results of Apriori-like methods.

Example 4. The third rule of Fig. 1 states that the correlation between marital status and voting behavior is fading. Another way of expressing the same fact is by observing the time series of the confidence of the association rule `marital-status → voting-behavior`; in this time series, there is a falling trend. In this case, instead of

1. A drop in atmospheric pressure precedes rainfall in 60% of cases.
2. The sequence `Committee` → `Board` → `Council` occurs approximately every month.
3. Marital status is becoming less of a determinant of voting behaviour.
4. Beachside flooding only occurs during spring high tides.
5. There is a higher incidence of earthquakes during and soon after periods of higher atmospheric pressure.
6. Some patients tend to develop reactions after two months with this combination of drugs.
7. The introduction of the Euro caused a different pattern of buying behaviour in offshore markets.

Fig. 1. Examples of temporal rules.

building time series of data and observing their evolution with time, we observe the evolution of an association rule derived from the data.

This rule deserves particular attention for the additional reason that it does not state a correlation between a specific marital status (e.g., bachelor or married) and a vote (e.g., conservative or liberal), but an association between the two attributes in general. The generalization of associations among values into associations among attributes leads to the discovery of functional dependencies. Thus, our framework not only covers temporal association rules but also temporal functional dependencies.

The second rule is similar to the third, although it reflects the evolution of associations instead of the ordering of nontemporal associations.

We consider the third column of the above taxonomy as the one containing some of the most challenging issues. We attempt, therefore, to further model the subject of knowledge discovery over the mining results so that other mining paradigms can be exploited in this context.

2.4 A Tour Through the Studies in Temporal Knowledge Discovery

Fig. 2 reflects the organization of the following sections in which we discuss mining algorithms that incorporate temporal concepts. According to Table 1, we distinguish among three types of timestamped objects. We consider the paradigms of Apriori-like discovery and classification (which includes clustering), thereby distinguishing between methods reflecting ordering, i.e., analyzing sequences, and those that ignore ordering. In the figure, shaded areas represent domains that are either beyond the scope of our study or for which we know of no relevant research.

3 Apriori-Like Discovery of Association Rules

This Section takes a closer look at Apriori-like mechanisms for rule discovery and the manner in which they can be applied over temporal data. The fundamental paradigm is that of association rules' discovery in which correlations between objects occurring in the same transaction are identified without taking any ordering of the objects into account.

3.1 Temporal Association Rules

Association rules typically find correlations between items in transaction data sets that record activity (such as purchases) on multiple items as part of a single transaction [30]. Retail activities, for example, are stored in large transaction databases. Associations found among them can assist organizational decision-makers in planning marketing strategies.

Importantly, each basket of items is normally treated individually with no record of the associated customer or client who purchased these goods (although timestamps may be recorded). However, in cases where client histories exist, temporal patterns on purchasing or other behavior over time can be discovered and used in strategic planning [31], [32].[3]

As an example of this, consider the following scenario: While a nontemporal association rule might suggest that the presence of mature stands of River Red Gum Eucalypts is associated with the presence of the endangered Red-Tailed Black Cockatoo, a temporal association rule may indicate that the presence of the Cockatoo usually occurs *some time after* the Eucalypts stand has reached maturity. This may indicate that a recovery plan for the endangered Cockatoo would involve maintaining what might otherwise be considered ageing stands of River Red Gums.

In another scenario, the existence of stands of wattle (Australian Acacia) may *finish* as Eucalypts start to mature, indicating that there may be some biological connection between the (short-lived) wattles and the (longer-lived) Eucalypts. This temporal information is a natural extension to existing association rule semantics and can form valuable institutional knowledge.

It should be noted that the presence of a temporal association rule may suggest a number of interpretations:

- The earlier event plays some role in causing the later event,
- There is a third (set of) events that cause both other events,
- The confluence of events is coincidental.

The first interpretation is associated with the concept of *causal rule* [33], [34], [35]. A "causal rule" describes a relationship in which changes in one part of the modeled

[3] Note that, even if client details are not recorded, second order mining over association rules may yield useful temporal rules. This latter discovery category is covered in Section 3.2.3.

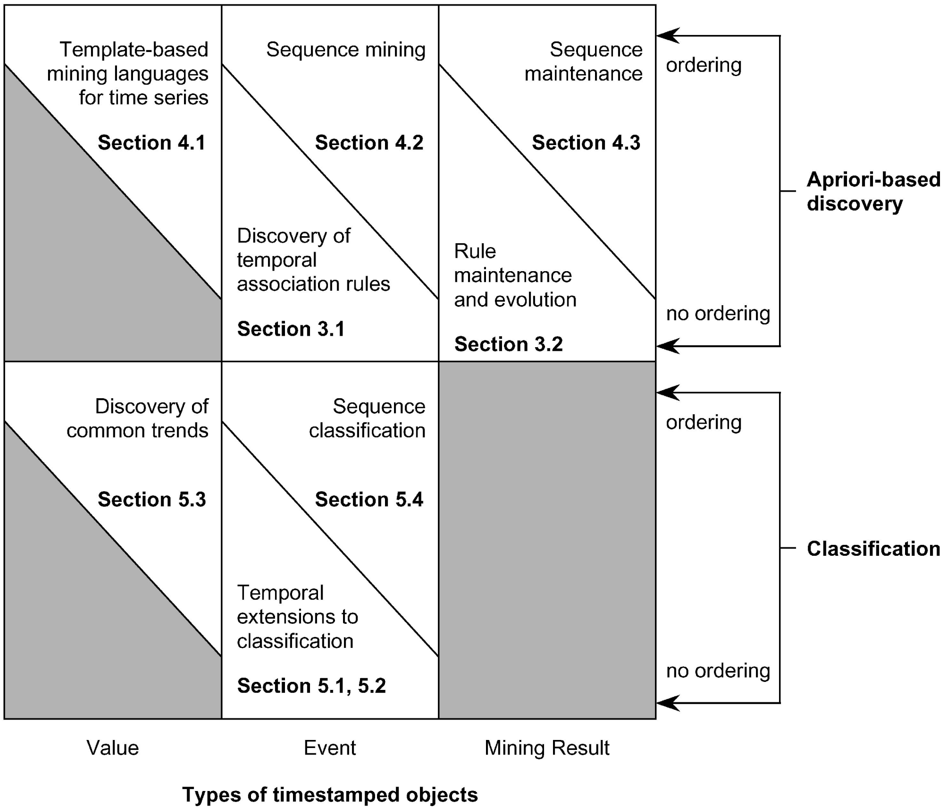

Fig. 2. Organization of the studies according to the taxonomy of Table 1.

reality cause subsequent changes in other parts of the domain. Causal rules are common targets of scientific investigation within the medical domain, where the search for factors that may cause or aggravate particular medical conditions is a fundamental objective. In this domain, KDD tools can be applied at a preliminary stage, namely, to discover associations that can be observed as candidate causal rules. The tests for causality follow in a subsequent stage, involving expert guidance and extensive statistical tests.

Temporal association rules are particularly appropriate as candidates for causal rules' analysis in temporally adorned medical data, such as in the histories of patients' medical visits. Patients are associated with both static properties, such as gender, and temporal properties, such as age or current medical treatments, any or all of which may be taken into account during mining. Rule 7 in Fig. 1 is an example of a causal rule from another domain.

While the concept of association rule discovery is the same for temporal and nontemporal rules, algorithms designed for conventional rules cannot be directly applied to extract temporal rules. The reason is that classical association rules have no notion of order, while time implies an ordering. This ordering affects the statistical properties of the data and the semantics of the rules being extracted from them. For example, assume that the association data between holders of *Insurance policy A* and *Investment portfolio B* are as found in Table 2.

From those facts it can be seen that 80 percent of the people with *Insurance policy A* progress to *Investment portfolio B* afterward. However, if ordering is ignored, we see that only 36 percent of the holders of A or B hold both. Thus, if the miner reduces the search space by skipping rules with a probability less than, say, 50 percent, the desirable rule will be ignored ([25]).

This problem was recognized early and specialized miners for the discovery of *sequential patterns* have emerged (Section 4).

3.2 Evolution and Maintenance of Static Association Rules

In the previous section, we discussed temporal aspects by observing time as a property of the data. Time is also a property of the mining results, which are discovered for a data set *as it was at some point of time*. Afterward, data changes take place and their statistical impact on the mining results must be traced.

TABLE 2
Investment Policy Example

Total number of customers:	10,000
Holders of A:	1,000
Holders of B:	2,000
Customers that started B after A:	800
Customers that started B before A:	0

3.2.1 Life Expectancy and Evolution of Association Rules

Changes on association rules caused by updates in the underlying data are referred to in the literature as a problem of "rule maintenance."

The problem of updating knowledge derived from volatile data is not peculiar to data mining. Pechoucek et al., investigate the concepts of "weak update" and "strong update" of knowledge in the context of 1) inductive logic programming and of 2) explanation, based generalization [36]. Weak updates correspond to an incremental modification of the knowledge as new data become available. In a strong update, the whole data set is taken into account to replace the old knowledge base with a new one. For inductive logic programming, the authors derive formulae computing the overhead of each type of update and of the performance degradation of the inductive logic program.

Pechoucek et al. start with the assumption that a strong knowledge update results in a knowledge base of higher quality than a corresponding series of weak updates. However, data updates have a more fundamental impact on the results of data mining. In particular, mining rules are used for strategic decisions, which affect the content of future transactions. Thus, transactions recorded after taking actions based upon the mining results have different statistical properties than those from which the mining results were originally drawn. Hence, a strong knowledge update does not necessarily lead to a better knowledge base than a corresponding series of weak updates.

In the context of data mining, research concentrates on "weak knowledge updates" according to the terminology in [36]. A main challenge is seen in reducing the overhead of rediscovering association rules in the presence of data updates. To this purpose, the support of large and small itemsets in the set of deleted transactions and of large itemsets in the set of inserted transactions should be recalculated, while accesses to the original data set must be minimized.

Cheung et al. propose the FUP algorithm for the updating of association rules in a data warehouse, where only insertions are permitted [37], [38]. For mining in an operational database, where deletions may also take place, the faster version, FUP_2, presented in [39] can be used instead. More recently, Ayan et al. proposed the UWEP algorithm that removes itemsets that cease to be large by reading the newly inserted data once and the original data set at most once [40].

Rainsford uses temporal information to focus on the mining activities on the most relevant data subset [25], [41]. In particular, association rule discovery is not performed over the whole database but only over the transactions recorded within a user-specified time window. The rule discovery mechanism verifies whether already discovered rules remain or became *strong* (i.e., frequent) within the time window, pruning out outdated rules that do not show enough statistical support.

3.2.2 Unexpectedly Evolving Association Rules

Chakrabarti et al. [42] investigate the discovery of unexpected patterns in market basket analysis by observing the variation of the correlation of the purchases of items over time. Their emphasis is on the evolution of the purchases of each item over time, as in time series analysis. The study stands between association rules' discovery, where the item distributions in the data set are assumed stationary, and time series analysis, where the variations in the distributions are captured by segmenting the time series into windows of some given fixed size. In conventional association rule discovery, one might miss correlations between purchases of items unless the time interval between observations is sufficiently small. On the other hand, variations in the frequencies of purchases of correlated items are not necessarily regular and cannot be captured by segments of fixed size, as commonly assumed in time series analysis.

Chakrabarti et al. thus propose the automatic segmentation of a time series (in this case, a sequence of baskets on the time axis) by observing it as a compression problem. A model is sought that allows the representation of (a part of) the sequence with minimum bits. Thus, segment change occurs whenever the chosen model is no longer optimal. This representation is based on a measure of surprise, which is modeled as the expected versus observed probability of coexistence between two items or item groups. Hence, the correlation among items in a group is found interesting if the underlying time series consists of multiple segments where this correlation varies. The rationale behind this approach is that nonvarying correlations are already known and, thus, of lesser interest.

This approach indirectly addresses the problem of how temporally proximate the records in a data set should be in order to apply association rule discovery on them, without the results being blurred by variations of the correlations over time. A comparison of this technique with simple association rule discovery could demonstrate the qualitative improvements achieved when taking time into account. Moreover, information on the structure and length of the automatically discovered segments, which are skipped in the experiments, could also shed light on any delay between introducing a market policy and observing its results.

In a simpler setting, Chen and Petrounias also investigate the temporal aspects of association rules, namely, the validity time interval of a rule discovered in multiple mining sessions [43]. They propose an algorithm that discovers the longest time intervals during which a rule is valid, assuming that rule discovery has been performed with the same support and confidence threshold settings in all mining sessions. Moreover, they anticipate the importance of identifying rules that are valid periodically and propose a mechanism for discovering the longest validity periods of a given rule.

3.2.3 Higher Order Mining—Data Mining over Mining Results

The mining of previously mined rules (or *higher order* knowledge discovery) is an area which has received little attention and yet holds the promise of reducing the overhead of data mining. Of particular interest to us in this paper is the fact that sequences of rules may be ordered or time-stamped in some manner, thus making themselves applicable to some forms of temporal mining.

The rationale behind the mining of rules is twofold. First, the knowledge discovery process is applied on small sets of rules instead of huge amounts of data. Second, it offers a different sort of mining result—one that is arguably closer to the forms of knowledge that might be considered interesting. For example, *Medium Income is associated with Caravan Ownership* is arguably of less use than *Medium Income is becoming more associated with Caravan Ownership*.

This issue is addressed by Abraham and the authors in [44], [45], [46] and in the work of Hoschka and Klösgen [14]. In the former set, previously mined rules found by an association rule algorithm are mined to discover changes in the rules. From that, changes in rule structure, such as a change in support or confidence (which can be interpreted as *X is becoming more or less of a determinant in the prediction of Y*), or changes to a rule to add an extra antecedent term (which might be interpreted as *other factors are starting to also influence the association*) can be drawn.

Care needs to be taken in the interpretation of results from data mining in general but particularly from these higher order rule mining exercises. Specifically, higher order rule mining is only able to directly discuss changes in the rule sets themselves and, by indirect reference, to changes in the data.

4 Template-Based Mining for Sequences

Timestamped data can be scalar values, such as stock prices, or events, such as telecommunication signals. Although the difference in the datatype seems secondary at first glance, time-stamped scalar values of an ordinal domain (such as stock prices and populations) form curves, so-called "time series," and reveal trends. This observation leads to a large potential for analytical studies of time series which is not transferrable to generic time sequences. For the latter, no trends can be defined and, in general, only pattern matching techniques can be applied.

4.1 Describing and Discovering Common Trends in Time Series

Trend discovery is applied by comparing time series of continuous data and searching for similar shapes, according to some domain-specific notion of similarity. This type of pattern discovery is used when studying the evolution of stock prices, related populations, etc. The rationale is again related to prediction—if one time series shows the same trend as another but with a known time delay, observing the trend of the latter allows assessments about the future behavior of the former.

Before attempting to discover similarities among time series, the time series must first become comparable. This can be non trivial as the time series may differ in, among other things, *scale*. Searching for patterns at different resolutions will yield different results and the choice of resolution is largely domain dependent. Solutions to this problem are based on time warping techniques ([47], [48], [49], [50]) and on dynamic Fourier transforms, whereby only the few first coefficients of each time series are retained and used for comparison ([51]).

A second problem associated with pattern discovery from time-series data concerns proximity. The proximity of events in time commonly determines if any significant relationship between them can be inferred. For example, a decrease in the population of wolves in a forest can be related to a decrease in the population of hares in the previous year, but is unlikely to be relevant to a decrease of the hare population 30 years earlier. Many application domains may involve delayed reactions and, therefore, determining an appropriate proximity is essential. This problem is commonly tackled by examining only data laying within a time "window," as explained below.

In [52], the problem of discovering trend similarities in time series is decomposed into three subproblems: 1) finding all pairs of gap-free subsequences of minimal length that are similar, 2) stitching those subsequences into longer similar sequences that probably contain gaps, and 3) selecting similar sequences of maximal length.

Two sequences are defined to be similar if *they have sufficient nonoverlapping time-ordered pairs of similar subsequences*. For gap-free subsequences, the notion of similarity is somewhat different—two subsequences are similar if *one lies within an envelope of a specified width around the other, after ignoring outliers*. Transformations on amplitude are allowed to yield two subsequences comparable.

For each part of the decomposed problem an algorithm is proposed [52]. First, to find similar subsequences of length ω (called *windows*), a self-join is applied on the data set. An R-tree index is employed to speed up the search and a dedicated self-join algorithm is proposed that exploits the fact that the data set is joined with itself. Next, the subproblem of stitching the discovered subsequences together is reformulated as that of finding the longest path in an acyclic graph. The vertices of this graph are pairs of matching windows, while arcs connect consecutive pairs that satisfy certain constraints. The last subproblem is again modeled as longest path discovery in a graph. The vertices of this graph are subsequence matches, but arcs are drawn only for matches on nonoverlapping subsequences aligned on the time axis. In the experiments performed using this mining mechanism, similarities among US mutual funds have been identified.

While algorithms applying Fourier transforms and time warping techniques implicitly determine the scale at which the pattern similarity search should be performed, algorithms adhering to the problem decomposition proposed in [52] face the challenge of appropriately specifying the window size ω and the minimum length threshold used to compare sequences. Large windows imply a coarser (and, thus, a faster) search, at the risk of overlooking similar time series with small dissimilarities. A small window size implies a more detailed search with many more hits in the first step of the mining process. The minimum length threshold is actually an interestingness measure—it specifies how long the similar part of two time series should be, at a minimum, to yield them interesting.

One motivation for similarity discovery in time series is the detection of common trends. These trends can often be reflected by specifying "shapes" of interest, e.g., steep peaks, upward or downward moves, etc. Thus, pattern discovery can be driven by a template-based mining language in which the analyst specifies the shapes that should be looked for. Agrawal et al. defined a shape definition language (SDL) to describe patterns or shapes occurring in historical data [53]. The underlying algorithm compares every two consecutive values in a time series and decides the movement direction in the interval between the values. On the basis of SDL, Agrawal and Srikant defined a query language for the specification of time series patterns and trends [4]. One limitation of such a query language is that linear patterns are most intuitively described visually, typically with textual descriptions involving the use of informal language. Agrawal and Srikant allow the user to create their own language with complex patterns being defined in terms of primitives such as "up" or "down."

In [54], shapes in time series are similarly captured by an arbitrary gradient alphabet for the description of movement directions. However, instead of assessing the direction among consecutive values, the algorithm discovers subseries conforming to the desired trend, which is expressed as a sequence of symbols from the alphabet. To compare subseries, a series' length unit is used, namely, a user-supplied value that must be applicable to all series under consideration.

Finally, in [55], an approach is adopted in which the curves are segmented and are then analyzed using probabilistic interestingness criteria. Effectively, two curves are considered similar if they concur with, or can be easily deformed to, an ideal prototype. The critical component is the rules for deformation that allow some some elasticity in the time or amplitude.

4.2 Sequence Mining

Sequence miners discover correlations among the events in sequences. One example of this domain concerns drug misuse. Drug misuse can occur unwittingly, when a patient is prescribed two or more interacting drugs within a given time period of each other. Drugs that interact undesirably are recorded along with the time frame as a pattern that can be located within patient records. Wade et al. [56] describe a set-based approach to the detection of temporal patterns in patient drug usage data. Rules that describe such instances of drug misuse are then successfully induced based on medical records.

One particularity of this type of pattern discovery in comparison with trend discovery is the lack of *shapes* (Section 4.1) since the impact of a series of events cannot be shaped. This implies a dramatic increase of the search space of patterns. Constraints of statistical nature are used to reduce this space, as discussed below. The discovery of patterns satisfying statistical constraints has been addressed in several publications. We concentrate on studies focusing on events ordered in time explicitly. Pattern matching algorithms designed for other application domains, such as GIS and protein data analysis, are not covered here. For an overview of pattern discovery on biomolecular data, the reader is referred to [57] and approximate matching algorithms are discussed in [58].

In the pioneering work of Agrawal and Srikant [4], the problem of sequence mining is modeled as follows: Given a collection of transactions ordered in time where each transaction contains a set of items, the goal is to discover sequences of maximal length with support above a given threshold. A *sequence* is an ordered list of elements, an *element* being a set of items appearing together in a transaction. Elements need not be adjacent in time, but their ordering in a sequence may not violate the time ordering of the supporting transactions. The rationale behind frequent sequences lies in detecting precedence relationships and ordered associations that make themselves statistically remarkable.

The mechanism proposed to achieve this goal relies on the principles of association rule discovery [30]. All frequent items are discovered using the Apriori algorithm proposed in [30]. Following that, the collection of transactions is filtered to remove all nonfrequent items/elements. At the end of this step, each transaction consists of as many entries as there are frequent elements it contains, thus allowing repetitions of items. Following this, frequent sequences of length k are built from frequent sequences of length $k - 1$ by applying a self-join operation to the latter set and comparing the resulting sequences with the data set of modified transactions to select only those that are frequent. Finally, nonmaximal frequent sequences are removed from the result.

Two variations are proposed in [4] to enhance this mechanism, based on postponing the construction of some intermediate results and on pruning nonmaximal sequences as early as possible. Note that this pruning is undesirable in many applications since the support of a nonmaximal sequence is typically higher than that of a maximal sequence containing it. Thus, in the continuation of this research in [59], the base algorithm is used as a reference for comparisons to a more innovative and efficient technique. The *GSP* algorithm discovers frequent sequences, allowing for time constraints among the sequence elements. Moreover, it supports the notion of a *sliding window*, i.e., a time interval within which items are observed as belonging to the same transaction, even if they originate from different transactions. GSP outperforms the original algorithm of [4] by intelligently reducing the number of candidates considered in each step, speeding up the process of computing the support of each sequence and exploiting the time constraints to reduce the search space. Finally, an extension of the algorithm is described which can discover frequent sequences containing a mixture of items and *generalizations* of items, according to a semantic taxonomy.

Research on sequence mining at the University of Helsinki has been oriented toward the discovery of *episodes* that occur frequently within sequences [60]; an extension of the miner based on temporal logic appeared in [61]. An *episode* is formally a conjunction of events bound to given variables and satisfying unary and binary predicates declared for those variables. Mannila and Toivonen distinguish between serial and parallel episodes, as well as between simple episodes (those containing no binary predicates) and nonsimple episodes. They propose an algorithm based on the iterative construction of *simple* frequent episodes from simple frequent subepisodes. Although the existence of binary predicates seems to be permitted in the algorithm, their computation is an extension based on exhaustive search.

The results of the miner of [60] are episode rules of the form $P[V] \Rightarrow Q[W]$, where V, W are time intervals. If those intervals can be given in advance, the algorithm is efficient. Otherwise, the overhead is large because all possible intervals must be considered. This procedure is enhanced by predetermining and enumerating the time intervals instead of generating them on the basis of a moving time window. This enhancement has the drawback that the quality of the discovered episodes is affected by the way the expert defines the intervals in the problem domain.

Bettini et al. propose a complete framework for the discovery of frequent time sequences, placing particular emphasis on the support of temporal constraints on multiple time granularities [62]. They introduce the notion of "event structure," which is essentially a template guiding the mining process. It is comprised of temporal constraints, expressed as directed acyclic graphs. The mining process is then modeled as a pattern matching process performed by a "timed finite automaton," i.e., a finite automaton equipped with clocks for the different time granularities appearing in the data set.

The above studies concentrate on the discovery of *frequent* sequences. However, for some application domains, finding frequent occurrences of a sequence is an inappropriate selection criterion; in some cases, the more significant (i.e., useful) sequences are the unusual. Such an application domain is that of error discovery. If a defective component C always causes a malfunction A, there is a strong causal relationship between the two, which is expressed as a high confidence in the probability of malfunction A occurring sometime after installing this component. However, errors are rare events and the statistical support of this sequence CA is low however, because malfunctions are the exception, not the rule. Hence, if we restrict the search space to frequent sequences, CA will be rejected. If we reduce the threshold above which a sequence is considered frequent, it will be practically impossible to inspect the result and distinguish CA from any trivial sequence.

The *PlanMine* algorithm proposed by Zaki et al. deals with some of these particularities in the context of plan failure prediction [63]. The application domain is that of emergency plan simulations. Data mining is used to trace the events that cause a plan to fail. In other domains dealing with failure prediction, the mining objective is the discovery of rare events (see, e.g., [64] below). PlanMine takes a different approach. It attempts to identify *frequent* events that always precede plan failures *by* filtering out frequent but uninteresting events, i.e., events not preceding a plan failure. When frequent and uninteresting events are removed, the remaining events become more dominant in the data set.

PlanMine applies the SPADE algorithm [65] to identify all frequent sequences. SPADE produces an initial set of sequence rules which are pruned in several steps. In a first phase, the data set of plans is split into good and bad (i.e., failed) plans whereby successful events are completely removed from the latter subset on the grounds that only a failed event may cause a plan failure. The second pruning phase involves the removal of *normative patterns*. These are sequences that are frequent in the data set of bad plans *and* have high support in the data set of good plans. Next, *redundant patterns* are eliminated. A sequence is redundant if it contains a subsequence having the same support value as itself for both data sets of good and bad plans. Finally, *dominated patterns* are removed. A pattern is dominated if it contains a subsequence with lower support in good plans and higher support in bad plans than the pattern itself. In the experiments, the pruning steps of PlanMine are shown to effectively reduce the number of frequent sequences to an interesting and highly predictive subset.

The application domain addressed in [63] is indicative of the problems encountered when data mining is restricted in the discovery of *frequent* sequences. A suite of postmining steps, as undertaken by PlanMine, is necessary to reduce the initial result into a useful set of rules. A more robust and generic approach would be the incorporation of more general constraints into the mining process. This approach is adopted by Weiss and Hirsh in their mining algorithm for the discovery of rare events [64]; this algorithm adheres to the classification paradigm and is, therefore, discussed in Section 5.4.

In [66], [67], a fundamentally different approach is introduced. Instead of discovering frequent sequences in an iterative manner, the sequence miner *WUM* builds patterns satisfying arbitrary constraints. In [67], a mining language, *MINT*, is discussed in which the user can specify the appropriate constraints for the application. Differently than in the model of [62], the emphasis of the template-based language is not on temporal but on structural and statistical constraints. In MINT, frequent sequence discovery corresponds to the specification of a lower-bound support threshold. In exactly the same way, the user can specify an upperbound support threshold to restrict the search space to that of rare sequences. Absolute constraints on statistical support correspond to first-order predicates. Second-order predicates are also supported to allow constraints on the confidence among the elements of a sequence.

The mining algorithm behind MINT is described in [66]. The miner is not applied to the original log of transactions, but on a condensed disk-resident tree. This tree is built incrementally by a back-end service that extracts sequences from the original log and adds them on the tree, merging common sequence prefixes. The number of merged prefixes corresponding to each tree node is retained in the node's statistics and is used to compute support values during mining. The miner generates candidate sequences while traversing the tree, using heuristics to reject subsequences as soon as possible. By processing a relatively small preaggregated data structure, the execution overhead is reduced without large space tradeoffs.

The mining language of WUM is intended to support user-defined *interestingness* criteria. The theoretical problem of modeling *interestingness* for temporal mining is addressed in [68] and discussed in Section 6.

Finally, the sequence miner MiDAS [69] has also been designed especially for the discovery of navigation patterns in the web. Similarly to WUM, MiDAS is also equipped with a template-based mining language in which sophisticated statistic and structural constraints can be expressed. The two miners differ in the way they model navigation patterns though. For MiDAS, a navigation pattern is a frequent temporal sequence discovered by a mining query, while WUM regards as a navigation pattern, a group of nonmergeable sequences that together satisfy the mining query.

5 Classification of Temporal Data

Classification of temporal data can be studied from many viewpoints. First, we discuss ways of generalizing classification algorithms intended for nontemporal data to deal with temporal information. In this context, we consider mechanisms for attribute induction in which concept hierarchies over temporal data can be incorporated. Similarly, the discovery of temporal dependencies generalizes the discovery of static functional dependencies.

Next, we discuss classification techniques for time series analysis, followed by classification for general temporal sequences. While there is a large corpus of work on classifiers for time series, classification is less often applied on general temporal sequences. A discussion of related work concentrating on nontemporal sequences, such as the analysis of genomic sequences or proteins, are beyond the scope of this study.

5.1 Incorporating Temporal Concepts to Conventional Classification Algorithms

The classification and characterisation rule discovery algorithms, discussed in a series of papers by Han and others [70], [71], [72], can be extended to accommodate time in a number of ways. Minimally, time can simply be provided as a concept hierarchy, such as:

$$\text{Day} \rightarrow \text{Week} \rightarrow \text{Quarter} \rightarrow \text{Financial Year}.$$

Algorithms deriving concepts can use one of those concept hierarchies to generalize data, estimate their statistical support in the sample, and try to find trends in them. However, time can be generalized in many ways. An alternative concept hierarchy could be:

$$\text{Day} \rightarrow \text{Month} \rightarrow \text{Calendar Year}.$$

Both hierarchies are equally valid for the generalization process. However, the statistics are computed on different groups of the same data and yield different results.

Accommodating temporal semantics to the algorithm would yield better results. This approach is taken in [73], where an attribute-oriented induction algorithm is customized to incorporate temporal intervals and perform generalizations on them.

Temporal semantics that can be modeled in the concept hierarchies may include:

- *Multiple Interval and Event Semantics*: Most concept ascension techniques generalize intervals to ones of a higher granularity but have some difficulties generalizing according to multiple hierarchies. For instance, given one hierarchy that generalizes days of the week and another that generalizes months, classifying episodes that occur *during weekends in the summer months* is difficult for some routines at present as it requires the same attribute to be generalized according to multiple concepts. This problem is addressed by Li et al. [74] who develop an algorithm that first generates candidate patterns and then matches them against user-supplied granularities.
- *Relative Time*: The concepts of *before* and *after* in sequences are addressed in sequence analysis (see Section 4.1). Richer Allen-style interval relationships, such as *during*, *overlaps*, *contemporary of*, etc., are more problematic. Incorporating them into a mining algorithm remains an open problem.
- *Linguistic Reference to Temporal Data*: While a static problem also, there are particular problems in the manner in which we linguistically refer to time. *She was in Adelaide on Monday* and *She arrived in Adelaide on Monday* have different temporal semantics. The former refers to an interval, the latter to a event during an interval. This issue is also strongly associated with problems of varying granularity, discussed in more detail in [75] and elsewhere.
- *Multidimensional Semantics*: Some temporal systems hold data across multiple time lines. Research to date has exclusively assumed that all recorded events are recorded according to the same time dimension and with the same clock.

5.2 Inducing Temporal Dependencies

Attribute induction, as discussed in [70], [71], can also be applied on temporal data. Temporal induced dependencies, introduced in [26], are apparent (inductive) relationships between attributes that may only be valid at particular times or may specify a temporal relationship between parts of the functional dependency. For example, given the relation in Fig. 3, there is a temporal induced dependency of

$$\begin{aligned} &Leave.(Event, Employee)[Sick, Mary] - Contains \rightarrow^{-} \\ &\qquad Leave.(Event, Employee)[Sick, Bob]. \end{aligned} \tag{1}$$

More simply, whenever Bob is off sick, Mary is also off sick. Such dependencies are rarely used at present, but they represent one of the ways in which discovered rules may be fed back into the database system from which they were found, this time as constraints.

5.3 Classification of Time Series

In Section 4.1, we have seen that similarities among time series can be discovered by specifying the shapes of interest by means of templates and then finding all series containing those shapes. For this process, the analyst should predetermine the shapes that may be of interest. In contrast, classification techniques group series containing similar patterns together automatically.

The issues of scaling and proximity mentioned in Section 4.1 must also be resolved when classification is applied. In fact, some approaches to the discovery of classes of similar time series also incorporate mechanisms that

Leave

Event	Employee	Start-Date	End-Date
Sick	Bob	11-Feb	11-Feb
Sick	Bob	13-Mar	15-Mar
Sick	Bob	1-Jan	3-Jan
Annual	Bob	1-May	3-May
Sick	Emma	11-Jan	12-Jan
Annual	Emma	13-May	23-May
Sick	Mary	10-Feb	13-Feb
Sick	Mary	12-Mar	15-Mar
Sick	Mary	17-Feb	17-Feb
Sick	Mary	1-Jan	5-Jan
Sick	Tom	1-Apr	1-Apr
Sick	Tom	6-Mar	6-Mar

Fig. 3. Example of temporal leave relation.

yield series obtained at different resolutions comparable [47], [48]. Space precludes a discussion of all studies on time series classification in this paper. We thus discuss a selection of recent works.

In [76], Weigend et al. propose the usage of a clustering algorithm for the analysis of financial data. The problem addressed concerned the impact of trade-specific and market-specific features on trading styles in the T-bond futures market. Most of those variables, such as market volume and volatility at opening of trade, have a time dimension. The characteristics to be assessed by the analysis concerned the values of trade profit and time until expiration. The approach of choice was clustering with the AutoClass miner [77].

A clustering algorithm forms clusters in the space defined by the values of the given features and assigns "similar" data items (in this case, time series for trades) into the same cluster. Comparing the values of the same feature across time series poses a problem however—the values of the same feature in two time series are recorded at different time points. Practically, this implies that no two trades will have the same market volume at opening, except in the unlikely case that they have opened at exactly the same time point. More severely, the feature values are only comparable for the same timepoint.

Weigend et al. alleviate this problem by exponentially smoothing the values of the features within an appropriate time interval. The appropriate selection of this time interval is critical. Volatility and short term market volume at opening are smoothed within a 30 minute interval, while a 3,000 minute interval is used for long term market volume at opening. Obviously, this smoothing procedure is facilitated by the fact that a common baseline for the time intervals can be chosen. This would not be the case if, for example, some time series were recorded in a minute period while others were recorded every three hours. The analysis produced a number of clusters with their distinguishing features differing in mean profits, trading time until expiration, and trading length. This indicates that at least some of the clusters underlie different trading styles. Those styles can be traced from the feature values shared by the members of each cluster.

The work of Weigend et al. also reflects the difficulties in applying a mining algorithm on time-based data. Reasonable definitions of time intervals and preparatory smoothing of values are necessary to obtain comparable values. For the comparison of trends with different recording periods, this problem becomes more acute.

In the work of Oates [78], a clustering technique is applied on a preclassified multivariate time series. The goal is to discover patterns that are both shared among multiple time series and are significantly more/less frequent among the series of a class than among the series belonging to the complement of this class. To achieve this goal, time series are clustered using dynamic time warping as a similarity measure. The clusters' centroids are the patterns of interest. The frequency of each pattern inside and outside each class is computed, thus identifying patterns whose presence or absence characterizes a class.

Das et al. [80] combine clustering and rule induction for the discovery of similar patterns within one or among multiple time series and then apply a variation of the J-measure proposed by Smyth and Goodman [79] to rank the discovered rules by "informativeness". In particular, clustering is applied to discretize the time series. More precisely, a sliding window is moved over the series, producing subsequences that are then grouped by an arbitrary similarity measure. The cluster centroids are mapped into letters of a symbolic alphabet and each subsequence is replaced by the letter representing the centroid of the cluster it belongs to. Thus, time series are transformed into classic sequences and rule discovery or sequence mining [30], [60] can be applied on them.

5.4 Classification of Sequences of Events

The studies mentioned thus far concentrate on the classification of time series. For the classification of general temporal sequences, Zaki et al. propose *FeatureMine*, a feature extraction mechanism that serves as a preprocessor to a classification algorithm [81]. Its goal is to reduce the number of potentially useful features to be considered in the classification phase whereby a "feature" in this context is a sequence of items or itemsets.

FeatureMine uses three pruning criteria: 1) Features ought to be frequent, 2) they should be distinctive in at least one class, and 3) feature sets should not contain redundant features. The last heuristic gives rise to two pruning rules, one stating that no feature showing an accuracy of 100 percent is specialized further and one stating that if two features are correlated so that one always implies the other, then the latter is removed. In this context, a specialization of a feature is any set containing this feature and additional ones.

For the efficient traversal of the data collection, *FeatureMine* applies an algorithm across the guidelines of the *PlanMine* sequence miner [63] described in Section 4.2.

Weiss and Hirsh propose a supervised learning technique to predict rare events in sequences [64]. Their machine learning system, *"timeweaver"*, uses a training set of event sequences to train a genetic algorithm-based miner and a test set to tune its performance. However, the fact that mining is applied on sequences rather than unordered data items, together with the goal of predicting rare events, leads to a customized set of quality measures and to a specialized mining mechanism.

The *timeweaver* has been designed to predict hardware failures. The business goal is to identify events, which precede such a failure *and* lie within a time range (the "monitoring time") at which intervention is possible to prevent the failure. The mining goal is then to generate "prediction patterns" that forecast the target events (in this case, the failures). A prediction pattern is a sequence of events (probably interleaved with wildcards) subject to ordering and temporal constraints.

The semantics of event prediction are redefined in that "a target event is predicted if at least one prediction is made within its prediction period, regardless of any subsequent inaccurate predictions." This definition implies that the quality (or reliability) of a positive prediction is not affected by the existence of negative predictions concerning the same target event. Conventional quality measures based on accuracy are not appropriate for this notion of event prediction. Two measures are defined, *recall* and *precision*. Recall is the percentage of events correctly predicted. Precision is the percentage of predictions that are correct.

The *timeweaver* uses a genetic algorithm to generate patterns of varying length. The genetic algorithm operates with a population of patterns that are initially only one event long. The crossover operator supports the generation of longer patterns in subsequent populations. The selection, crossover, and mutation operators are designed with two incompatible goals in mind—to prevent premature convergence and to ensure diversity of the population so that most of the target events are covered. For this, the notion of "shared fitness" is defined which depends on the precision and recall of its population member and on its phenotype distance from the other members of the population. After generating a satisfactory initial group of patterns, a set of prediction rules is created in a second pass by selecting those patterns that improve the collective recall of the set without sacrificing precision.

Experiments with the *timeweaver* are reported with satisfactory results and it has also been compared favorably to three other classifiers (FOIL, C4.5, and RIPPER). The major reason for their poor performance is argued to be the high skew in the classification examples due to the rarity of the events to be predicted, although, for the latter two, the encoding of the event sequences into classified examples also resulted in some loss of the temporal information.

Dietterich and Michalski also investigate the use of nontemporal classification methods to temporal pattern discovery [82]. Their approach was to develop a *sequence-generating* rule that characterizes observed events and that can be used to generate credible and consistent next events (in the sense that the predicted events adhere to at least one of the possible descriptions of the observed events). An example of the use of the approach can be found in [83].

6 Measuring Interestingness

6.1 Interestingness in Static Data

While the development of data mining techniques has become an effective method for overcoming the time and human limitations inherent in analyzing large quantities of data, despite much important work [84], [85], [86], [87], [88], the problem of deciding which rules are of interest still remains a difficult issue. The difficulty of the problem is hardly surprising; the number of rules possible to mine from a database increases rapidly with the number of attributes and in some cases, with the number of domain values and, thus, with a large number of attributes, even the most efficient filtering mechanisms are likely to encounter problems. This is especially so since interesting rules:

- are not necessarily those that occur the most frequently,
- are not structured or do not contain any elements which are different from *noninteresting* rules,
- may only be interesting in certain contexts, unknown to the mining routine.

In the initial association rule mining proposals of Agrawal et al. [30], the two metrics of support and confidence were included. Support is the extent to which the data was relevant (either positively or negatively) to the rule in question while confidence is the extent to which, within those that were relevant, the proposal is upheld. For some applications, these two interestingness metrics can be misleading ([87]) and a number of other measures have been proposed [89].

6.2 Extensions to Temporal Mining

The search space of patterns satisfying some statistical properties of dominance and high confidence is still large. Moreover, as pointed out by Silberschatz and Tuzhilin [86], not all patterns that are statistically dominant are of interest. The work of [86] focuses on associations among objects that have no time dimension, such as products purchased together in a supermarket. The notion of *nterestingness* for event sequences is addressed in [68] in which Berger and Tuzhilin suggest that a pattern is interesting if the ratio of its actual to expected occurrences exceeds a given threshold. Since this measure yields the complete string as the most interesting pattern, a (rather artificial) limit to the maximum pattern length must be given. It is proven that the problem of finding interesting patterns is NP-complete. Moreover, it is pointed out that the property of interestingness is not monotone since a pattern may be interesting, even if its subpatterns are not. Hence, all algorithms building frequent episodes incrementally are inappropriate for the discovery of interesting patterns. This affects all the algorithms mentioned above with the exception of the algorithm in [66], which uses a different discovery principle to alleviate exactly this problem of nonmonotonicity.

Berger and Tuzhilin propose two algorithms, both of which build patterns incrementally by extending each pattern in both directions by probing all applicable temporal operators from a predefined set [68]. The *naive* algorithm proceeds exhaustively. The *main* algorithm has a more intelligent way of selecting which pattern to expand at each step. Omitting all details here, this is the pattern that mazimizes the expected interestingness when extended in any of the two directions.

Das et al. perform sequence mining and rank discovered rules according to their "informativeness" [80]. They use the J-measure of "informativeness" proposed by Smyth and Goodman [79]. This measure compares the posterior probability of each rule consequent given the antecedent with the prior probability of the consequent, as done by the cross-entropy measure, but also takes the prior probability of the antecedent into account. For the rule $A \rightarrow B$ the J-measure is expressed by the formula shown in Fig. 4

$$J(B,A) = p(A) \cdot \left(p(B|A) \cdot log \left(\frac{p(B|A)}{p(B)} \right) + (1 - p(B|A)) \cdot log \left(\frac{1 - p(B|A)}{1 - p(B)} \right) \right)$$

Fig. 4. A measure of informativeness for mining results [79].

In their work on activity monitoring, Fawcett and Provost define interestingness on the basis of deviation from normal activity [28]. In their model, a temporal sequence is a series of events, which may evolve normally in time or exhibit "positive activity," i.e., nonnormal behavior. The authors discuss different approaches for specifying the notion of positive activity. The *profiling* method builds patterns of normal activity and observes all deviations from these patterns as positive activities. The *discriminating* method instead builds patterns of positive activity, usually with reference to normal activity, and uses these patterns to detect positive activities directly.

While these studies focus on sequences of *events* occurring in time order, Chakrabarti et al. [42] discuss interestingness in the context of *rule* evolution. In particular, they observe how the statistics of association rules vary over time. A variation is interesting if it is "surprising," i.e., unexpected. Their approach also addresses the subject of rule evolution, as discussed in Section 3.2.

7 Data Mining Requirements and Environments

In recent years, there have been numerous efforts in formalizing the temporal properties of data and building temporal databases. A number of relational and object-oriented temporal DBMSs have already emerged. They are equipped with powerful query languages for the retrieval of information according to their temporal properties. Recent advances in those query languages include the support of aggregation and grouping, a prerequisite for temporal warehousing and mining.

7.1 Some Examples of Temporal Mining Systems

At present, no temporal data mining systems have been developed to mine from temporal databases and few from the other forms of data discussed above. Nevertheless, some temporal inductive learning algorithms have been developed, albeit in an application specific manner. For example, in the area of medical health, Wade et al. developed a system that detects temporal patterns in patient drug usage data [56]. The temporal context is used to ensure that the inferences derived regarding drug use are temporally valid (for example, drugs are sometimes only incompatible if they are taken concurrently). The system searches for known drug combinations and no explicit mining of new data is performed.

In another medical example, the RX project [33] utilizes knowledge discovery techniques to discover causal relationships from temporal data. In what is an early example of a knowledge-based management system, RX consults a knowledge base to determine interesting facts about the temporal data. Facts confirmed by the medical expert are added to the knowledge base. The process is two-phase; in the first phase, hypotheses are constructed by a discovery module from a subset of the database; in the second, these hypotheses are validated against the entire database.

There is a high dependence upon human interaction within the learning process in RX due to the uncertainty of the domain and the possibility for spurious relationships. While fully autonomous mining processes were originally seen as an achievable goal, current opinion indicates that this is unlikely [90].

In contrast, more recently, Agrawal et al. [91] discuss a system of mining process models from the logs generated by previous executions of a process. The goal is to allow easier introduction of a new workflow system through the provision of a model that captures existing behavior. The system produces activity graphs and, in this respect, the research has some similarities with that of web mining research [92].

7.2 Temporal Mining in Temporally Aware Systems

To date, temporal mining has been applied predominantly on nontemporal and usually nondatabase data. There are two interdependent reasons for this. First, sequences of events are not easy to model and query in a conventional relational DBMS. Despite the date and time manipulation services offered by SQL, the functionality required (such as identifying shapes in time series [53]) is not supported yet. Second, analysis of temporal data is one of the earliest domains of knowledge discovery. Suites of algorithms have been developed around the simpler notion of a flat data file. These two reasons have resulted in the area of temporal data analysis being largely separated from the advances in temporal databases.

Database systems with temporal capabilities are now beginning to emerge and the requests able to be asked of them commonly exceed those able to be asked of conventional databases. Establishing a framework for temporal data analysis is a necessity and, in some cases, it may be the need for temporal data analysis that may drive the use of a temporal database system.

Saraee and Theodoulidis proposed an initial agenda of knowledge discovery issues to be addressed in the context of temporal databases [93]. They point out that concepts such as confidence and interestingness should be extended to incorporate the temporal semantics of the data. They call for the exploitation of the time dimension during the discovery of frequent episodes and sequence rules. They further address the notion of roll-up and drill-down for temporal data (albeit using a different terminology). Finally, they stress the importance of incorporating the fundamental operations of knowledge discovery into the database core. In this direction, they mention the ORES temporal DBMS, which supports classification, generalization, aggregation, and grouping on temporal data modeled according to the Entity Relationship Time data model [94].

Grouping and aggregation of temporal data is also the focal point of the work of Dumas et al. [95]. They present the temporal query language and processing mechanism of the TEMPOS DBMS mentioned above [96]. TEMPOS offers pattern matching, grouping on conventional and temporal attributes, and a warehouse-like notion of *roll-up*. Such services are still not adequate for temporal data mining but offer the basic functionality for temporal OLAP.

More recently, Shasha focuses on database support for time series in financial data [97] and provides a list of necessary features. First, sequences should be treated as first class objects within the database. Combinations of multiple sequences for statistical analysis, aggregation, etc. are necessary. User-defined functions will be needed as even the most expressive query engine cannot express all combinations and interpolations that may be required for a particular application. In terms of processing support, execution must be efficient, both in RAM and on disc. Good performance should be combined with a relational vocabulary, which allows modeling of different value semantics (such as inventory and expense values, which cannot be interpolated in the same way). Finally, the semantics of the time dimension should be better exploited, including the distinction between valid and transaction time. This discussion is combined with a critical presentation of four software products for time series analysis, namely, FAME, S-Plus, SAS, and KSQL, which offer, to varying extents, some of the listed features.

An important aspect of data processing support concerns the *efficient* discovery of trend similarities. Studies on this subject propose preprocessing, indexing, and offline precomputation of auxiliary information. In [51], a Discrete Fourier Transform for each sequence is proposed, as a result of which only the first few coefficients of the transformed structure need to be stored in the (R-tree) index. The IMPACTS system proposed by Huang and Yu in [98] aims to support efficient processing of complex queries posed interactively, i.e., for which no intermediate data can be computed in advance. However, no template-based mining language is proposed in [98] to exploit the services of IMPACTS.

Index-based techniques for similarity search among time series and for fast subsequence matching have been proposed in [51], [99], [100], [101]. The goal is to speed up the discovery of time-series that contain subsequences similar to a given pattern (or shape [53]). Goldin and Kanellakis apply a constraint-based mechanism to the same problem of trend similarity querying [102]. Rafiei and Mendelzon use R-tree indexing to speed up their trend similarity queries [103]. In [104], [105], the focus is on efficiently executing the join operations underlying the similarity queries.

Jagadish et al. proposed an application-independent framework for similarity queries [106], including a query language for the specification of patterns and for the execution of similarity-preserving transformations. This framework was originally tested to find similarities among sequences. In later work, Rafiei and Mendelzon have designed a set of transformations and a query processing algorithm for similarity among time-series [103].

8 Conclusions and Further Research

In this paper, we have discussed a framework for reviewing research related to temporal mining and have reviewed research contributions related to various aspects of the temporal data mining and knowledge discovery. This has included a discussion of temporal rules and their semantics, the discovery of temporal rules (including temporal associations rules, time sequences and series, temporal classification, etc.), as well as, the issue of interestingness, and temporal mining (and meta-mining) environments. Many of the techniques we have addressed are applicable equally to temporal data sets, series of static data sets, and temporally oriented data warehouses.

Although much research is being performed in this domain, there are still several open issues. First, concepts, algorithms, and supportive techniques (such as indexing) designed for mining over static data should be extended to take the temporal dimension into account. In particular, mining algorithms for static data are not directly applicable on temporal data. Concepts such as "interestingness" should incorporate the temporal dimension of the data and of the rules extracted from them. Database systems should offer data structures and operators appropriate for efficient temporal data mining.

Second, knowledge evolves with time and must be maintained and updated. Incremental methods of data mining ([25], [37], [39], [41], [107], [108]) address only one aspect of the problem, namely, the updating of previously discovered rules as the underlying collection of static data changes. Further aspects to be taken into account include: 1) the changes in a collection of time-stamped data with different valid time intervals, 2) the effect of the time intervals separating the mining sessions, and 3) the temporal properties of the rules being discovered. This last issue is also relevant to higher order data mining, namely, to the discovery of new rules by mining the results of consecutive data mining sessions.

A further particularly challenging extension to this work is that of spatio-temporal data mining. Mining from geographic data, for example, frequently requires temporal techniques, as geographical phenomena (and, therefore, the data) over which knowledge discovery techniques are applied are rarely stationary. Such geographic examples include analyses of geological formations, the geographical distribution of wildlife on our planet, sea water quality, or the impact of fires in forest regeneration, and commonly have both temporal and spatial aspects in which the evolution of a situation is captured. Research into spatial knowledge discovery is outside of the scope of this paper, but some of the papers in the area are listed in [12]. In addition, a specialist workshop on temporal, spatial and spatio-temporal data mining was held with PKDD [109].

In summary, temporal data mining is a challenging research area for which many exciting problems remain open. The incorporation of temporal semantics to existing data mining techniques provides additional semantics to static rules and, in some cases, may enlarge the applicability of data mining to new application domains. This holds particularly for application domains like law, medicine, and financial and environmental analysis, where sophisticated temporal data mining techniques hold the potential to yield useful information.

Acknowledgments

This work was conceived while the authors attended the Integrating Spatial and Temporal Databases Workshop at Schloss Dagstuhl in November 1998. We would like to record our thanks to IBFI GmbH for the ability to attend and for providing a conducive atmosphere in which to work. We would also like to express our sincere thanks to the anonymous referees for their careful and detailed reviews.

References

[1] H. Mannila, "Methods and Problems in Data Mining," *Proc. Int'l Conf. Database Theory,* F. Afrati and P. Kolaitis, eds., pp. 41-55, 1997.

[2] C.J. Matheus, P.K. Chan, and G. Piatetsky-Shapiro, "Systems for Knowledge Discovery in Databases," *IEEE Trans. Knowledge and Data Eng.,* vol. 5, no. 6, pp. 903-913, Dec. 1993.

[3] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery: An Overview," *Advances in Knowledge Discovery and Data Mining,* U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds., pp. 1-34, 1996.

[4] R. Agrawal and R. Srikant, "Mining Sequential Patterns," *Proc. 11th Int'l Conf. Data Eng.,* P.S. Yu and A.S.P. Chen, eds., pp. 3-14, 1995.

[5] A.U. Tansel, J. Clifford, S.K. Gadia, S. Jajodia, A. Segev, and R.T. Snodgrass, *Temporal Databases: Theory, Design and Implementation.* Redwood City, Calif.: Benjamin Cummings, 1993.

[6] C. Zaniolo, S. Ceri, C. Faloutsos, R.T. Snodgrass, V.S. Subrahmanian, and R. Zicari, *Advanced Database Systems.* San Francisco: Morgan Kaufmann, 1997.

[7] N. Kline, "An Update of the Temporal Database Bibliography," *SIGMOD Record,* vol. 22, no. 4, pp. 66-80, 1993.

[8] L.E. McKenzie, "Bibliography: Temporal Databases," *SIGMOD Record,* vol. 15, no. 4, pp. 40-52, 1986.

[9] M.D. Soo, "Bibliography on Temporal Databases," *SIGMOD Record,* vol. 20, no. 1, pp. 14-23, 1991.

[10] R.B. Stam and R. Snodgrass, "A Bibliography on Temporal Databases," *Data Eng.,* vol. 7, no. 4, pp. 53-61, 1988.

[11] Y. Wu, S. Jajodia, and X.S. Wang, "Temporal Database Bibliography Update," *Temporal Databases—Research and Practice,* O. Etzioni, S. Jajodia, and S. Sripada, eds., pp. 338-366, 1998.

[12] J.F. Roddick and M. Spiliopoulou, "A Bibliography of Temporal, Spatial and Spatio-Temporal Data Mining Research," *SIGKDD Explorations,* vol. 1, no. 1, pp. 34-38, 1999, the bibliography is also being maintained and updated through the authors' website.

[13] J.F. Roddick, K. Hornsby, and M. Spiliopoulou, "An Updated Bibliography of Temporal, Spatial, and Spatio-Temporal Data Mining Research," *Post-Workshop Proc. Int'l Workshop Temporal, Spatial and Spatio-Temporal Data Mining (TSDM 2000),* J.F. Roddick and K. Hornsby, eds., pp. 147-163, 2001.

[14] P. Hoschka and W. Klösgen, "A Support System for Interpreting Statistical Data," *Knowledge Discovery in Databases,* G. Piatetsky-Shapiro and W.J. Frawley, eds., pp. 325-345, 1991.

[15] J. Han, W. Gong, and Y. Yin, "Mining Segment-Wise Periodic Patterns in Time-Related Databases," *Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining,* pp. 214-218, 1998.

[16] J.F. Allen, "Maintaining Knowledge About Temporal Intervals," *Comm. ACM,* vol. 26, no. 11, pp. 832-843, 1983.

[17] C. Freksa, "Temporal Reasoning Based on Semi-Intervals," *Artificial Intelligence,* vol. 54, pp. 199-227, 1992.

[18] M.B. Vilain, "A System for Reasoning about Time," *Porc. Nat'l Conf. Artificial Intelligence,* pp. 197-201, 1982.

[19] R. Snodgrass and I. Ahn, "Temporal Databases," *Computer,* vol. 19, pp. 35-42, 1986.

[20] C. Jensen, C.E. Dyerson, M. Bshlen, J. Clifford, R. Elmasri, S.K. Gadia, F. Grandi, P. Hayes, S. Jajodia, W. Käfer, N. Kline, N. Lorentzos, Y. Mitsopoulos, A. Montanari, D. Nonen, E. Peressi, B. Pernici, J.F. Roddick, N.L. Sarda, M.R. Scalas, A. Segev, R.T. Snodgrass, M.D. Soo, A. Tansel, P. Tiberio, and G. Wiederhold, "A Consensus Glossary of Temporal Database Concepts," *Temporal Databases—Research and Practice,* O. Etzioni, S. Jajodia, and S. Sripada, eds., pp. 367-405, Feb. 1998.

[21] T.L. Dean and D.V. McDermott, "Temporal Database Management," *Artificial Intelligence,* vol. 32, no. 1, pp. 1-55, 1987.

[22] D. McDermott, "A Temporal Logic for Reasoning about Processes and Plans," *Cognitive Science,* vol. 6, pp. 101-155, 1982.

[23] J.F. Allen, "An Interval-Based Representation of Temporal Knowledge," *Proc. Seventh Int'l Joint Conf. Artificial Intelligence,* A. Drinan, ed., pp. 221-226, 1981.

[24] S. Al-Naemi, "A Theoretical Framework for Temporal Knowledge Discovery," *Proc.Int'l Workshop Spatio-Temporal Databases,* pp. 23-33, 1994.

[25] C.P. Rainsford, "Accommodating Temporal Semantics in Data Mining and Knowledge Discovery," PhD thesis, Univ. of South Australia, 1999.

[26] J.F. Roddick, NG. Craske, and T.J. Richards, "Handling Discovered Structure in Database Systems," *IEEE Trans. Knowledge and Data Eng.,* vol. 8, no. 2, pp. 227-240, Apr. 1996.

[27] J. Clifford, V. Dhar, and A. Tuzhilin, "Knowledge Discovery from Databases: The NYU Project," Technical Report IS-95-12, New York Univ., 1995.

[28] T. Fawcett and F. Provost, "Activity Monitoring: Noticing Interesting Changes in Behavior," *Proc. Fifth Int'l Conf. Knowledge Discovery and Data Mining,* S. Chaudhuri and D. Madigan, eds., pp. 53-62, 1999.

[29] *Time Series Prediction: Forecasting the Future and Understanding the Past,* A.S.Weigend and N.A.Gershenfeld, eds., *Proc NATO Advanced Research Workshop on Comparative Time Series Analysis,* May 1993.

[30] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* vol. 22, pp. 207-216, 1993.

[31] X. Chen, I. Petrounias, and H. Heathfield, "Discovering Temporal Association Rules in Temporal Databases," *Proc. Int'l Workshop Issues and Applications of Database Technology (IADT '98),* pp. 312-319, 1998.

[32] W. Klösgen, "Deviation and Association Patterns for Subgroup Mining in Temporal, Spatial, and Textual Data Bases," *Proc. First Int'l Conf. Rough Sets and Current Trends in Computing (RSCTC '98),* pp. 1-18, 1995.

[33] R.L. Blum, "Discovery, Confirmation and Interpretation of Causal Relationships from a Large Time-Oriented Clinical Database: The RX Project," *Computers and Biomedical Research,* vol. 15, no. 2, pp. 164-187, 1982.

[34] R.L. Blum, "Discovery and Representation of Causal Relationships from a Large Time-Oriented Clinical Database: The RX Project," *Lecture Notes in Medical Informatics,* vol. 19, 1982.

[35] J.M. Long, E.A. Irani, and J.R. Slagle, "Automating the Discovery of Causal Relationships in a Medical Records Database," *Knowledge Discovery in Databases,* G. Piatetsky-Shapiro and W. J. Frawley, eds., pp. 465-476, 1991.

[36] M. Pechoucek, O. Stepánková, and P. Miksovský, "Maintenance of Discovered Knowledge," *Proc. Third European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD '99),* J. Zytkow and J. Rauch, eds., pp. 476-483, Sept. 1999.

[37] D.W. Cheung, V.T. Ng, and B.W. Tam, "Maintenance of Discovered Knowledge: A Case in Multi-Level Association Rules," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining (KDD-96),* E. Simoudis, J. Han, and U. Fayyad, eds., pp. 307-310, 1996.

[38] D.W. Cheung, J. Han, V.T. Ng, and C.Y. Wong, "Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique," *Proc. Int'l Conf. Data Eng., (ICDE '96)* S.Y.W. Su, ed., pp. 106-114, 1996.

[39] D.W. Cheung, S.D. Lee, and B. Kao, "A General Incremental Technique for Maintaining Discovered Association Rules," *Proc. Fifth Int'l Conf. Database Systems for Advanced Applications,* 1997.

[40] N.F. Ayan, A.U. Tansel, and E. Arkun, "An Efficient Algorithm to Update Large Itemsets with Early Pruning," *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '99),* pp. 287-291, Aug. 1999.

[41] C.P. Rainsford, M.K. Mohania, and J.F. Roddick, "A Temporal Windowing Approach to the Incremental Maintenance of Association Rules," *Proc. Eighth Int'l Database Workshop, Data Mining, Data Warehousing and Client/Server Databases (IDW '97),* J. Fong, ed., pp. 78-94, 1997.

[42] S. Chakrabarti, S. Sarawagi, and B. Dom, "Mining Surprising Patterns Using Temporal Description Length," *Proc. 24th Int'l Conf. Very Large Databases (VLDB '98),* A. Gupta, O. Shmueli, and J. Widom, eds., pp. 606-617, 1998.

[43] X. Chen and I. Petrounias, "Mining Temporal Features in Association Rules," *Proc. Third European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD '99),* J. Zytkow and J. Rauch, eds., pp. 295-300, Sept. 1999.

[44] T. Abraham and J.F. Roddick, "Incremental Meta-Mining from Large Temporal Data Sets," *Advances in Database Technologies, Proc. First Int'l Workshop Data Warehousing and Data Mining (DWDM '98),* Y. Kambayashi, D.K. Lee, E.-P. Lim, M. Mohania, and Y. Masunaga, eds., pp. 41-54, 1999.

[45] T. Abraham, "Knowledge Discovery in Spatio-Temporal Databases," PhD thesis, Univ. of South Australia, 1999.

[46] M. Spiliopoulou and J.F. Roddick, "Higher Order Mining: Modelling and Mining the Results of Knowledge Discovery," *Proc. Second Int'l Conf. Data Mining Methods and Databases,* 2000.

[47] D.J. Berndt and J. Clifford, "Finding Patterns in Time Series: A Dynamic Programming Approach," *Advances in Knowledge Discovery and Data Mining,* U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds., pp. 229-248, 1995.

[48] E. Keogh and M. Pazzani, "An Enhanced Representation of Time Series which Allows Fast and Accurate Classification, Clustering and Relevance Feedback," *Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining (KDD '98),* R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, eds., pp. 239-241, 1998.

[49] E. Keogh and M. Pazzani, "Relevance Feedback Retrieval of Time Series Data," *Proc. 22nd Ann. Int'l ACM-SIGIR Conf. Research and Development in Information Retrieval,* 1999.

[50] E. Keogh and M. Pazzani, "Scaling Up Dynamic Time Warping to Massive Datasets," *Proc. Third European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD '99),* J. M. Zytkow and J. Rauch, eds., pp. 1-11, 1999.

[51] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search in Sequence Databases," *Proc. Int'l Conf. Foundations of Database Organisation and Algorithms, (FODO '93),* D. Lomet, ed., pp. 69-84, 1993.

[52] R. Agrawal, K.-I. Lin, H.S. Sawhney, and K. Shim, "Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases," *Very Large Datatbases,* pp. 490-501, 1995.

[53] R. Agrawal, G. Psaila, E.L. Wimmers, and M. Zaot, "Querying Shapes of Histories," *Proc. 21st Int'l Conf. Very Large Databases (VLDB '95),* U. Dayal, P.M.D. Gray, and S. Nishio, eds., pp. 502-514, 1995.

[54] Y. Qu, C. Wang, and S.X. Wang, "Supporting Fast Search in Time Series for Movement Patterns in Multiple Scales," *Proc. Seventh Int'l Conf. Information and Knowledge Management (CIKM '98),* pp. 251-258, 1998.

[55] E. Keogh and P. Smyth, "A Probabilistic Approach to Fast Pattern Matching in Time Series Databases," *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining,* D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, eds., pp. 24-30, 1997.

[56] T.D. Wade, P.J. Byrns, J.F. Steiner, and J. Bondy, "Finding Temporal Patterns—A Set Based Approach," *Artificial Intelligence in Medicine,* no. 6, pp. 263-271, 1994.

[57] *Pattern Discovery in Biomolecular Data: Tools, Techniques, and Applications,* J. Wang, B. Sharpiro, and D. Shasa, eds. Oxford Univ. Press, 1999.

[58] D. Shasha and K. Zhang, "Approximate Tree Pattern Matching," *Pattern Matching in Strings, Trees, and Arrays,* A. Apostolico and Z. Galil, eds., chapter 14, Oxford Univ. Press, 1999.

[59] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalisations and Performance Improvements," *Proc. Int'l Conf. Extending Database Technology (EDBT '96),* P.M.G. Apers, M. Bouzeghoub, and G. Gardarin, eds., pp. 3-17, 1996.

[60] H. Mannila and H. Toivonen, "Discovering Generalised Episodes Using Minimal Occurences," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining (KDD-96),* pp. 146-151, 1996.

[61] B. Padmanabhan and A. Tuzhilin, "Pattern Discovery in Temporal Databases: A Temporal Logic Approach," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining,* E. Simoudis, J. Han, and U. Fayyad, eds., 1996.

[62] C. Bettini, S.X. Wang, S. Jagodia, and J.-L. Lin, "Discovering Frequent Event Patterns with Multiple Granularities in Time Sequences," *IEEE Trans. Knowledge and Data Eng.,* vol. 10, no. 2, pp. 222-237, Mar./Apr. 1998.

[63] M.J. Zaki, N. Lesh, and M. Ogihara, "Planmine: Sequence Mining for Plan Failures," *Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining (KDD '98),* R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, eds., pp. 369-373, 1998, A more detailed version appears in *Artificial Intelligence Review*, special issue on the application of data mining, 1999.

[64] G.M. Weiss and H. Hirsh, "Learning to Predict Rare Events in Event Sequences," *Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining (KDD '98),* R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, eds., pp. 359-363, 1998.

[65] M.J. Zaki, "Efficient Enumeration of Frequent Sequences," *Proc. Seventh Int'l Conf. Information and Knowledge Management (CIKM'98),* pp. 68-75, 1998.

[66] M. Spiliopoulou, "The Laborious Way from Data Mining to Web Mining," *Int'l J. Computer Systems, Science and Eng.,* special issue on semantics of the web, vol. 14, pp. 113-126, Mar. 1999.

[67] M. Spiliopoulou and L. C. Faulstich, "WUM: A Tool for WebUtilization Analysis," *Proc. Extending Database Technology Workshop (WebDB '98),* pp. 184-203, 1999.

[68] G. Berger and A. Tuzhilin, "Discovering Unexpected Patterns in Temporal Data Using Temporal Logic," *Temporal Databases—Research and Practice,* O. Etzion, S. Jajodia, and S. Sripada, eds., pp. 281-309, 1998.

[69] A.G. Büchner, M. Baumgarten, S.S. Anand, M.D. Mulvenna, and J.G. Hughes, "Navigation Pattern Discovery from Internet Data," *Proc. Conf. Knowledge Discovery in Databases (WEBKDD '99),* Aug. 1999.

[70] Y. Cai, N. Cercone, and J. Han, "An Attribute-Oriented Approach for Learning Classification Rules from Relational Databases," *Proc. Sixth IEEE Int'l Conf. Data Eng.,* pp. 281-288, 1990.

[71] J. Han, Y. Cai, and N. Cercone, "Data-Driven Discovery of Quantitative Rules in Relational Databases," *IEEE Trans. Knowledge and Data Eng.,* vol. 5, no. 1, pp. 29-40, 1993.

[72] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, and A. Swami, "An Interval Classifier for Database Mining Applications," *Proc. 18th Int'l Conf. Very Large Data Bases,* L.-Y. Yuan, ed., pp. 560-573, 1992.

[73] C.P. Rainsford and J.F. Roddick, "Database Issues in Knowledge Discovery and Data Mining," *Australian J. Information Systems,* vol. 6, no. 2, pp. 101-128, 1999.

[74] Y. Li, X.S. Wang, and S. Jajodia, "Discovering Temporal Patterns in Multiple Granularities," *Proc. Int'l Workshop Temporal, Spatial and Spatio-Temporal Data Mining (TSDM 2000),* J.F. Roddick and K. Hornsby, eds., 2000.

[75] J.F. Roddick and J.D. Patrick, "Temporal Semantics in Information Systems—A Survey," *Information Systems,* vol. 17, no. 3, pp. 249-267, 1992.

[76] A.S. Weigend, F. Chen, S. Figlewski, and S.R. Waterhouse, "Discovering Technical Trades in the T-Bond Futures Market," *Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining (KDD '98),* R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, eds., pp. 354-358, 1998.

[77] P. Cheeseman and J. Stutz, "Bayesian Classification (AutoClass): Theory and Results," *Knowledge Discovery in Data Bases II,* 1995.

[78] T. Oates, "Identifying Distinctive Subsequences in Multivariate Time Series by Clustering," *Proc. Fifth Int'l Conf. Knowledge Discovery and Data Mining,* S. Chaudhuri and D. Madigan, eds., pp. 322-326, 1999.

[79] P. Smyth and R.M. Goodman, "An Information Theoretic Approach to Rules Induction from Databases," *IEEE Trans. Knowledge and Data Eng.,* vol. 4, no. 4, pp. 301-316, Aug. 1992.

[80] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth, "Rule Discovery from Time Series," *Proc. Fourth Int'l Conf. (KDD '98),* R. Agrawal, P. Stolorz, and G. Piatesky-Shapiro, eds., pp. 16-22, Aug. 1998.

[81] M.J. Zaki, N. Lesh, and M. Ogihara, "Mining Features for Sequence Classification," *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '99),* pp. 342-346, Aug. 1999.

[82] T.G. Dietterich and R.S. Michalski, "Discovering Patterns in Sequences of Events," *Artificial Intelligence,* vol. 25, pp. 187-232, 1985.

[83] R. Sasisekharan, V. Seshadri, and S.M. Weiss, "Data Mining and Forecasting in Large-Scale Telecommunication Networks," *IEEE Expert,* vol. 11, no. 1, pp. 37-43, 1996.

[84] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo, "Finding Interesting Rules from Large Sets of Discovered Association Rules," *Proc. Third Int'l Conf. Information and Knowledge Management,* N.R. Adam, B.K. Bhargava, and Y. Yesha, eds., pp. 401-407, 1994.

[85] W. Klösgen, "Efficient Discovery of Interesting Statements in Databases," *J. Intelligent Information Systems,* no. 4, pp. 53-69, 1995.

[86] A. Silberschatz and A. Tuzhilin, "What Makes Patterns Interesting in Knowledge Discovery Systems," *IEEE Trans. Knowledge and Data Eng.,* vol. 8, no. 6, pp. 970-974, Dec. 1996.

[87] C. Silverstein, R. Motwani, and S. Brin, "Beyond Market Baskets: Generalising Association Rules to Corelations," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* J. Peckham, ed., pp. 265-276, 1997.

[88] B. Padmanabhan and A. Tuzhilin, "A Belief-Driven Method for Discovering Unexpected Patterns," *Proc. Knowledge Discovery in Databases (KDD '98),* pp. 94-100, Aug. 1998.

[89] R.J. Bayardo Jr. and R. Agrawal, "Mining the Most Interesting Rules," *Proc. Fifth Int'l Conf. Knowledge Discovery and Data Mining,* S. Chaudhuri and D. Madigan, eds., pp. 145-154, 1999.

[90] J.F. Roddick, "Data Warehousing and Data Mining: Are We Working on the Right Things?" *Advances in Database Technologies,* Y.Kambayashi, D.K. Lee, E.-P. Lim, Y. Masunaga, and M. Mohania, eds., pp. 141-144, 1999.

[91] R. Agrawal, D. Gunopulos, and F. Leymann, "Mining Process Models from Workflow Logs," *Proc. Sixth Int'l Conf. Extending Database Technology (EDBT '98),* H.-J. Schek, F. Saltor, and I. Ramos, eds., pp. 469-483, 1998.

[92] S.K. Madria, S.S. Bhowmick, W.K. Ng, and E.-P. Lim, "Research Issues in Web Data Mining," *Proc. First Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK '99),* M.K. Mohania and A.M. Tjoa, eds., pp. 303-312, 1999.

[93] M.H. Saraee and B. Theodoulidis, "Knowledge Discovery in Temporal Databases: The Initial Step," *Proc. Conf. Deductive Object Oriented Databases Post-Conf. Workshop Knowledge Discovery in Databases and DOOD,* K. Ong, S. Conrad, and T.W. Ling, eds., pp. 17-22, 1995.

[94] B. Theodoulidis, A. Ait-Braham, G. Andrianopoulos, J. Chaudhary, G. Karvelis, and S. Sou, "The ORES Temporal Database Management System," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* R.T. Snodgrass and M. Winslett, eds., p. 511, 1994.

[95] M. Dumas, M.C. Fauvet, and P.C. Scholl, "Handling Temporal Grouping and Pattern-Matching Queries in a Temporal Object Model," *Proc. Seventh Int'l Conf. Information and Knowledge Management (CIKM'98),* 1998.

[96] M.C. Fauvet, S. Chardonnel, D. Marlon, P.C. Scholl, and P. Dumolard, "Analyse de Données Géographiques: Application des Bases de Données Temporelles," *Revue Internationale de Géomatique,* 1999.

[97] D. Shasha, "Time Series in Finance: The Array Database Approach." http://www.cs. nyu.edu/cs/faculty/shasha/papers/jagtalk.html Aug. 1998.

[98] Y.-W. Huang and P.S. Yu, "Adaptive Query Processing for Time-Series Data," *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD'99),* pp. 282-286, Aug. 1999.

[99] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," *Proc. ACM SIGMOD Conf. Management of Data,* pp. 419-429, 1994.

[100] F. Korn, H.V. Jagadish, and C. Faloutsos, "Efficiently Supporting ad hoc Queries in Large Datasets of Time Sequences," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* J. Peckham, ed., pp. 289-300, 1997.

[101] B.K. Yi, H.V. Jagadish, and C. Faloutsos, "Efficient Retrieval of Similar Time Sequences under Time Warping," *Proc. Int'l Conf. Data Eng. (ICDE '98),* pp. 201-208, 1998.

[102] D.Q. Goldin and P.C. Kanellakis, "On Similarity Queries for Time-Series Data: Constraint Specification and Implementation," *Proc. First Int'l Conf. Principles and Practice of Constraint Programming (CP '95),* pp. 137-153, 1995.

[103] D. Rafiei and A.O. Mendelzon, "Similarity-Based Queries for Time Series Data," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* J. Peckham, ed., pp. 13-25, 1997.

[104] J.C. Shafer and R. Agrawal, "Parallel Algorithms for High-Dimensional Similarity Joins for Data Mining Applications," *Proc. Int'l Conf. Very Large Databases (VLDB '97),* pp. 176-185, 1997.

[105] K. Shim, R. Srikant, and R. Agrawal, "High-Dimensional Similarity Joins," *Proc. Int'l Conf. Data Eng. (ICDE '97)* pp. 301-313, 1997.

[106] H.V. Jagadish, A.O. Mendelzon, and T. Milo, "Similarity-Based Queries," *Proc. ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems,* pp. 36-45, 1995.

[107] J. Hong and C. Mao, "Incremental Discovery of Rules and Structure by Hierarchical and Parallel Clustering," *Knowledge Discovery in Databases,* G. Piatetsky-Shapiro and W.J. Frawley, eds., chapter 10, pp. 177-194, 1991.

[108] K. Wang and J. Tan, "Incremental Discovery of Sequential Patterns," *Proc. ACM SIGMOD Workshop Research Issues on Data Mining and Knowledge Discovery,* 1996.

[109] *Temporal, Spatial and Spatio-Temporal Data Mining Proc. First Int'l Workshop.* J.F. Roddick and K. Hornsby, eds., 2001.

John F. Roddick received the BSc (Eng) (Hons) degree from Imperial College, London, the MSc degree from Deakin University, and the PhD degree from La Trobe University. He currently holds the SACITT chair of Information Technology in the School of Informatics and Engineering at the Flinders University of South Australia. Also, he has held positions at the Universities of South Australia and Tasmania and was a project leader and a consultant in the information technology industry. His technical interests include data mining and knowledge discovery, schema versioning, and enterprise systems. He is editor-in-chief of the *Journal of Research and Practice in Information Technology*, he is a fellow of the Australian Computer Society and the Institution of Engineers, Australia. He is a member of the IEEE Computer Society and the ACM.

Myra Spiliopoulou received the BSc and MSc degrees in mathematics and the PhD degree in computer science from the University of Athens, Greece, in 1986 and 1992, respectively. Between 1987-1994, she worked as a research assistant in the Department of Informatics, University of Athens and was involved in national and European projects on parallel database query optimization, hypermedia and multimedia modeling and querying, and on computer-aided education. Between 1994 and 2000 she was with the Institute of Information Systems at the Humboldt University, Berlin, Germany. In September 2000, she joined the faculty of Computer Science at the University of Magdeburg, Germany, as a guest professor for one semester. Since April 2001, she has been the professor of e-business in the Leipzig Graduate School of Management, Germany. Her research interests cover several KDD areas, including, web usage mining, temporal data mining, and XML DTD extraction from texts with mining techniques. She considers data and text mining as one phase of a complex process and investigates methodologies for the support of the whole process, in particular purloining techniques for data preparation and postmining techniques for the evaluation, and the lifelong monitoring of the results. She is a member of the ACM and of the IEEE Computer Society.