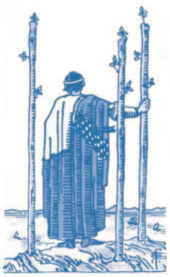


Methodology for Mining Meta Rules from Sequential Data

A Thesis
Presented to
The Faculty of Sciences

by

Paul Cotofrei

In Fulfillment
of the Requirements for the Degree
Doctor ès Science

Computer Science Department
University of Neuchâtel
June 2005

IMPRIMATUR POUR LA THESE

Methodology for Mining Meta Rules from Sequential Date

Paul COTOFREI

UNIVERSITE DE NEUCHATEL

FACULTE DES SCIENCES

La Faculté des sciences de l'Université de Neuchâtel, sur le rapport des membres du jury

MM. K. Stoffel (directeur de thèse),
J. Savoy, G. Jäger (Berne)
et I. Vaduva (Bucarest)

autorise l'impression de la présente thèse.

Neuchâtel, le 27 mai 2005

La doyenne:

Prof. M. Rahier

To my wife, Gina, and my son, Radu

ACKNOWLEDGEMENTS

This work would not be possible without the support and the comprehensibility of the people with whom I shared the ups and downs of the last six years of my life. I am greatly indebted to Professor Kilian Stoffel, for his courage of accepting me as PhD student, for his open-mindedness, enthusiasm and capacity to promote relationships based on mutual respect and friendship. Thanks to him and to all past and present members (Laura, Thorsten, Claudia, Iulian and Erik), it was always reigning a truly "family ambiance" inside our research group, Knowledge Information and Data Processing.

I am also grateful to Professor Jacques Savoy, especially for the constant support he gave to me and my family during the difficult process of integration (those which are at thousands of miles far from their home will understand). I want also to thanks my colleague, Dr. Abdelkader Belkoniene, for his kind encouragement and for sharing with me memorable personal experiences.

Finally, I'd like to acknowledge the help and advice given me by Professor Gerhard Jaeger, from the University of Bern, and by Professor Ion Vaduva, from the University of Bucharest.

This thesis was also supported by the Swiss National Science Foundation, Grant 2100-063 730, and by the University of Neuchâtel, which kindly hosted and supported my PhD. studies.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The purpose of this thesis is to respond to an actual necessity – the need to discover knowledge from huge data collection comprising multiple sequences that evolve over time – by proposing a methodology for temporal rule extraction. To obtain what we called *temporal rules*, a discretisation phase that extracts *events* from raw data is applied first, followed by an inference phase, where classification trees are constructed based on these events. The discrete and continuous characteristics of an event, according to its definition, allow the use of statistical tools as well as of techniques from artificial intelligence on the same data.

A theoretical framework for this methodology, based on first-order temporal logic, is also defined. This formalism permits the definition of the main notions (event, temporal rule, constraint) in a formal way. The concept of *consistent linear time structure* allows us to introduce the notions of *general interpretation*, of *support* and of *confidence*, the lasts two measure being the expression of the two similar concepts used in data mining. These notions open the possibility to use statistical approaches in the design of algorithms for inferring higher order temporal rules, denoted temporal meta-rules.

The capability of the formalism is extended to "capture" the concept of time granularity. To keep an unitary viewpoint of the meaning of the same formula at different time scales, the usual definition of the interpretation for a predicate symbol, in the frame of a temporal granular logic, is changed: it returns now the degree of truth (a real value between zero and one) and not the meaning of truth (one of the values *true* or *false*).

Finally, a probabilistic model is attached to the initial formalism to define a stochastic first-order temporal logic. By using advanced theorems from the stochastic limit theory, it was possible to prove that a certain amount of dependence (called near-epoch dependence) is the highest degree of dependence which is sufficient to induce the property of consistency.

CHAPTER I

INTRODUCTION

> "We are deluged by data — scientific data, medical data, demographic data, financial data, and marketing data. People have no time to look at this data. Human attention has become a precious resource. So, we must find ways to automatically analyze the data, to automatically classify it, to automatically summarize it, to automatically discover and characterize trends in it, and to automatically flag anomalies. This is one of the most active and exciting areas of the database research community. Researchers in areas such statistics, visualization, artificial intelligence, and machine learning are contributing to this field. The breath of the fields makes it difficult to gasp its extraordinary progress over the last few years". (Jim Gray, *Microsoft Research*, in **Foreword** of *Data Mining, Concepts and Techniques*, Han and Kamber [2001])

1.1 Data Mining

The situation described by the researcher from Microsoft is a reality in today's world : our capabilities of both generating and collecting data have been increasing rapidly in the last several decades. This explosive growth in stored data has generated an urgent need for new techniques and automated tools that can intelligently assist us in transforming the vast amount of data into useful information and knowledge. The discipline concerned with this task is now known as *data mining*.

If we try to capture this concept into a formal definition, then we can define data mining as

> "the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner" (Hand et al. [2001], pg. 1).

The relationship and summaries derived through a data mining exercise are often referred to as *models* or *patterns*. Examples include linear equations, rules, clusters, graphs, tree structures, and recurrent patterns in time series. The relationships and structures found within a set of data must, of course, be novel. Clearly, novelty — which remains an open research problem — must be measured relative to the user's prior knowledge. Unfortunately, few data mining algorithms take a user's prior knowledge into account.

While novelty is an important property of the relationships we seek, it is not sufficient to qualify a relationship as being worth finding. In particular, the relationships must be also understandable. For instance simple relationships are more readily understood than complicatesd ones, and may be well preferred, all else being equal.

The definition above refers to *observational data*, as opposed to *experimental data*. Data mining typically deals with data that have already been collected for some purpose other than data mining analysis. This means that the objectives of the data mining exercise play no role in the data collection strategy. This is one way in which data mining differs from of statistics, in which data are often collected by using efficient strategies to answer specific questions. For this reason, data mining is often referred to as *secondary* data analysis.

Many people treat data mining as a synonym for another popular used term, *Knowledge Discovery in Databases*, or *KDD* (term originated in the artificial intelligence (AI) research field). Alternatively, others view data mining simply as an essential step in the process of knowledge discovery in databases [Piatesky-Shapiro and Frawley, 1991]. The KDD process (see Fig. 1) consists of an iterative sequence of the following steps:

1. **Data cleaning** (to remove noise and inconsistent data);

2. **Data integration** (where multiple data sources may be combined);

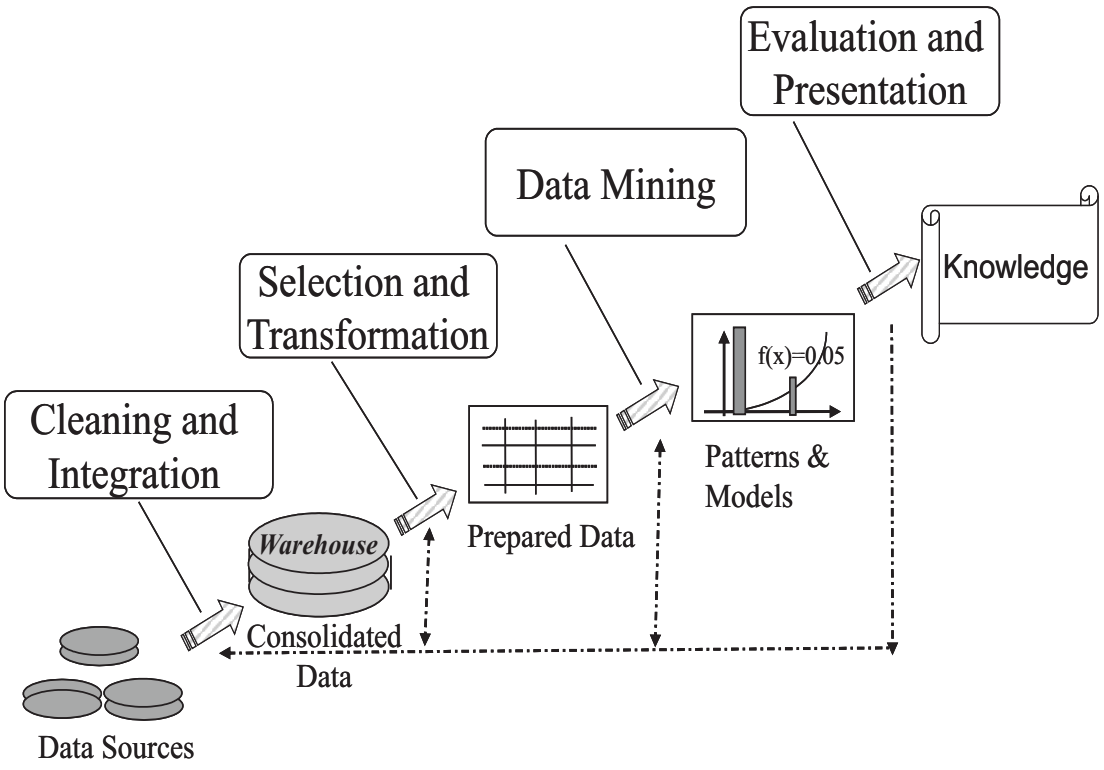


Figure 1: Data mining as a step in the process of knowledge discovery

3. **Data selection** (where data relevant to the analysis task are retrieved from the database);

4. **Data transformation** (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance);

5. **Data mining** (an essential process where intelligent methods are applied in order to extract data patterns);

6. **Pattern evaluation** (to identify the truly interesting patterns representing knowledge based on some interestingness measures);

7. **Knowledge presentation** (where visualization and knowledge representation techniques are used to present the mined knowledge to the user);

To construct and evaluate specific data mining algorithms, a number of principles can be applied:

- determine the nature and structure of the representation to be used;
- decide how to quantify and compare how well different representations fit the data (that is, choosing a score function);
- choose an algorithmic process to optimize the score function; and
- decide what principles of data management are required to implement the algorithms efficiently.

Data mining involves an integration of techniques from multiple disciplines such as database technology [Han et al., 1996], statistics [Hosking et al., 1997], machine learning [Michalski et al., 1998], high-performance computing [Maniatty and Zaki., 2000], pattern recognition [Han et al., 1998], neural network [Bigus, 1996], data visualization [Card et al., 1999], information retrieval [Chakrabarti et al., 1998], image and signal processing [Subrahmanian, 1998] and spatial/temporal data analysis [Miller and Han, 2000]. By performing data mining, interesting knowledge, regularities, or high-level information can be extracted from databases and viewed or browsed from different angles. The discovered knowledge can be applied to decision making, process control, information management and query processing. Therefore, data mining is considered one of the most promising interdisciplinary developments in the information industry.

1.2 Contributions and Structure

In many applications, the data of interest comprise multiple sequences that evolve over time. Examples include financial market data, currency exchange rates, network traffic data, signals from biomedical sources, etc. Although traditional time series techniques can sometimes produce accurate results, few can provide easily understandable results. However, a drastically increasing number of users with a limited statistical background would like to use these tools. At the same time, we have a number of tools developed by researchers in the field of artificial intelligence, which produce understandable rules.

However, they have to use ad-hoc, domain-specific techniques for transforming the time series to a "learner-friendly" representation. These techniques fail to take into account both the special problems and special heuristics applicable to temporal data and therefore often result in unreadable concept description.

As a possible solution to overcome these problems, we proposed to develop a methodology that integrates techniques developed both in the field of machine learning and in the field of statistics. The machine learning approach is used to extract symbolic knowledge and the statistical approach is used to perform numerical analysis of the raw data. The overall goal consists in developing a series of methods able to extract/generate temporal rules, having the following characteristics:

- Contain explicitly a temporal (or at least a sequential) dimension.
- Capture the correlation between time series.
- Predict/forecast values/shapes/behavior of sequences (denoted events)
- Present a structure readable and comprehensible by a human expert.

From a data mining perspective, our methodology can be viewed as belonging to the domain of temporal data mining, which focuses on the discovery of causal relationships among events that may be ordered in time and may be causally related [Roddick and Spiliopoulou, 2002, Antunes and Oliveiro, 2001]. 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. The contributions in this domain encompass the discovery of temporal rules, of sequences and of patterns. However, in many respects this is just a terminological heterogeneity among researchers that are, nevertheless, addressing the same problem, albeit from different starting points and domains.

The overall structure of the thesis is made up of two major parts: the algorithmic viewpoint of the methodology, which presents the main applications/tools from raw data to

temporal rules, and the theoretical foundation of the methodology, which permits an abstract view on temporal rules. Looking at the thesis from a chapter to chapter viewpoint, it proceeds as follows: Chapter 2 contains a detailed description of the two main steps of the proposed methodology (see Cotofrei and Stoffel [2002d]). These steps may be structured in the following way:

- Transforming sequential raw data into sequences of events: Roughly speaking, an event can be regarded as a labelled sequence of points extracted from the raw data and characterized by a finite set of predefined features. The features describing the different events are extracted using statistical methods.

- Inferring temporal rules: An induction process is applied, using sets of events as training sets, to obtain one (or more) classification trees. Then temporal rules are extracted from these classification trees.

The use of classification trees to generate temporal rules is a novel idea, even if similar, but limited approaches may be founded in Kadous [1999] or in Karimi and Hamilton [2000]. Our contribution consisted in the definition of a parameterized procedure for the specification of the training set, which allows the capture of the temporal dimension, even if "time", as attribute, is not processed during the classification tree induction. The concept of event, as we defined it (type and features), permits also the application of the methodology in a non-supervised mode.

In Chapter 3 we extend our methodology with an innovative formalism based on first-order temporal logic, which permits an abstract view on temporal rules (see Cotofrei and Stoffel [2002a,b,c]). The theoretical framework we proposed permits to define the main concepts used in temporal data mining (*event, temporal rule, constraint, support, confidence*) in a formal way. The notion of *consistent linear time structure* allows us to introduce the notion of *general interpretation*. These two important concept, extensively developed in the next chapters, express the fact that the structure on which the first-order

temporal logic is defined represents a homogenous model (let call it M) and therefore the conclusions (or inferences) based on a finite model $\tilde{M}$ for M are consistent. As far as the author has been able to ascertain, these concepts have not been previously formulated. A series of lemmas and corollaries concerning the properties of the concept of *support* for different types of formulae are proved and a final section, showing the connection between the methodology and the formalism, closes the chapter.

Chapter 4 contains an extension of the formalism to include the concept of time granularity (see Cotofrei and Stoffel [2005 (to appear]). We define the process from which a given structure of time granules μ (called *temporal type*) induces a first-order linear time structure M_μ on the basic (or absolute) linear time structure M. The major change for the temporal logic based on M_μ is at the semantic level: for a formula p, the interpretation does not assign a meaning of truth (one of the values $\{true, false\}$), but a degree of truth (a real value from $[0, 1]$). This kind of interpretation is a concrete application of the concept of general interpretation. Consequently, we can give an answer to the following question: if temporal type μ is *finer than* temporal type ν, what is the relationship between the interpretations of the same formula p in the linear time structures M_μ and M_ν. Our contribution is reflected in a theorem proving that only the time independent information may be transferred without loss between worlds with different granularities. By extending the concept of consistency to granular time structure M_μ, we succeeded to demonstrate that this property is inherited from the basic time structure M if the temporal type μ satisfies certain conditions. The most important consequence of this result, with profound implications in practice, is that the confidence of a temporal rule does not depend on the granularity of time. We also study the variation process for the set of satisfiable events (degree of truth equal one) during the transition between two time structures with different granularities. By an extension at the syntactic and semantic level we define a mechanism for aggregation of events, that reflects the following intuitive phenomenon: in a coarser world, not all events inherited from a finer world are satisfied, but in exchange there are new events which

become satisfiable.

In the next chapter we are concerned with a fundamental characteristic of the knowledge: the uncertainty. If the uncertainty is an irreducible aspect of our knowledge about the world, the probability is the most well-understood and widely applied logic for computational scientific reasoning under uncertainty. Therefore, we attach a probabilistic model (more precisely, a stochastic process) to our formalism to obtain a probabilistic first-order temporal logic. In the literature, the problem of the connection between the joint distribution law and the semantics, in the framework of a probabilistic first-order logic, was not studied. Even if the independence restriction for the stochastic process is sufficient to deduce the property of consistency for the temporal structure M, it is not suitable for modelling temporal data mining. A temporal rule expresses the intrinsic dependence between successive events in time. By using advanced theorems from the stochastic limit theory, we succeeded to prove that a certain amount of dependence (called near-epoch dependence) is sufficient as well to induce the property of consistency (showed to be equivalent with the strong law of large numbers). Because we use in this chapter many specialized terms, concepts and theorems from the probability/statistics theory, an overview of these terms is provided in appendix A.

Chapter 6 expresses the fact that defining a formalism stating from a concrete methodology is not a unidirectional way. It is also possible that inferences made at a formal level (i.e. based on an abstract model) to be translated in a practical application. Our formalism allows the application of an inference phase in which higher order temporal rules (called temporal meta-rules) are inferred from local temporal rules. The process of inferring temporal meta-rules is related to a new approach in data mining, called *higher order mining* (see Spiliopoulou and Roddick [2000]), i.e. mining from the results of previous mining runs. According to this approach, the rules generated by the first induction process are first order rules and those generated by the second inference process (i.e temporal meta-rules) are higher order rules. Our formalism does not impose which methodology must be used

to discover first order rules. As long as these rules may be expressed according to the same formal definition, the strategy (here including algorithms, criterions, statistical methods), developed to infer temporal meta-rules may be applied (see [Cotofrei and Stoffel, 2003, 2004]).

Finally, the last chapter draws some general conclusions about the theoretical and practical consequences of the logical model and emphasize what we consider an important and still open problem of our formalism. We also want to mention that the most important results of this thesis will be published as a chapter in the book *Foundation of Data Mining and Knowledge Extraction* (Lin and Liau [2005 (to appear]) and that our methodology was patented in 2004 by US Patent Office under the name "Sequence Miner".

1.3 Related Work

The main tasks concerning the information extraction from temporal data and on which the researchers concentrated their efforts over the last years may be divided in several directions.

- **Similarity/Pattern Querying** The main problem addressed by this body of research concerns the measure of similarity between two sequences or sub-sequences respectively. Different models of similarity were proposed, based on different similarity measures. The Euclidean metric and an indexing method based on Discrete Fourier Transformation were used for matching full sequences [Agrawal et al., 1993] as well as for sub-pattern matching [Faloutsos et al., 1994]. This technique has been extended to allow shift and scaling in the time series [Goldin and Kanellakis, 1995]. To overcome the sensibility of Euclidean metric to outliers, other measures, e.g. the envelope ($|X_i - Y_i| < \epsilon$), were proposed. Different methods (e.g. window stitching) were developed to allow matching similar series despite gaps, translation and scaling [Agrawal and Srikant, 1995, Das et al., 1997, Faloutsos and et al., 1997]. Dynamic time warping based matching is another popular technique in the context of speech

processing [Sakoe and Chiba, 1978], sequence comparison [Erickson and Sellers, 1983], shape matching [McConnell, 1991] and time series data pattern matching [Berndt and Clifford, 1994, Keogh and Pazzani, 1999, Keogh et al., 2002b]. Efficient indexing techniques for time sequences using this metric were developed [Yi et al., 1998]. For all similarity search methods, there is a heavy reliance on the user-specified tolerance ϵ. The quality of the results and the performance of the algorithms are intrinsically tied to this subjective parameter, which is a real usability issue.

- **Clustering/Classification**. In this direction, researchers mainly concentrate on optimal algorithms for clustering/classifying sub-sequences of time series into groups/classes of similar sub-sequences. A first technique for temporal classification is the Hidden Markov Model [Rabiner and Juang, 1986, Lin et al., 2001]. It turned out to be very useful in speech recognition (it is the basis for a lot of commercial systems). Another recent development for temporal classification tasks is Dynamic Bayes Networks (DBNs) [Zweig and Russell, 1998, Friedman et al., 1998], which improve HMMs by allowing a more complex representation of the state space. A technique that has gained some use is Recurrent Neural Networks [Bengio, 1996, Guimares, 2000]. This method utilizes a normal feed-forward neural network, but introduces a "context layer" that is feed back to the hidden layer one time-step later and this allows for retention of some state information. Some work has also been completed on signals with high-level event sequence description where the temporal information is represented as a set of time-stamped events with parameters. Applications of this method can be found in network traffic analysis systems [Mannila et al., 1997] or network failure analysis systems [Oates et al., 1998]. Recently, the machine learning approaches opened new directions. A system for supervised classification on univariate signals using piecewise polynomial modelling was developed in Mangaranis [1997] and a technique for agglomerative clustering of univariate time series based on enhancing the time series with a line segment representation was studied in

Keogh and Pazzani [1998].

- **Pattern finding/Prediction** These methods, concerning the search for periodicity patterns in time series databases, may be divided into two groups: those that search full periodic patterns (where every point contributes, precisely or approximately, to the cyclic behavior of the time series) and those that search partial periodic patterns which specify the behavior at some but not all points in time. For full periodicity search there is a rich collection of statistic methods, like FFT [Loether and McTavish, 1993]. For partial periodicity search, different algorithms were developed, which explore properties related to partial periodicity such as the a-priori property and the max-subpattern-hit-set property [Han et al., 1998]. New concepts of partial periodicity were introduced, like segment-wise or point-wise periodicity and methods for mining these kinds of patterns were developed [Han et al., 1999].

- **Causal and Temporal Rules** Besides these, some researches were devoted to the extraction of explicit rules from time series. 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 visit [Long et al., 1991, Chen and Petrounias, 2000]. Inter-transaction association rules, proposed by Lu et al. [1998], are implication rules whose two sides are totally ordered episodes with time-interval restriction on the events. In Bettini et al. [1998b] a generalization of these rules is developed, having episodes with independent time-interval restrictions on the left-hand and right-hand side. Cyclic association rules were considered in Ozden et al. [1998], adaptive methods for finding rules whose conditions refer to patterns in time series were described in Das et al. [1998], Tsumoto [1999], Hoppner [2001], and a general architecture for classification and extraction of comprehensible rules (or descriptions) was proposed in Kadous [1999].

CHAPTER II

THE METHODOLOGY

The approaches concerning the information extraction from temporal/sequential data, described in Section 1.3, have mainly two shortcomings, which we tried to overcome.

The first problem is the type of knowledge inferred by the systems, often not easily understood by a human user. In a wide range of applications, (e.g. almost all decision making processes) it is unacceptable to produce rules that are not understandable by an end user. Therefore, we decided to develop inference methods that produce knowledge represented in the form of general Horn clauses, which are at least comprehensible for a moderately sophisticated user. In the fourth approach, (*Causal and Temporal Rules*), a similar representation is used. However, the rules inferred by these systems have a more restricted form than the rules we propose.

The second problem consists in the number of time series investigated during the inference process. Almost all methods mentioned above are based on one-dimensional data, i.e. they are restricted to one time series. The methods we propose are able to handle multi-dimensional data.

Two of the most important scientific communities which brought relevant contributions to data analysis (the statisticians and database researchers) chose two different ways: statisticians concentrated on the continuous aspect of the data, the large majority of statistical models being continuous models, whereas the database community concentrated much more on the discrete aspects, and in consequence, on discrete models. For our methodology, we adopt a mixture of these two approaches, which gives a better description of the reality of data and which generally allows us to benefit from the advantages of both approaches.

The two main steps of the methodology for temporal rules extraction are structured in the following way:

1. Transforming sequential raw data into sequences of events: Roughly speaking, an event can be seen as a labelled sequence of points extracted from the raw data and characterized by a finite set of predefined features. The features describing the different events are extracted using statistical methods.

2. Inferring temporal rules: We apply a first induction process, using sets of events as training sets, to obtain several classification trees. Local temporal rules are then extracted from these classification trees and a final inference process will generate the set of temporal meta-rules.

2.1 Phase One

The procedure that creates a database of events from the initial raw data can be divided into two steps: time series discretisation, which extracts the discrete aspect, and global feature calculation, which captures the continuous aspect.

- *Time series discretisation*. During this step, the sequence of raw data is "translated" into a sequence of discrete symbols. By an abuse of language, an event means a subsequence having a particular shape. In the literature, different methods were proposed for the problem of discretizing time series using a finite alphabet (window's clustering method [Das et al., 1998], ideal prototype template [Keogh and Pazzani, 1998], scale-space filtering [Hoppner, 2002]). In the window's clustering method, a window of width w on the sequence $s = (x_1, x_2, \ldots, x_n)$ can be defined as a contiguous subsequence $s_i = (x_i, x_{i+1}, \ldots, x_{i+w-1})$. One extracts from s all windows (subsequences) of width w, and denotes $W(s)$ the set $\{s_i : i = 1 \ldots n-w+1\}$. Assuming we define a distance $d(s_i, s_j)$ between any two subsequences s_i and s_j of width w, this distance can be used to cluster the set of all subsequences from $W(s)$ into k clusters $C_1, C_2, \ldots, C_k$.

For each cluster C_h a symbol a_h is introduced and the discretised version $D(s)$ of the sequence s will be expressed using the alphabet $\sum = \{a_1, \ldots, a_k\}$. The sequence $D(s)$ is obtained by finding for each subsequence s_i the corresponding cluster $C_{j(i)}$ such that $s_i \in C_{j(i)}$ and by substituting the subsequence with the corresponding symbol $a_{j(i)}$. Thus $D(s) = (a_1, a_2, \ldots, a_{n-w+1})$.

In Cotofrei and Stoffel [2002d] we adopted a simpler solution, which implies also an easier implementation. Starting with the same sequence s, we calculate the sequence of the differences between two consecutive values. The sorted list of these differences is then divided into k intervals, such that each interval contains a percentage $1/k$ of values (in a statistical language, we calculated the $1/k$-quantile from the population of differences). Each interval will then be labelled using a symbol (a_i for the i^{th} interval). Therefore, the discretisation version of s, $D(s)$, is simply the "translation" of the sequence of differences into the sequence of corresponding symbols. The parameter k controls the degree of discretisation: a bigger k means a bigger number of events and consequently, less understandable rules. However, a smaller k means a rougher description of the data and finally, simpler rules but without significance.

If the sequence of differences $x_{i+1} - x_i$ is firstly normalized, and the quantile of the normal distribution are used, we obtain the discretisation algorithm proposed by Keogh et al. [2002a]. Another similar proposal (see Huang and Yu [1999]) suggests the segmentation of a sequence by computing the change ratio from one point to the following one, and representing all consecutive points with equal change ratios by a unique segment. After this partition, each segment is represented by a symbol and the sequence is represented as a string of symbols.

The advantage of these methods is that the time series is partitioned in a natural way, depending on its values. However, the symbols of the alphabet are usually chosen externally which means that they are imposed by the user, who has to know the most suitable symbols, or they are established in an artificial way. But the biggest

weakness of these methods which use a fixed length window is their sensibility to noise. Therefore, the scale-space filtering method, which finds the boundaries of the subsequences having a persistent behavior over multiple degree of smoothing, seems to be more appropriate and must be considered as a first compulsory pre-processing phase.

- *Global feature calculation.* During this step, one extracts various features from each sub-sequence as a whole. Typical global features include global maxima, global minima, means and standard deviation of the values of the sequence as well as the value of some specific point of the sequence, such as the value of the first or of the last point. Of course, it is possible that specific events will demand specific features, necessary for their description (e.g. the slope of the best-fitting line or the second real Fourier coefficient). The optimal set of global features is hard to be defined in advance, but as long as these features are simple descriptive statistics, they can be easily added or removed from the process.

EXAMPLE 2.1 Consider a database containing daily price variations of a given stock. After the application of the first phase we obtain an ordered sequence of events. Each event has the form (*name*, v_1, v_2), where the name is one of the strings {*peak, flat, valley*} – we are interested only in three kinds of shapes - and v_1, v_2 represent the mean, respectively, the standard error – we chose only two features as determinant for the event. The statistics are calculated using daily prices, supposed to be subsequences of length $w = 12$.

2.2 Phase Two

During the second phase, we create a set of temporal rules inferred from the database of events, obtained in phase one. Two important steps can be defined here:

- *First induction process*. During this step, different classification trees are constructed

using the event database as training database. From each classification tree, the corresponding set of temporal rules is extracted.

- *Second inference process*. During this step, a strategy derived from the higher order miner approach is applied on the previously inferred temporal rules sets to obtain the final set of temporal meta-rules.

2.2.1 First induction process.

There are different approaches for extracting rules from a set of events. Association Rules [Chen and Petrounias, 2000], Inductive Logic Programming [Rodriguez et al., 2000], Classification Trees [Karimi and Hamilton, 2000] are the most popular ones. For our methodology we selected the *classification tree approach.* It is a powerful tool used to predict memberships of cases or objects in the classes of a categorical dependent variable from their measurements on one or more predictor variables (or attributes). A variety of classification tree programs has been developed and we may mention QUEST [Loh and Shih, 1997], CART [Breiman et al., 1984], FACT [Loh and Vanichsetakul, 1988], THAID [Morgan and Messenger, 1973], CHAID [Kass, 1980] and last, but not least, C4.5 [Quinlan, 1993]. To justify our option (the C4.5 approach), a brief description of the algorithmic aspects involved in the process of "building" classification trees is necessary [StatSoft, Inc, 2004].

2.2.1.1 Classification Trees

A classification tree is constructed by recursively partitioning a learning sample of data in which the class and the values of the predictor variables for each case are known. Each partition is represented by a node in the tree. The classification trees readily lend themselves to being displayed graphically, helping to make them easier to interpret than they would be if only a strict numerical interpretation were possible.

The most important characteristics of a classification tree are the *hierarchical nature*

and the *flexibility*. The first characteristics means that the relationship of a leaf to the tree on which it grows can be described by the hierarchy of splits of branches (starting from the root) leading to the last branch from which the leaf hangs. This contrasts with the simultaneous nature of other classification tools, like discriminant analysis. The second characteristic reflects the ability of classification trees to examine the effects of the predictor variables one at a time, rather than just all at once. The process of constructing decision trees can be divided into the following four steps:

1. **Specifying the criteria for predictive accuracy**. The goal of classification tree analysis, simply stated, is to obtain the most accurate prediction possible. To solve the problem of defining *predictive accuracy*, the problem is "stood on its head," and the most accurate prediction is operationally defined as the prediction with the minimum costs. The notion of costs was developed as a way to generalize to a broader range of prediction situations the idea that the best prediction has the lowest misclassification rate. *Priors*, or *a priori probabilities*, specify how likely it is, without using any prior knowledge of the values for the predictor variables in the model, that a case or object will fall into one of the classes. In most cases, minimizing costs corresponds to minimizing the proportion of misclassified cases when priors are taken to be proportional to the class sizes and when misclassification costs are taken to be equal for every class. The tree resulting by applying the C4.5 algorithm is constructed to minimize the observed error rate, using equal priors. This criterion seems to be satisfactory in the frame of sequential data and furthermore has the advantage to not favour certain events.

2. **Selecting splits.** The second basic step in classification tree construction is to select the splits on the predictor variables used to predict membership of the classes of the dependent variables for the cases or objects in the analysis. These splits are selected one at the time, starting with the split at the root node, and continuing with splits of resulting child nodes until splitting stops, and the child nodes, which have not been

split, become terminal nodes. The three most popular *split selection methods* are:

- *Discriminant-based univariate splits* [Loh and Shih, 1997]. The first step is to determine the best terminal node to split in the current tree, and which predictor variable to use to perform the split. For each terminal node, p-values are computed for tests of the significance of the relationship of class membership with the levels of each predictor variable. The tests used most often are the Chi-square test of independence, for categorical predictors, and the ANOVA F-test for ordered predictors. The predictor variable with the minimum p-value is selected. The second step consists in applying the 2-means clustering algorithm of Hartigan and Wong [1979] to create two "super classes" for the classes presented in the node. For ordered predictor, the two roots for a quadratic equation describing the difference in the means of the "super classes" are found and used to compute the value for the split. This approach is well suited for our data (events and global features) as it is able to treat continuous and discrete attributes at the same tree.

- *Discriminant-based linear combination splits*. This method works by treating the continuous predictors from which linear combinations are formed in a manner that is similar to the way categorical predictors are treated in the previous method. Singular value decomposition methods are used to transform the continuous predictors into a new set of non-redundant predictors. The procedures for creating "super classes" and finding the split closest to a "super class" mean are then applied, and the results are "mapped back" onto the original continuous predictors and represented as a univariate split on a linear combination of predictor variables. This approach, inheriting the advantages of the first splitting method, uses a larger set of possible splits thus reducing the error rate of the tree, but, at the same time, increasing the computational costs.

- *CART-style exhaustive search for univariate splits*. With this method, all possible splits for each predictor variable at each node are examined to find the split producing the largest improvement in *goodness of fit* (or equivalently, the largest reduction in lack of fit). There exists different ways of measuring goodness of fit. The *Gini measure of node impurity* [Breiman et al., 1984] is a measure that reaches the value zero when only one class is present at a node and it is used in CART algorithm. Other two indices are the *Chi-square measure*, which is similar to Bartlett's Chi-square and the *G-square measure*, which is similar to the maximum-likelihood Chi-square. Adopting the same approach, the C4.5 algorithm uses the *gain criterion* as goodness of fit. If S is any set of cases, let $freq(C_i, S)$ stands for the number of cases in S that belong to class C_i. The entropy of the set S (or the average amount of information needed to identify the class of a case in S) is the sum:

$$\text{info}(S) = -\sum_{i=1}^{k} \frac{freq(C_i, S)}{|S|} \times \log_2\left(\frac{freq(C_i, S)}{|S|}\right).$$

After S is partitioned in accordance with n outcomes of a test X, a similar measurement is the sum:

$$\text{info}_X(S) = \sum_{i=1}^{n} \frac{|S_i|}{|S|} \times \text{info}(S_i).$$

The quantity $gain(X) = \text{info}(S) - \text{info}_X(S)$ measures the information that is gained by partitioning S in accordance with the test X. The gain criterion selects a test to maximize this information gain (which is also known as the mutual information between the test X and the class). The bias inherent in the gain criterion can be rectified by a kind of normalization in which the apparent gain attributable to the test with many outcomes is adjusted. By analogy with the definition of info(S), one defines

$$\text{split\_info}(X) = -\sum_{i=1}^{n} \frac{|S_i|}{|S|} \times \log_2\left(\frac{|S_i|}{|S|}\right),$$

representing the potential information generated by dividing S into n subsets. Then, the quantity $gain\_ratio(X) = gain(X)/\text{split\_info}(X)$ expresses the proportion of information generated by the split. The *gain ratio criterion* selects a test to maximize the ratio above, subject to the constraint that the information gain must be large – at least as great as the average gain over all tests examined. To successively create the partitions, the C4.5 algorithm uses two forms of tests in each node: a standard test for discrete attributes, with one outcome ($A = x$) for each possible value x of the attribute A, and a binary test, for continuous attributes, with outcomes $A \leq z$ and $A > z$, where z is a threshold value.

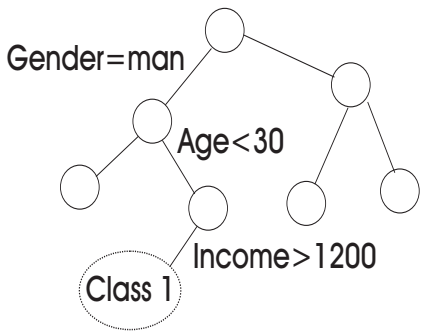
3. **Determining when to stop splitting**. There are two options for controlling when splitting stops:

   - *Minimum n*: the spitting process continues until all terminal nodes are pure or contain no more than a specified minimum number of cases or objects (it is the standard criterion chosen by C4.5 algorithm) and
   - *Fraction of objects*: the spitting process continues until all terminal nodes are pure or contain no more cases than a specified minimum fraction of the sizes of one or more classes (non feasible because of the absence of a priori information on the size of the classes).

4. **Selecting the "Right-Sized" Tree**. Usually we are not looking for a classification tree that classifies perfectly in the learning samples, but one which is expected to predict equally well in the test samples. There are two strategies that can be adopted here. One strategy is to grow the tree to just the right size, where the right size is determined by the user, from knowledge from previous research, diagnostic information from previous analysis, or even intuition. To obtain diagnostic information that determine the reasonableness of the choice of size for the tree, different options of cross-validation may be used. The second strategy consists in growing a tree until

it classifies (almost) perfectly the training set and then pruning at the "right-size". This approach supposes that it is possible to predict the error rate of a tree and of its subtrees (including leaves). Such a technique, called *minimal cost-complexity pruning* and developed by Breiman et al. [1984] considers the predicted error rate as the weighted sum of tree complexity and its error on the training cases, with the separate cases used primarily to determine an appropriate weighting. The C4.5 algorithm uses another technique, called *pessimistic pruning*, that uses only the training set from which the tree was built. The predicted error rate in a leaf is estimated as the upper confidence limit for the probability of error (E/N, where E is the number of errors and N is the number of covered training cases) multiplied by N. In our case, the lack of a priori knowledge about the "right size" of the tree, as demanded by the first strategy, makes the approach used by the C4.5 algorithm the better choice for us.

In any classification tree, the conditions that must be satisfied when a case is classified by a leaf (or terminal node) can be found by tracing all the test outcomes along the path from the root to that leaf. In the tree of Figure 2, the **Class 1** leaf is associated with the outcomes **Gender=man**, **Age<30** and **Income>1200**. This particular path may be expressed as a rule representing a conjunction of tests outcomes (here using the natural language): "If a person is a man and his age is less than 30 and he has an income greater than 1200 then the class is Class 1."



If (Gender=man) and (Age<30) and (Income>1200) then Class 1

Figure 2: Rule corresponding to a path from the root to the leave "Class 1", expressed as a conjunction of three outcome tests implying each a different attribute

2.2.1.2 The Training Set Construction

Before applying the decision tree algorithm to a database of events, an important problem has to be solved: establishing the training set. An n-tuple in the training set contains $n-1$ values of the predictor variables (or attributes) and one value of the categorical dependent variable, which represents the class. There are two different approaches on how the sequence that represents the classification (the values of the categorical dependent variable) is obtained. In a supervised methodology, an expert gives this sequence. The situation becomes more difficult when there is no prior knowledge about the possible classifications. Suppose, following the example 2.1, that we are interested in testing if a given stock value depends on other stock values. As the dependent variable (the stock price) is not categorical, it cannot represent a valid classification used to create a classification tree. The solution is to use the sequence of the names of events extracted from the continuous time series as a sequence of classes.

Let us suppose we have k sequences, $q_1, q_2, \ldots, q_k$, representing the predictor variables. Each q_{ij}, $i = 1, \ldots, k$, $j = 1, \ldots, n$ is the name of an event (*Remark:* we consider a simplified case, with no feature as predictor variable, but without influence on the following rationing). We also have a sequence $q_c = q_{c1}, \ldots, q_{cn}$ representing the classification. The training set will be constructed using a procedure depending on three parameters. The first, t_0, represents a time instant considered as *present time*. Practically, the first tuple contains the class q_{ct_0} and there is no tuple in the training set containing an event that starts after time t_0. The second, t_p, represents a time interval and controls the further back in time class $q_{c(t_0-t_p)}$ included in the training set. Consequently, the number of tuples in the training set is $t_p + 1$. The third parameter, h, controls the influence of the past events $q_{i(t-1)}, \ldots, q_{i(t-h)}$ on the actual event q_{it}. This parameter ($h$istory) reflects the idea that the class q_{ct} depends not only on the events at time t, but also on the events occurred before time t. Finally, each tuple contains $k(h+1)$ events (or values for $k(h+1)$ attributes, in the terminology of classification trees) and one class value (see Fig. 3). The first tuple

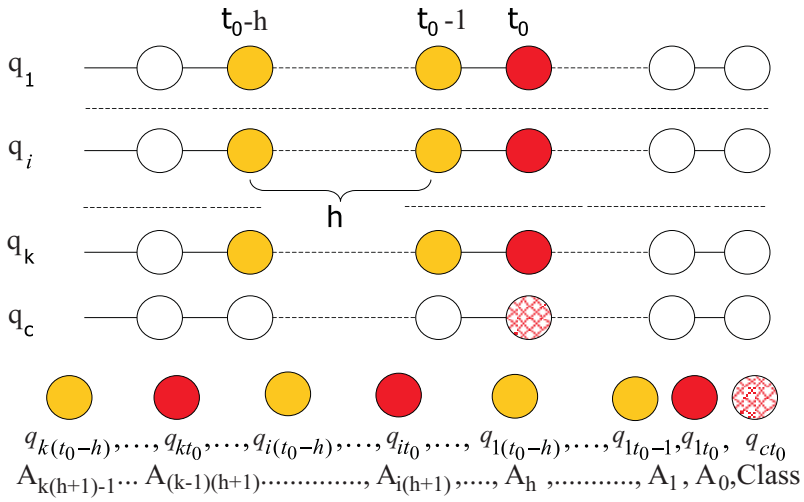


Figure 3: Graphical representation of the first tuple and the list of corresponding attributes

is $q_{ct_0}, q_{1t_0}, \ldots, q_{1(t_0-h)}, \ldots, q_{k(t_0-h)}$ and the last $q_{c(t_0-t_p)}, q_{1(t_0-t_p)}, \ldots, q_{k(t_0-t_p-h)}$. To adopt this particular strategy for the construction of the training set, we made an assumption: the events q_{ij}, $i = 1, \ldots, k$, j a fixed value, occur all at the same time instant. The same assumption allows us to solve another implementation problem: the time information is not processed during the classification tree construction, (time is not a predictor variable), but the temporal dimension must be captured by the temporal rules. The solution we chose to encode the temporal information is to create a map between the index of the attributes (or predictor variables) and the order in the time of the events. The $k(h+1)$ attributes are indexed as $\{A_0, A_1, \ldots, A_h, \ldots, A_{2h}, \ldots, A_{k(h+1)-1}\}$. As we can see in Fig. 3, in each tuple the values of the attributes from the set $\{A_0, A_{h+1}, \ldots, A_{(k-1)(h+1)}\}$ represent events which occur at the same time moment as the class event, those of the set $\{A_1, A_{h+2}, \ldots, A_{(k-1)(h+1)+1}\}$ represent events which occur one time moment before the same class event, and so on. Let be $\{i_0, \ldots, i_m\}$ the set of indexes of the attributes that appear in the body of the rule (i.e. the rule has the form

$$If\ (A_{i_0} = e_0)\ and\ (A_{i_1} = e_1)\ and\ \ldots\ and\ (A_{i_m} = e_m)\ Then\ Class\ e,$$

where e_{i_j} are events from the sequences $\{q_1, \ldots, q_k\}$ and e is an event from the sequence q_c. If t represents the time instant when the event in the head of the rule occurs, then an

event from the rule's body, corresponding to the attribute A_{i_j}, occurred at time $t - \bar{i}_j$, where $\bar{i}_j$ means i *modulo* $(h + 1)$.

2.2.1.3 A Synthetic Experiment

To illustrate the importance of the parameter h for the training set construction, and to exemplify the procedure for adding the temporal dimension to a rule generated by C4.5 algorithm, a simulation study, using a synthetic database, is made. The predictive variables are represented by three time series and, choosing a supervised situation, we dispose also of a sequence of class labels, representing the classification. Each series contain 500 values generated randomly, in a first phase, between 0 and 30. In a second phase, we modify some values in order to find, from time to time, decreasing sequences of length 5 (denoted *decrease*) in the first series, sequences of 5 almost equal values (denoted *stable*) in the second series and increasing sequences of length 5 (denoted *increase*) in the third series. As we may observe in Figure 4, where only the firsts 32 values of the three time series were represented graphically, such particular sequences start at time $t = 8$ and $t = 24$. If a *decrease* sequence starts at time t in the first time series, a *stable* in the second and an *increase* in the third series, then at time $t + 4$ the expert sets, in the classification sequence, the label **1**. For all other situations, the label class will be **0**. There are 39 class labelled **1** among all 500 cases, which represents 7.8% of all cases. The reason for this particular

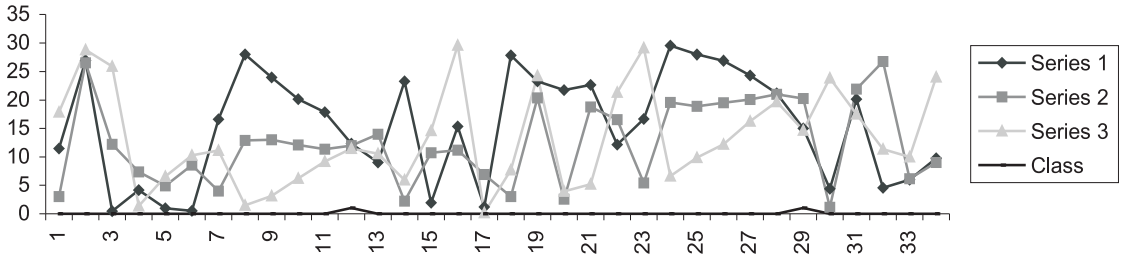


Fig. 1

Figure 4: Graphical representation of the first 32 values of predictive variables (Series 1-3) and of dependent variable (Class)

labelling process is that we want a classification that is independent of the numerical values of the series, but depends on some particular behaviors of the time series. A classification tree which would be constructed using only numerical values of the series in the training set would have a high error rate, due to the random character of the data.

During the discretisation phase we use a very simple approach, which consists in defining three intervals, $[-30, -1)$, $[-1, 1]$ and $(1, 30]$ and encoding them with the letters {**a**, **b**, **c**}. Each sequence of length two, $(s_{ji}, s_{j(i+1)})$, $j = 1..3$, $i = 1..499$, is thus labelled depending on those interval the difference $s_{ji} - s_{j(i+1)}$ falls into. In this way a sequence *decrease* will be labelled with the word **aaaa**, a sequence *stable* as **bbbb** and a sequence *increase* as **cccc**.

Different trees are constructed with the same parameters $t_0 = 280$ and $t_p = 274$ (the training set contains almost half of the data), but with different h. As we may observe in Fig. 5, as long as the parameter h increases, the observed errors (the number of misclassified cases in the training set) and the prediction errors (the number of misclassified cases when the classification tree is applied to the remaining cases in the database) diminish. This can be explained by the fact that past events influence the predictive accuracy at present time. The more information from the past we take in consideration, the more the classification tree becomes precise. On the other hand one can see that this influence is limited to a time window of length 4 (the classification trees for h greater than four are all identical).

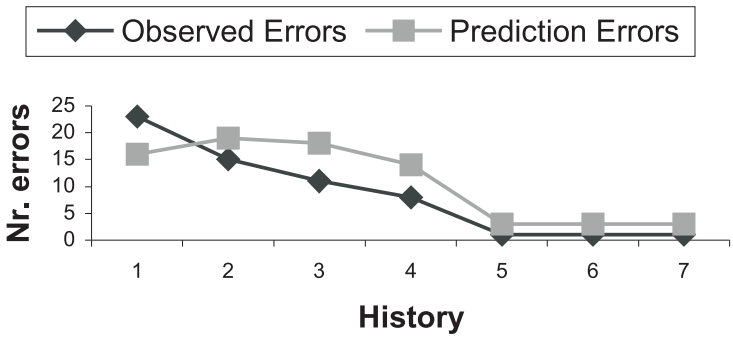


Figure 5: Graphical representation for the variation of observed and predicted errors, for different values of the parameter history

Consider the classification tree based on a training set with $h = 4$. Because the number of predictive series is three, the total number of attributes is $3 \cdot 5 = 15$. The rule implying the class **1**, produced by C4.5 system from this classification tree is:

`A0={a}, A4={a}, A5={b}, A6={b}, A8={b}, A9={b}, A14={c} -> class 1`

having a confidence of 93.8%. It is interesting to observe that the body of the rule does not contain all possible conditions (e.g. `A1 ={a}, A2={a}, A3={a}`, etc.), which means that not all events are significant for the classification. On the other hand we can see that for each time series the event farthest back in time, (`A4`, `A9` and respectively, `A14`) are used by the rule. To add the temporal dimension to the rule, the set of indexes of the attributes $\{0, 4, 5, 6, 8, 9, 14\}$ is transformed, by modulo 5, into the set $\{0, 4, 0, 1, 3, 4, 4\}$. Therefore, by applying the procedure for transforming the ordinary rules into temporal rules we obtain, using a more or less "natural language", the following rule: *If at time moments* $t - 4$ *and* t *the first time series decreases by more than one unit and at time moments* $t - 4$, $t - 3$, $t - 1$ *and* t *the second time series varies by maximum one unit and at time* $t - 4$ *the third time series increases by more than one unit then at time* t *we will have the class* ***1***.

As we already mentioned, in an unsupervised situation we take as the sequence of classes the sequence of event labels, more precisely, of those events considered as dependent from the others. For our database, let us suppose that the events extracted from the third time series are implied by the events extracted from series one and two. We set the

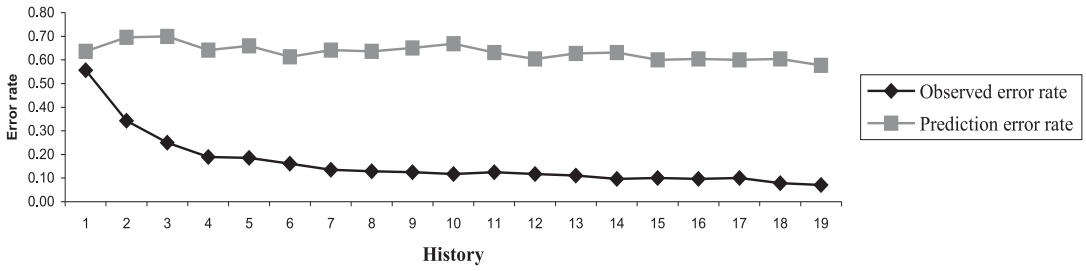


Figure 6: Graphical representation for the variation of observed and predicted errors rates, for different values of the parameter history, when predictor variables and class are independent in time

parameters for the training set procedure as $t_0 = 300$, $t_p = 280$ and h taking values between 0 and 18. Of course, due to the fact that the initial values of time series were generated randomly, we do not expect the C4.5 system to find some "nice" rules implying the corresponding events. Looking at Fig. 6 we can see that the observed error rate goes down even in this "independent context" when the parameter h increases. On the other hand, the prediction error rate remains almost stable, which is obvious because the remaining data events, being independent of the events in the training set (due to the random process generator), have small chances to fit the conditions of the generated rules. This behavior reflects a well-known phenomenon in the context of classification trees, called overfitting.

2.2.2 Second inference process

Different classification trees, constructed from different training sets, generate different sets of temporal rules. 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. The rationale behind the mining of rules is twofold. First, the knowledge discovery process is applied on small sets of rules (which correspond, in our case, to small training sets) 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.

The process that tries to infer temporal meta-rules from sets of local temporal rules is derived from the strategy of rules pruning, used by C4.5 system. Because this strategy may be theoretically applied not only to the rules generated by C4.5 algorithm, but to all rules having the form of a general Horn clause, for which measures like *support* or *confidence* may be defined, the modelling process of our methodology, at an abstract level, looks not only feasible, but absolutely necessary. To obtain an abstract view of temporal rules we propose and develop in the next chapter a formalism based on first-order temporal logic. This formalism allows not only to model the main concepts used by the algorithms

applied during the different steps of the methodology, but also to give a common framework to many of temporal rules extraction techniques, mentioned in the literature. A detailed description of a practical application of the second inference process, in the context of this formalism, is presented in the next to last chapter of the thesis, looping thus the known cycle of research: practice, theory, practice, theory, ...

2.3 Summary

The methodology we developed in this chapter tries to respond to an actual necessity, the need to discover knowledge from data for which the notions of "time" or "sequential order" represent an important issue. We proposed to represent this knowledge in the form of general Horn clauses, a more comprehensible form for a final user without sophisticated statistical background. To obtain what we called "temporal rules", a discretisation phase that extracts "events" from raw data is applied first, followed by an inference phase, which constructs classification trees from these events. The discrete and continuous characteristics of an "event", according to its definition, allow us to use statistical tools as well as techniques from machine learning on the same data.

To capture the correlation between events over time, a specific procedure for the construction of a training set (used later to obtain the classification tree) is proposed. This procedure depends on three parameters, among others, the so-called history that controls the time window of the temporal rules. A particular choice for indexing the attributes in the training set allows us to add the temporal dimension to the rules extracted from the classification trees. The experiments we conducted on a synthetic database showed that the process of event extraction has a major influence on the observed error rate when the classification depends rather on the shape of the time series than on their numerical values. As long as the parameter h increases, the observed error rate decreases, until the time window is large enough to capture (almost) all the relations between events. This dependence between the observed error rates and the parameter h permits us to stop the process of adding

new attributes as soon as the structure of the classification tree becomes stable and thus prevents us from overfitting the tree.

CHAPTER III

FORMALISM OF TEMPORAL RULES

Although there is a rich bibliography concerning formalism for temporal databases, there are very few articles on this topic for temporal data mining. In Al-Naemi [1994], Chen and Petrounias [1998], Malerba et al. [2001], general frameworks for temporal mining are proposed, but usually the researches on causal and temporal rules are more concentrated on the methodological/algorithmic aspect, and less on the theoretical aspect. In this chapter, we extend our methodology with a formalism based on first-order temporal logic, which permits an abstract view on temporal rules. The formalism also allows the application of an inference phase in which higher order temporal rules (called temporal meta-rules) are inferred from local temporal rules, the latter being extracted from different sequences of data. Using this strategy, known in the literature as higher order mining [Spiliopoulou and Roddick, 2000], we can guarantee the scalability (the capacity to handle huge databases) of our methodological approach, by applying statistical and machine learning tools.

3.1 Temporal Domain

Time is ubiquitous in information systems, but the mode of its representation/perception varies in function of the purpose of the analysis [Chomicki and Toman, 1997, Emerson, 1990, Augusto, 2001]. To define a formal system for temporal reasoning, a temporal ontology has to be chosen. Practically, this means to decide how the different aspects of time (the structure, the topology and the mode of reference) should be considered. What option to adopt in each case is not an easy matter because when a choice is made, several aspects of the problem become easier but some others worse.

One thing to decide is if time will be considered as linear, branching, circular or with a

different structure. Each of these characteristics could be represented axiomatically using a first-order language, as in Turner [1984]. While the two first options were favored in the literature, there are certain purposes where the other options must be considered. The most popular way to conceive time is as a line where temporal references could be aligned. The second option is a future-branching structure representing the past as linear and the present as a distinguished point where the future opens as a bunch of possibilities [Emerson, 1990, Haddawy, 1996, Wolper, 1989]. Circular time could be conceived as closed-open, or static-dynamic. The capability to reason over cyclical processes in industrial scenarios could provide reasons to adopt this view of time [Cuckierman and Delgrande, 1998].

Time could also be considered as organized in other ways, e.g. discrete, dense or continuous. This led to the so called topological time because temporal structures could be analyzed under the light of a topology. For most of the problems, the conceptual use of time by an agent as a succession of temporal phenomena organized in a discrete fashion is sufficient. Some problems could be more naturally represented under the hypothesis of a dense or continuous temporal structure like one isomorphic to $\mathbb{Q}$ or $\mathbb{R}$ [Kozen and Parikh, 1981]. It is important also to remember that the adoption of a particular topology leads to important differences in the kind of system to be defined. While $\mathbb{Z}$ can be axiomatized in a first-order theory, $\mathbb{Q}$ and $\mathbb{R}$ cannot. Although usually problems of continuous change lead to think that an $\mathbb{R}$-like structure must be used, some attempts have been made to represent continuous change using discrete structures [Hobbs, 1985, Barber and Moreno, 1997].

The last fundamental source of choice is the way to reference time. The problem of deciding which kind of reference must be considered as more natural has been subject to intense debate. Literature about the philosophy of time provides us with several articles from people sustaining an instant-based view of time [Lin, 1994, McDermott, 1982, Shoham, 1985], while others defend a period-based approach [Russell, 1936, Hamblin, 1972, Kamp, 1979]. Names vary with authors but usually *instants* and *time points* are

used to refer to punctual occurrences and it is usually employed in temporal database applications [Chomicki and Saake, 1998], while *periods* (or *intervals*) are used to talk about durative temporal references and are predominant in AI applications [Allen et al., 1991, Berger and Tuzhilin, 1998, Cohen, 2001, Rodriguez et al., 2000, Kam and Fu, 2000]. The difference induced by the two approaches, at the temporal logic level, is expressed in the set of temporal predicates: they are unary in the first case and binary in the second. Recently some proposals have explored the benefits to allow both kind of references at the same level of importance [Bochman, 1990a,b, Vila, 1994, Augusto, 1998]. It is interesting to see that both references could be defined in terms of each other. For example, periods could be seen as sets of instants or the duration denoted by two instants acting as beginning and ending points, whereas instants could be defined as the meeting point between two periods.

As a remark, it is necessary to remember that the above considered set of possibilities for defining different aspects of a temporal ontology are independent from each other. For example, the decision if the structure is linear or branching does not rule out considering if it is bounded or not or if it is discrete, dense or continuous. Concerning our methodology, we decided to chose a temporal domain represented by linearly ordered discrete instants.

DEFINITION 3.1 *A single-dimensional linearly ordered temporal domain is a structure* $T_P = (T, <)$*, where T is a set of time instants and "<" a linear order on T.*

In the sequel we will further assume that the underlying structure of time is isomorphic to the natural numbers with their usual ordering $(N, <)$. Under this assumption, time

(i) is discrete,

(ii) has an initial moment with no predecessors, and

(iii) is infinite in the future

These properties seem quite appropriate in view of our intended application: reasoning about the behavior of sequence(s) of events. Property (i) reflects the fact that raw data, from

which events are extracted, are stored on digital devices; property (ii) is appropriate since a sequence begins with an initial event; and property (iii) is appropriate since we develop our formalism for reasoning about ongoing, ideally nonterminating behavior.

3.2 Technical Framework

Databases being first-order structures, the first-order logic represents a natural formalism for their description. Consequently, the first-order temporal logic is the support for the formalism of temporal databases. For the purposes of our methodology we consider a restricted first-order temporal language **L** which contains only constant symbols $\{c, d, ..\}$, n-ary ($n \geq 1$) function symbols $\{f, g, ..\}$, variable symbols $\{y_1, y_2, ...\}$, n-ary predicate symbols ($n \geq 1$, i.e. no proposition symbols), the set of relational symbols $\{=, <, \leq, >, \geq\}$, the logical connective $\{\wedge\}$ and a temporal connective of the form X_k, $k \in \mathbf{Z}$, where k strictly positive means *after k time instants*, k strictly negative means *before k time instant* and $k = 0$ means *now*.

3.2.1 Syntax

The syntax of **L** defines terms, atomic formulae and compound formulae. The *terms* of **L** are defined inductively by the following rules:

T1 Each constant is a term.

T2 Each variable is a term.

T3 If $t_1, t_2, \ldots, t_n$ are terms and f is an n-ary function symbol then $f(t_1, \ldots, t_n)$ is a term.

The atomic formulae (or atoms) of **L** are defined by the following rules:

A1 If $t_1, \ldots, t_n$ are terms and P is an n-ary predicate symbol then $P(t_1, \ldots, t_n)$ is an atom.

A2 If t_1, t_2 are terms and ρ is a relational symbol then $t_1 \rho\, t_2$ is an atom (also called relational atom).

Finally, the (compound) formulae of **L** are defined inductively as follow:

F1 Each atomic formula is a formula.

F2 If $p,\ q$ are formulae then $p \wedge q,\ X_k p$ are formulae.

A Horn clause is a formula of the form $B_1 \wedge \cdots \wedge B_m \rightarrow B_{m+1}$ where each B_i is a positive (non-negated) atom. The atoms $B_i,\ i = 1, \ldots, m$ are called implication clauses, whereas B_{m+1} is known as the implicated clause. Syntactically, we cannot express Horn clauses in our language **L** because the logical connective $\rightarrow$ is not defined. However, to allow the description of rules, which formally look like Horn clauses, we introduce a new logical connective, $\mapsto$, which practically will represent a rewrite of the connective $\wedge$. Therefore, a formula in **L** of the form $p \mapsto q$ is syntactically equivalent to the formula $p \wedge q$. When and under what conditions we may use the new connective, is explained in the next definitions.

Definition 3.2 *An event (or temporal atom) is an atom formed by the predicate symbol E followed by a bracketed n-tuple of terms* $(n \geq 1)$ $E(t_1, t_2, \ldots, t_n)$. *The first term of the tuple,* t_1, *is a constant symbol representing the name of the event and all others terms are expressed according to the rule T3* $(t_i = f(t_{i1}, \ldots, t_{ik_i}))$. *A short temporal atom (or the event's head) is the atom* $E(t_1)$.

For each constant symbol t used as an event name, two other constant symbols, *start_t* and *stop_t*, are included in our language **L**. Consequently, for each temporal atom $E(t_1, t_2, \ldots, t_n)$, two temporal atoms, $E(start\_t_1, t_2, \ldots, t_n)$ and $E(stop\_t_1, t_2, \ldots, t_n)$, are defined. It is also important to mention that each term $t_{ij},\ j = 1..k_i$, in the expression $f(t_{i1}, \ldots, t_{ik_i})$, is a constant.

Definition 3.3 *A constraint formula for the event* $E(t_1, t_2, \ldots t_n)$ *is a conjunctive compound formula,* $E(t_1, t_2, \ldots t_n) \wedge C_1 \wedge C_2 \wedge \cdots \wedge C_k$, *where each* C_j *is a relational atom. The first term of* C_j *is one of the terms* t_i, $i = 1 \ldots n$ *and the second term is a constant symbol.*

For a short temporal atom $E(t_1)$, the only constraint formula that is permitted is $E(t_1) \wedge (t_1 = c)$. We denote such a constraint formula as *short constraint formula*.

DEFINITION 3.4 *A temporal rule is a formula of the form* $H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$, *where* H_{m+1} *is a short constraint formula and* H_i *are constraint formulae, prefixed by the temporal connectives* $X_{-k},\ k \leq 0$. *The maximal value of the index k is called the time window of the temporal rule.*

As a consequence of the Definition 3.4, a conjunction of constraint formulae $H_1 \wedge H_2 \wedge \cdots \wedge H_n$, each formula prefixed by temporal connectives $X_{-k},\ k \geq 0$, may be rewritten as $H_{\sigma(1)} \wedge \cdots \wedge H_{\sigma(n-1)} \mapsto H_{\sigma(n)}$, — σ being a permutation of $\{1..n\}$ — only if there is a short constraint formula $H_{\sigma(n)}$ prefixed by X_0.

Remark. The reason for which we did not permit the expression of the implication connective in our language is related to the truth table for a formula $p \rightarrow q$: even if p is false, the formula is still true, which is unacceptable for a temporal rationing of the form *cause*→ *effect*.

Looking at the rules which define terms in L (T1 – T3), we note that there are two types of symbols that are considered to be terms by definition: the constants and the variables. This distinction is lost in the rules defining atomic formulae (A1 – A2) and compound formulae (F1 – F2), which use the generic notion of term. Even though it is not important from a syntactic viewpoint what kind of symbol expresses a term, this distinction becomes vital for the semantics of a linear temporal logic, as we will show later. Therefore, we introduce the notion of *template*, by adding the following syntactic rules:

T2' Each variable is a template term.

T3' If at least one of the terms $t_1, t_2, \ldots, t_n$ is a template term and f is an n-ary function symbol then $f(t_1, \ldots, t_n)$ is a template term.

A1' If at least one of the terms $t_1, \ldots, t_n$ is a template term and P is an n-ary predicate symbol then $P(t_1, \ldots, t_n)$ is an atom template.

A2' If t_1 or t_2 are template terms and ρ is a relational symbol then $t_1 \rho\, t_2$ is a relational atom template.

F1' If $p,\ q$ are template formulae (atoms) then $p \wedge q,\ X_k p$ are template formulae.

According to these new syntactic rules, we may now define the corresponding templates for the concepts of temporal atom, constraint formula and temporal rule.

Definition 3.5 *An event (or temporal atom) template is an atom formed by the predicate symbol E followed by a bracketed n-tuple of terms* $(n \geq 1)$ $E(y_1, y_2, \ldots, y_n)$*, where each term* y_i *is a variable symbol. A short temporal atom template is the atom* $E(y_1)$.

Definition 3.6 *A template constraint formula for the template event* $E(y_1, y_2, \ldots y_n)$ *is a template conjunctive compound formula,* $C_1 \wedge C_2 \wedge \cdots \wedge C_k$*, where each* C_j *is a relational atom template. The first term of* C_j *is one of the variable symbols* y_i*,* $i = 1 \ldots n$ *and the second term is a constant symbol.*

Consequently, a short constraint formula template is the relational atom $y_1 = c$.

Definition 3.7 *A temporal rule template is a formula of the form* $H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$*, where* H_{m+1} *is a short constraint formula template and* H_i *are template constraint formulae, prefixed by the temporal connectives* $X_{-k},\ k \geq 0$*. The maximum value of the index k is called the time window of the temporal rule template.*

Practically, the only formulae constructed in L are temporal atoms, constraint formulae, temporal rules and the corresponding templates.

3.2.2 Semantics

The semantics of L is provided by an interpretation I over a domain D (in our formalism, D is always a linearly ordered domain). The interpretation assigns an appropriate meaning over D to the (non-logical) symbols of L: essentially, the *n*-ary predicate symbols are

interpreted as concrete, n-ary relations over D, while the n-ary function symbols are interpreted as concrete, n-ary functions on D (Note: an n-ary relation over D may be viewed as an n-ary function $D^n \rightarrow B$, where B is the set $\{true, false\}$). More precisely we assign a meaning to the symbols of L as follows:

- for an n-ary predicate symbol P, $n \geq 1$, the meaning $I(P)$ is a function $D^n \rightarrow B$
- for an n-ary function symbol, f, $n \geq 1$, the meaning $I(f)$ is a function $D^n \rightarrow D$,
- for a constant symbol c the meaning $I(c)$ is an element of D,
- for a variable symbol y the meaning $I(y)$ is an element of D.

The interpretation I is extended to arbitrary terms as $I(f(t_1, \ldots, t_n)) = I(f)(I(t_1), \ldots, I(t_n))$. For p an atomic or compound formula, the meaning of truth under interpretation I – written $I \models p$ – is defined as:

- $I \models P(t_1, \ldots, t_n)$, where P is an n-ary predicate symbol and $t_1, \ldots, t_n$ are terms, if and only if $I(P)(I(t_1), \ldots, I(t_n)) = true$.
- $I \models t_1 \rho\, t_2$, where $t_1,\ t_2$ are terms and ρ is a relational symbol, if and only if $I(t_1)\rho I(t_2)$.
- $I \models p \wedge q$, where $p,\ q$ are formulae, if and only if $I(p) = true$ and $I(q) = true$.

Usually, the domain D is imposed during the discretisation phase, which is a preprocessing phase used in almost all knowledge extraction methodologies. Based on Definition 2, an event can be seen as a labelled (constant symbol t_1) sequence of points extracted from raw data and characterized by a finite set of features (terms $t_2, \cdots, t_n$). Let D_e be the set containing all the strings used as event names. We will extend this set by adding, for each $e \in D_e$, the strings *start_e* and *stop_e*. Finally, the domain D is the union $D_e \cup D_f$, where D_e is the extended set of strings and D_f represents the union of all sub-domains corresponding to chosen features.

To define a first-order linear temporal logic based on L, we need a structure having a temporal dimension and capable to capture the relationship between a time moment and the interpretation I at this moment.

Definition 3.8 *Given L and a domain D, a (first order) linear time structure is a triple* $M = (S, x, \boldsymbol{I})$*, where S is a set of states,* $x : \mathbb{N} \rightarrow S$ *is an infinite sequence of states* $(s_{(1)}, s_{(2)}, \ldots, s_{(n)}, \ldots)$ *and* $\boldsymbol{I}$ *is a function that associates with each state s an interpretation* $\boldsymbol{I}_s$ *of all symbols from L.*

Remark: The notation $s_{(i)}$, representing the state at position i in the infinite sequence x, was chosen to avoid confusion with the notation s_i, representing the state number i of the set S.

In the framework of linear temporal logic, the set of symbols is divided into two classes, the class of global symbols and the class of local symbols. Intuitively, a global symbol w has the same interpretation in each state, i.e. $\mathbf{I}_s(w) = \mathbf{I}_{s'}(w) = \mathbf{I}(w)$, for all $s, s' \in S$; the interpretation of a local symbol may vary, depending on the state at which it is evaluated. The formalism of temporal rules assumes that all function symbols (including constants) and all relational symbols are global, whereas the predicate symbols and variable symbols are local. Consequently, as the temporal atoms, constraint formulae, temporal rules and the corresponding templates are expressed using the predicate symbol E or the variable symbols y_i, the meaning of truth for these formulae depends on the state at which they are evaluated. Given a first order time structure M and a formula p, we denote the instant i (or equivalently, the state $s_{(i)}$) for which $\mathbf{I}_{s_{(i)}}(p) = true$ by $(M, i) \models p$ – or simply $i \models p$, if there is no confusion for M – i.e. at time instant i the formula p is true. Therefore, $i \models E(t_1, \ldots, t_n)$ means that at time i an event with the name $\mathbf{I}(t_1)$ and characterized by the global features $\mathbf{I}(t_2), \ldots, \mathbf{I}(t_n)$ occurs. Using this definition, we can also define:

- $i \models E(start\_t_1, \ldots, t_n)$ iff $i \models E(t_1, \ldots, t_n)$ and $((i-1) \not\models E(t_1, \ldots, t_n))$,
- $i \models E(stop\_t_1, \ldots, t_n)$ iff $i \models E(t_1, \ldots, t_n)$ and $((i+1) \not\models E(t_1, \ldots, t_n))$

Concerning the event template $E(y_1, \ldots, y_n)$, the interpretation of the variable symbols y_j at the state s_i, $\mathbf{I}_{s_i}(y_j)$, is chosen such that $i \models E(y_1, \ldots, y_n)$ for every time moment i. Because

- $i \models p \wedge q$ if and only if $i \models q$ and $i \models q$, and
- $i \models X_k p$ if and only if $i + k \models p$,

a constraint formula (template) is true at time i if and only if all relational atoms are true at time i and $i \models E(t_1, \ldots, t_n)$, whereas a temporal rule (template) is true at time i if and only if $i \models H_{m+1}$ and $i \models (H_1 \wedge \cdots \wedge H_m)$.

Remark. The fact that the symbols of language L are divided into two sets (local and global), according to the persistence of their interpretation along the infinite sequence of states $s_1, s_2, \ldots$, is the main reason for the introduction of the notion of *template*. Consider, as example, the temporal atom $E(t_1, t_2, t_3)$ and its corresponding template $E(y_1, y_2, y_3)$. In our vision, the event template is a kind of event pattern and because there is a real event which matches the pattern (an event with name $I(t_1)$ and features $I(t_2)$ and $I(t_3)$), the interpretation of the template must be true at each moment. For this reason we imposed the condition that the interpretation of variable symbols must be chosen such that $i \models E(y_1, y_2, y_3)$ for every time moment i. On the other hand, we expect that in the real word, an event occurs only at certain moments, i.e. the interpretation of the event is evaluated *true* only at these moments. Because the terms $t_i,\ i = 1..3$ are global symbols (as constant and function symbols) and $I(E(t_1, t_2, t_3)) = I(E)(I(t_1), I(t_2), I(t_3))$, the only way to achieve the variability in time for the event interpretation is to include the predicate symbol E in the set of local symbols.

3.2.3 Consistency

Now suppose that the following assumptions are true:

A. For each formula p in L, there is an algorithm that calculates the value of the interpretation $\mathbf{I}_s(p)$, for each state s, in a finite number of steps.

B. There are states (called incomplete states) that do not contain enough information to calculate the interpretation for all formulae defined at these states.

C. It is possible to establish a measure, (called *general interpretation*) about the degree of truth of a compound formula along the entire sequence of states $(s_{(1)}, \ldots, s_{(n)}, \ldots)$.

The first assumption expresses the calculability of the interpretation **I**. The second assumption expresses the situation when only the body of a temporal rule can be evaluated at time moment i, but not the head of the rule. Therefore, for the state s_i, we cannot calculate the interpretation of the temporal rule and the only solution is to estimate it using a general interpretation. This solution is expressed by the third assumption. (*Remark:* The second assumption violates the condition about the existence of an interpretation at each state s_i, as defined in Definition 3.8. But it is well known that in data mining sometimes data is incomplete or is missing. Therefore, we must modify this condition as "***I*** *is a function that associates with* `almost` *each state s an interpretation* $\boldsymbol{I}_s$ *of all symbols from L* ").

However, to ensure that this general interpretation is well defined, the linear time structure must present some property of consistency. Practically, this means that if we take any sufficiently large subset of time instants, the conclusions we may infer from this subset are sufficiently close from those inferred from the entire set of time instants.

Definition 3.9 *Given L and a linear time structure M, we say that M is a consistent time structure for L if, for every formula p, the limit* $supp(p) = \lim_{n\to\infty} \frac{\#A}{n}$ *exists, where* $A = \{i \in \{1, \ldots, n\} \mid (M, i) \models p\}$ *and # means "cardinality". The notation* $supp(p)$ *denotes the support (of truth) of p.*

Based on the concept of consistency, we can now define a function (denoted general interpretation) to measure the degree of truth for an n-ary predicate symbol P, along a sequence of states $s_1, s_2, \ldots$.

Definition 3.10 *Given L and a consistent linear time structure M for L, the general interpretation* I_G *for an n-ary predicate P is a function* $D^n \to [0, 1]$*, such that, for each n-tuple*

of terms $\{t_1, \ldots, t_n\}$, $I_G(P(t_1, \ldots, t_n)) = supp(P(t_1, \ldots, t_n)$.

The general interpretation is naturally extended to constraint formulae, temporal rules and the corresponding templates. There is another useful measure, called *confidence*, but available only for temporal rules (templates). This measure is calculated as a limit ratio between the number of certain applications (time instants where both the body and the head of the rule are true) and the number of potential applications (time instants where only the body of the rule is true). The reason for this choice is related to the presence of incomplete states, where the interpretation for the implicated clause cannot be calculated.

DEFINITION 3.11 *The confidence of a temporal rule (template)* $H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$ *is the limit (if exists)* $\lim\limits_{n\to\infty} \dfrac{\#A}{\#B}$, *where* $A = \{i \in \{1, \ldots, n\} \mid i \models H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}\}$ *and* $B = \{i \in \{1, \ldots, n\} \mid i \models H_1 \wedge \cdots \wedge H_m\}$.

This definition is well-defined, because if the cardinality of the set $B = \{i \in \{1, \ldots, n\} \mid i \models H_1 \wedge \cdots \wedge H_m\}$ is zero, then necessarily the cardinality of the set $A = \{i \in \{1, \ldots, n\} \mid i \models H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}\}$ is also zero, and, by convention, we consider the ratio $0/0$ as being zero.

3.2.3.1 Properties of the Support and Confidence

It is not sufficient to demand the existence of the support for each atomic formula over L, in the frame of Definition 3.9, to assure the existence of the support for *each* formula over L. In fact, if p, q are atomic formulae then the existence of the limits $\lim\limits_{n\to\infty} n^{-1}\#\{i \in \{1, \ldots, n\} \mid i \models p\}$ and $\lim\limits_{n\to\infty} n^{-1}\#\{i \in \{1, \ldots, n\} \mid i \models q\}$ does not imply the existence of the limit $\lim\limits_{n\to\infty} n^{-1}\#\{i \in \{1, \ldots, n\} \mid i \models p \wedge q\}$. As example, consider a formula p which is true at time instants $2k + 1$, $k \in \mathbf{N}$, and another formula q which is true at time instants even or odd, depending on the interval these points belong to. More precisely, q is true for all odd time instants from the interval $[2^{k+2} - 3, \ldots, 2^{k+2} + 2^{k+1} - 4]$ and for all even time instants from the interval $[2^{k+2} + 2^{k+1} - 3, \ldots, 2^{k+3} - 4]$, with $k = 0, 1, \ldots$. A consequence of this

formal definition is that the sequence of time instants for which q is true is composed of successive subsequences of odd and even points, of length 2^k, $k = 0, 1, 2....$ The sequence $x_n = n^{-1}\#\{i \leq n \,|\, i \models q\}$, $n \in \mathbf{N}$, may be expressed in the following general form:

$$x_n = \begin{cases} \frac{\lceil \frac{n}{2} \rceil}{n} & \text{if } n \in [2^{k+2} - 3, \ldots, 2^{k+2} + 2^{k+1} - 4], \\ \frac{\lfloor \frac{n}{2} \rfloor}{n} & \text{if } n \in [2^{k+2} + 2^{k+1} - 3, \ldots, 2^{k+3} - 4]. \end{cases}$$

Each of these disjoint subsequences converges to the same value $1/2$, therefore the sequence x_n is convergent. A similar argument may be used to prove the convergence of the sequence $y_n = n^{-1}\#\{i \leq n \,|\, i \models p\}$, $n \in \mathbf{N}$. Consider now the sequence corresponding to the formula $p \wedge q$, i.e. $z_n = n^{-1}\#\{i \leq n \,|\, i \models p \wedge q\}$. In each interval $[2^{k+2} - 3, \ldots, 2^{k+3} - 4]$, the odd time instants for which q is true are $2^{k+2} - 3, 2^{k+2} - 1, \ldots, 2^{k+2} + 2^{k+1} - 3$, i.e. 2^k points. As the formula p is always true for odd time instants, we can conclude that these 2^k points are also the only points in the above interval for which $p \wedge q$ is true. Lets take two subsequences of the sequence z_n, the first $z'_k = z_{2^{k+3}-4}$ and the second $z''_k = z_{2^{k+2}+2^{k+1}-4}$, $k \in \mathbf{N}$. If we take account of the partition induced on $\mathbf{N}$ by the sets of intervals $[2^{k+2} - 3, \ldots, 2^{k+3} - 4]$ and of the list of points for which $p \wedge q$ is true in each such interval, the general form of the two subsequences is given by:

$$z'_k = z_{2^{k+3}-4} = \frac{1 + 2 + \cdots + 2^k}{2^{k+3} - 4} = \frac{2^{k+1} - 1}{2^{k+3} - 4} = \frac{1}{4}$$

and

$$z''_k = z_{2^{k+2}+2^{k+1}-4} = \frac{1 + 2 + \cdots + 2^k}{2^{k+2} + 2^{k+1} - 4} = \frac{2^{k+1} - 1}{2^{k+2} + 2^{k+1} - 4}.$$

Obviously, the first subsequence converges to $1/4$. For the second subsequence, consider the following transformations:

$$z''_k = \frac{2^{k+1} - 1}{2^{k+2} + 2^{k+1} - 4} = \frac{2^{k+1}(1 - 2^{-k-1})}{2^{k+1}(2 + 1 - 2^{-k-3})} = \frac{1 - 2^{-k-1}}{3 - 2^{-k-3}}.$$

Consequently,

$$\lim_{k \to \infty} z''_k = \lim_{k \to \infty} \frac{1 - 2^{-k-1}}{3 - 2^{-k-3}} = \frac{1 - \lim_{k \to \infty} 2^{-k-1}}{3 - \lim_{k \to \infty} 2^{-k-3}} = \frac{1}{3}.$$

Because the two subsequences converge to two different limits, the sequence z_n has no limit and so our affirmation is proved.

Two useful lemmas concerning the support for specific formulae are presented in the following (a language L and M a consistent time structure for L are implicitly considered). The first involves the expression of the support for a formula prefixed by a temporal connective.

LEMMA 3.1 *The support of a formula $X_{\pm 1}p$ is equal with the support of the formula p.*

Proof For $n \geq 2$, consider $x_n = n^{-1}\#\{i \leq n \,|\, i \models X_{\pm 1}p\}$ and $y_n = n^{-1}\#\{i \leq n \,|\, i \models p\}$. Then

$$x_n = \frac{\#\{i \leq n \,|\, i \models X_{\pm 1}p\}}{n} = \frac{\#\{i \leq n \,|\, i \pm 1 \models p\}}{n} = \frac{\#\{i \leq n \pm 1 \,|\, i \models p\}}{n \pm 1} \cdot \frac{n \pm 1}{n} = y_{n\pm 1} \cdot \frac{n \pm 1}{n}$$

Therefore

$$\mathrm{supp}(X_{\pm 1}p) = \lim_{n\to\infty} x_n = \lim_{n\to\infty} y_{n\pm 1} \cdot \frac{n \pm 1}{n} = \lim_{n\to\infty} y_{n\pm 1} \cdot \lim_{n\to\infty} \frac{n \pm 1}{n} = \mathrm{supp}(p).$$

As a consequence of lemma 3.1, we have

COROLLARY 3.1 *For all $k \in \mathbf{Z}$, $supp(X_k p) = supp(p)$.*

If a formula is syntactically defined using only global symbols from L, then the value of its interpretation remains the same along the entire sequence of states s_i. We denote such a formula as global formula. A simple example is a formula of type $t_1 \rho\, t_2$, with terms t_1, t_2 and ρ a relational symbol. A formula which is not global is called a local formula.

LEMMA 3.2 *If p and q are formulae and p is a global formula then $supp(p \wedge q) = supp(p) \cdot supp(q)$.*

Proof If p is a global formula then the meaning of truth for this formula is either *true* or *false*. Therefore, $\{i \leq n \,|\, i \models p\}$ is either n or 0, and so $\mathrm{supp}(p)$ is either one or zero. If the interpretation of p is *true* in each state, then $\{i \leq n \,|\, i \models p \wedge q\} = \{i \leq n \,|\, i \models q\}$ and consequently $\mathrm{supp}(p \wedge q) = \mathrm{supp}(q) = 1 \cdot \mathrm{supp}(q) = \mathrm{supp}(p) \cdot \mathrm{supp}(q)$. If the interpretation

of p is false in each state, then $\{i \leq n \mid i \models p \wedge q\} = 0$ and consequently $\text{supp}(p \wedge q) = 0 = 0 \cdot \text{supp}(q) = \text{supp}(p) \cdot \text{supp}(q)$.

As a consequence of lemma 3.2 and of the observation that $t_1 \rho\, t_2$ is a global formula, we have

Corollary 3.2 *Given a constraint formula $E(t_1, \ldots, t_m) \wedge C_\wedge \ldots \wedge C_k$, we have the relation $supp\,(E(t_1, \ldots, t_m) \wedge C_\wedge \ldots \wedge C_k) = supp\,(E(t_1, \ldots, t_m)) \prod_{i=1}^{k} supp(C_i)$.*

A generalization of lemma 3.2 is given by the following corollary.

Corollary 3.3 *Let be $p_1, p_2, \ldots, p_n$ global formulae in L and $q_1, q_2, \ldots, q_m$ local formulae in L. Then $supp(p_1 \wedge \ldots \wedge p_n \wedge q_1 \wedge \ldots \wedge q_m) = supp(q_1 \wedge \ldots \wedge q_m) \cdot \prod_{i=1}^{n} supp(p_i)$.*

By applying the corollaries 3.1, 3.2 and 3.3 to a temporal rule, we obtain

Corollary 3.4 *Let be $H_1 \wedge \ldots H_m \mapsto H_{m+1}$ a temporal rule, where for each $i \leq m$, H_i has the form $X_{-w_i}(E(t_{i1}, \ldots, t_{in}) \wedge C_{i1} \wedge \ldots \wedge C_{ik_i})$, whereas H_{m+1} has the form $E(t_{(m+1)1}) \wedge (t_{(m+1)1} = c)$. Then the support of the temporal rule (**H**) is given by the following expression*

$$supp(\boldsymbol{H}) = supp\left(E(t_{(m+1)1}) \bigwedge_{j=1}^{m} X_{-w_j} E(t_{j1}, \ldots, t_{jn})\right) \cdot supp(t_{(m+1)1} = c) \cdot \prod_{j=1}^{m} \prod_{l=1}^{k_j} supp(C_{jl}).$$

The relation connecting the supports of two formulae, as it was expressed in Lemma 3.2, is in fact the consequence of a more general property, called *support independence*, which is very similar to the concept of stochastic independence.

Definition 3.12 *The pair $\{p, q\}$ of formulae is said to be independent iff $supp(p \wedge q) = supp(p) supp(q)$.*

Definition 3.13 *The formula p is said to be **almost true** (respectively **false**) iff $supp(p) = 1$ (respectively $supp(p) = 0$).*

As a consequence of this definition, a global formula is always an almost true (false) formula. Therefore, Lemma 3.2 is in fact a consequence of the following lemma:

Lemma 3.3 *If p is a formula almost true (false) then the pair $\{p, q\}$ of formulae is independent for each formula q in L.*

The proof of the lemma is based on the well-known relation between two sets, $\#\{A \cup B\} = \#\{A\} + \#\{B\} - \#\{A \cap B\}$, which implies the inequalities $\#\{A \cap B\} \leq \#\{A\}$ and $\#\{A\} \leq \#\{A \cup B\}$. If we take $A = \{i \leq n \,|\, i \models p\}$ and $B = \{i \leq n \,|\, i \models q\}$ and pass to the limit in the expression of the support for p and $p \wedge q$, we obtain:

- $\text{supp}(p) = 0 \Rightarrow \text{supp}(p \wedge q) = 0 \Rightarrow \text{supp}(p \wedge q) = \text{supp}(p)\text{supp}(q)$
- $\text{supp}(p) = 1 \Rightarrow \text{supp}(p \wedge q) = \text{supp}(q) \Rightarrow \text{supp}(p \wedge q) = \text{supp}(p)\text{supp}(q)$

The following lemmas concern the properties of the measure *confidence* and we start by establishing the connection between the property of consistency and the existence of the confidence.

Lemma 3.4 *If M is a consistent linear time structure for L then every temporal rule (template) $H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$ for which $(H_1 \wedge \cdots \wedge H_m)$ is not an almost false formula has a well-defined confidence.*

Proof Consider M a consistent linear time structure for L. According to the definition 3.9, there exist the limits of the sequences

$$\text{supp}(H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}) = \lim_{n\to\infty} \frac{\#\{i \in \{1, \ldots, n\} \,|\, i \models H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}\}}{n}$$

and

$$\text{supp}(H_1 \wedge \cdots \wedge H_m) = \lim_{n\to\infty} \frac{\#\{i \in \{1, \ldots, n\} \,|\, i \models H_1 \wedge \cdots \wedge H_m\}}{n} > 0.$$

Therefore,

$$\begin{aligned} conf(H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}) &= \lim_{n\to\infty} \frac{\#\{i \le n \,|\, i \models H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}\}}{\{i \le n \,|\, i \models H_1 \wedge \cdots \wedge H_m\}} \\ &= \lim_{n\to\infty} \frac{\#\{i \le n \,|\, i \models H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}\}}{n} \cdot \frac{n}{\#\{i \le n \,|\, i \models H_1 \wedge \cdots \wedge H_m\}} \\ &= \left(\lim_{n\to\infty} \frac{\#\{i \le n \,|\, i \models H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}\}}{n}\right)\left(\lim_{n\to\infty} \frac{n}{\#\{i \le n\} \,|\, i \models H_1 \wedge \cdots \wedge H_m\}}\right) \\ &= \frac{\mathrm{supp}(H_1 \wedge \cdots \wedge H_m \wedge H_{m+1})}{\mathrm{supp}(H_1 \wedge \cdots \wedge H_m)} \end{aligned}$$

If $\mathrm{supp}(H_1 \wedge \cdots \wedge H_m) = 0$, then either $\#\{i \le n \,|\, i \models H_1 \wedge \cdots \wedge H_m\}$ is zero for all n (case in which $conf(H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1})$ is zero by convention), or $n^{-1}\#\{i \le n \,|\, i \models H_1 \wedge \cdots \wedge H_m\}$ is a strict positive sequence converging to zero (case in which the existence of the confidence is not even guaranteed).

Finally, as a consequence of Lemma 3.4 and of Corollary 3.4, we obtain

Corollary 3.5 *Let be $H_1 \wedge \ldots H_m \mapsto H_{m+1}$ a temporal rule, where for each $i \le m$, H_i has the form $X_{-w_i}(E(t_{i1}, \ldots, t_{in}) \wedge C_{i1} \wedge \ldots \wedge C_{ik_i})$, whereas H_{m+1} has the form $E(t_{(m+1)1}) \wedge (t_{(m+1)1} = c)$. If $(H_1 \wedge \cdots \wedge H_m)$ is not an almost false formula then the confidence of the temporal rule ($\boldsymbol{H}$) is given by the following expression*

$$conf(\boldsymbol{H}) = \frac{supp\left(E(t_{(m+1)1}) \bigwedge_{j=1}^{m} X_{-w_j} E(t_{j1}, \ldots, t_{jn})\right)}{supp\left(\bigwedge_{j=1}^{m} X_{-w_j} E(t_{j1}, \ldots, t_{jn})\right)} \cdot supp(t_{(m+1)1} = c)$$

3.2.4 Consistent Time Structure Model

For different reasons, (the user has no access to the entire sequence of states, or the states he has access to are incomplete), the general interpretation cannot be calculated. A solution is to estimate I_G using a finite linear time structure, i.e. a model.

Definition 3.14 *Given L and a consistent time structure $M = (S, x, \boldsymbol{I})$, a model for M is a structure $\tilde{M} = (\tilde{T}, \tilde{x})$ where $\tilde{T}$ is a finite temporal domain $\{i_1, \ldots, i_n\}$, $\tilde{x}$ is the subsequence of states $\{x_{(i_1)}, \ldots, x_{(i_n)}\}$ (the restriction of x to the temporal domain $\tilde{T}$) and for each i_j, $j = 1, \ldots, n$, the state $x_{(i_j)}$ is a complete state.*

Now we may define the estimator for a general interpretation:

DEFINITION 3.15 *Given L and a model $\tilde{M}$ for M, an estimator of the general interpretation for an n-ary predicate P, $I_{G(\tilde{M})}(P)$, is a function $D^n \rightarrow [0,1]$, assigning to each atomic formula $p = P(t_1, \ldots, t_n)$ the value defined as the ratio $\frac{\#A}{\#\tilde{T}}$, where $A = \{i \in \tilde{T} \mid i \models p\}$. The notation $supp(p, \tilde{M})$ will denote the estimated support of p, given $\tilde{M}$.*

The extension of this definition to the other types of formulae in L demands a deeper analysis. Consider, as example, the model $\tilde{M}$ induced by the sequence of $n > 1$ states $\tilde{x} = x_{(1)}, \ldots, x_{(n)}$. The interpretation of a formula $X_1 p$ at the state $x_{(n)}$ can not be calculated, because $n \models X_1 p$ if $(n+1) \models p$ and $x_{(n+1)} \notin \tilde{x}$. Therefore, the cardinality of the set $A = \{i \leq n \mid i \models X_1 p\}$ is strictly smaller than n, which means that, for p a global formula having the meaning of truth *true*, the estimated support is

$$\text{supp}(X_1 p, \tilde{M}) = (n-1)/n \neq 1 = \text{supp}(X_1 p).$$

The fact that the support estimator is biased seems at first glance without importance, especially when, as in this case, the bias (n^{-1}) tends to zero for $n \uparrow \infty$. But considering a formula of type $X_n p$, it is evidently that the interpretation can not be calculated at none of the states from $\tilde{x}$, and so even the support estimator is not defined. Before indicating how the expression $\frac{\#A}{\#\tilde{T}}$ must be adjusted to avoid this kind of problem, we start by defining the standard form of a formula in L.

DEFINITION 3.16 *A formula $X_{k_1} p_1 \wedge X_{k_2} p_2 \wedge \ldots \wedge X_{k_n} p_n$, where $n \geq 1$ and p_i are atoms of L, is in standard form if exists $i_0 \in \{1, \ldots, n\}$ such that $k_{i_0} = 0$ and for all $i = 1..n$, $k_i \leq 0$.*

For an atomic formula p, it is clearly that its standard form is $X_0 p$. Another example of formula in standard form is a temporal rule (template), where the head of the rule is prefixed by X_0 and all other constraint formulae are prefixed by $X_{-k}, k \geq 0$. According to Corollary 3.1, if M is a consistent time structure then the support of a formula in L does not change if it is prefixed with a temporal connective X_k, $k \in \mathbb{Z}$. Therefore, to each formula p

in L corresponds an equivalent formula (under the measure supp) having a standard form (denoted $\mathcal{F}(p)$). Based on this concept, we can now give a non equivocal definition for *time windows*:

Definition 3.17 *Let be p a formula in L having the standard form* $X_{k_1}p_1 \wedge X_{k_2}p_2 \wedge \ldots \wedge X_{k_n}p_n$. *The time window of p – denoted* $w(p)$ *– is defined as* $\max\{|k_i| : i = 1..n\}$

In the following, a formula having a time window equal zero will be called *temporal free formula*, whereas a formula with a strictly positive time window will be called a *temporal formula*. The concept of time window allows us to define a non biased estimator for the support measure.

Definition 3.18 *Given L and a model* $\tilde{M}$ *for M, the estimator of the support for a formula p in L and having* $w(p) < \#\tilde{T} = m$, *denoted* $supp(p, \tilde{M})$, *is the ratio*

$$\frac{\#A}{m - w(p)}, \text{ where } A = \{i \in \tilde{T} \mid i \models \mathcal{F}(p)\}.$$

According to this definition, if $w(p) \geq m$ the estimator $\text{supp}(p, \tilde{M})$ is not defined. The use of the standard form of the formula, in the construction of the set A, eliminates the interpretation problem for a formula of type $X_k p$, $k \geq m$. Moreover, it is easy to see that $\text{supp}(X_k p, \tilde{M}) = \text{supp}(p, \tilde{M})$, for all $k \in \mathbb{Z}$.

Definition 3.19 *Given L and a model* $\tilde{M}$ *for M, an estimate of the general interpretation for a formula p is given by*

$$I_{G(\tilde{M})}(p) = \begin{cases} supp(p, \tilde{M}), & \text{if } w(p) < \#\tilde{T}, \\ 0 & \text{if } w(p) \geq \#\tilde{T} \end{cases} \tag{3.1}$$

Once again, the estimation of the confidence for a temporal rule (template) is defined as:

Definition 3.20 *Given a model* $\tilde{M} = (\tilde{T}, \tilde{x})$ *for M, the estimation of the confidence for the temporal rule (template)* $H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$ *is the ratio* $\frac{\#A}{\#B}$, *where* $A = \{i \in \tilde{T} \mid i \models$

$H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}\}$ *and* $B = \{i \in \tilde{T} \mid i \models H_1 \wedge \cdots \wedge H_m\}$. *The notation* $conf(H, \tilde{M})$ *will denote the estimated confidence of the temporal rule (template)* H *given* $\tilde{M}$.

According to the same arguments used in the definition of a correct support estimator, the existence of a confidence estimator for a temporal rule H is guaranteed only for models having a number of states greater than the time window of the rule. Moreover, if $\tilde{\tilde{T}}$ is the set obtained from $\tilde{T}$ by deleting the first $w(H_1 \wedge \ldots \wedge H_{m+1}) - w(H_1 \wedge \ldots \wedge H_m)$ states, then we can obtain a non biased confidence estimator if in the expression of the set $B = \{i \in \tilde{T} \mid i \models H_1 \wedge \cdots \wedge H_m\}$ the set $\tilde{T}$ is replaced with $\tilde{\tilde{T}}$.

3.3 Methodology Versus Formalism

As it was extensively presented in Chapter 2, the methodology for temporal rules extraction may be structured in two phases. During the first phase one transforms sequential raw data into sequences of events. Practically, this means to establish the set of events, identified by names, and the set of features, common for all events.

In the frame of our formalism, during this phase we establish the set of temporal atoms which can be defined syntactically in L. For this we start by defining the first-order temporal language L. Considering as raw data the database described in Example 2.1, we include in L a 3-ary predicate symbol E, three variable symbols $y_i, i = 1..3$, two 12-ary function symbols f and g, two sets of constant symbols – $\{d_1, \ldots, d_6\}$ and $\{c_1, \ldots, c_n\}$ – and the usual set of relational symbols and logical(temporal) connectives. As we showed in the above example and according to the syntactic rules of L, an event is defined as $E(d_i, f(c_{j_1}, \ldots, c_{j_{12}}), g(c_{k_1}, \ldots, c_{k_{12}}))$, whereas an event template is defined as $E(y_1, y_2, y_3)$.

Also provided during this phase is the semantics of L. Firstly, the domain $D = D_e \cup D_f$ (see Sect. 3.2.2) is defined. According to the results of the discretisation algorithm applied to the raw data from the cited example, the domain D_e is defined as {*peak, start_peak, stop_peak, flat, start_flat, stop_flat, valley, start_valley, stop_valley*}. During the step *global features calculation*, two features – the mean and the standard error – were selected

to capture the continuous aspect of the events. Consequently, the domain $D_f = \mathfrak{R}^+$, as the stock prices are positives real numbers and the features are statistical functions.

Secondly, a linear time structure M, i.e. a triple $(S, x, \mathbf{I})$ (see Def. 3.8) is specified. The database of events, obtained after the first phase of the methodology, contains tuples with three values, (v_1, v_2, v_3). For a tuple with a recording index i, the first value expresses the name of the event – *peak*, *flat*, *valley* – which occurs at time moment i and the two other values express the result of the two features. Therefore, we define a state s as a triple (v_1, v_2, v_3), the set S as the set of all tuples from database and the sequence x as the ordered sequence of tuples in the database (see Table 1).

Table 1: The first nine states of the linear time structure M (example)

Index	State	Index	State	Index	State
1	$(peak, 10, 1.5)$	4	$(flat, 1, 0.5)$	7	$(valley, 15, 1.9)$
2	$(peak, 10, 1.5)$	5	$(flat, 1, 0.5)$	8	$(flat, 3, 1.1)$
3	$(peak, 14, 2.2)$	6	$(flat, 1, 0.5)$	9	$(peak, 12, 1.2)$

At this stage the interpretation of all symbols (global and local) can be defined. For the global symbols (function symbols and relational symbols), the interpretation is quite intuitive. Therefore, the meaning $\mathbf{I}(d_j)$ is an element of D_e, the meaning $\mathbf{I}(c_j)$, $j = 1..n$, is a positive real number, whereas the meaning $\mathbf{I}(f)$, respectively $\mathbf{I}(g)$, is the function $f : D_f^{12} \rightarrow D_f$, $f(\mathbf{x}) = \overline{\mathbf{x}}$, respectively the function $g : D_f^{12} \rightarrow D_f$, $g(\mathbf{x}) = \text{se}(\mathbf{x})$ (we used the standard notations in statistics for the mean and standard error estimators).

The interpretation of a local symbol (the variable symbols y_i and the predicate symbol E) depends on the state at which it is evaluated. According to the assumption A (see Sect. 3.2.3), the function $\mathbf{I}_{s_i}(E)$ defined on D^3 with values in $B = \{true, false\}$ is provided by a finite algorithm. This algorithm will receive as input at least the state s_i and will provide as output one of the values from B. Therefore, the interpretation of $E(t_1, t_2, t_3)$ evaluated at s_i is defined as:

ALGORITHM 1 ***Temporal atom evaluation***

Consider the state $s_i = (v_1, v_2, v_3)$

If $(\boldsymbol{I}_{s_i}(t_1) = v_1)$ *and* $(\boldsymbol{I}_{s_i}(t_2) = v_2)$ *and* $(\boldsymbol{I}_{s_i}(t_3) = v_3)$

Then $\boldsymbol{I}_{s_i}(E(t_1, t_2, t_3)) = true$

Else $\boldsymbol{I}_{s_i}(E(t_1, t_2, t_3)) = false$

Finally, the interpretation of the variable symbol y_j at the state s_i is given by $\mathbf{I}_{s_i}(y_j) = v_j$, $j = 1..3$, which satisfies the condition imposed on the interpretation of a temporal atom template (see Sect. 3.2.2), which is $\mathbf{I}_{s_i} E(y_1, y_2, y_3) = true$ for each state s_i. Having well-defined the language L, the syntax and the semantics of L, as well as the linear time structure M, we can construct the temporal atoms evaluated as *true* at the time moment i (see Table 2).

Table 2: The temporal atoms evaluated *true* at the first nine states of M (example)

State	Temporal atom
1	$E(peak, 10, 1.5)$, $E(start\_peak, 10, 1.5)$
2	$E(peak, 10, 1.5)$, $E(stop\_peak, 10, 1.5)$
3	$E(peak, 14, 2.2)$, $E(start\_peak, 14, 2.2)$, $E(stop\_peak, 14, 2.2)$
4	$E(flat, 1, 0.5)$, $E(start\_flat, 1, 0.5)$
5	$E(flat, 1, 0.5)$
6	$E(flat, 1, 0.5)$, $E(stop\_flat, 1, 0.5)$
7	$E(valley, 15, 1.9)$, $E(start\_valley, 15, 1.9)$, $E(stop\_valley, 15, 1.9)$
8	$E(flat, 3, 1.1)$, $E(start\_flat, 3, 1.1)$, $E(stop\_flat, 3, 1.1)$
9	$E(peak, 12, 1.2)$, $E(start\_peak, 12, 1.2)$, $E(stop\_peak, 12, 1.2)$

During the second phase of the methodology, we generate a set of temporal rules inferred from the database of events, obtained in phase one. The first induction process consists in creating classification trees, each based on a different training set. In the frame of our formalism, choosing a training set is equivalent to choosing a model $\tilde{M}$ for the linear time structure M. All the states from these models are complete states, because the algorithm which constructs the tree must know, for each time moment, the set of predictor events and the corresponding dependent event. Once the classification tree is constructed,

the test contained in each node becomes a relational atom and the set of all relational atoms situated on a path from root to a leaf become a constraint formula template. The variable symbols y_i included in the template are generated by the following rule:

- if the attribute concerned by the test is related to the event name, it is replaced by y_1
- if the attribute concerned by the test is related to the feature mean (respectively standard error), it is replaced by y_2 (respectively y_3).

The constraint formula template becomes a temporal rule template by adding temporal connectives according to the procedure which links a temporal dimension to a rule generated by C4.5 algorithm. Finally, the confidence of a temporal rule template is calculated according to the Definition 3.20.

Remark: The values of the categorical dependent variable, or the classes, may be obtained either in a supervised mode (e.g. given by an expert) or in an unsupervised mode (e.g. the names of the events). In the latter case, we may restrict the possible values to the set $\{start\_event_1, stop\_event_1, \ldots, start\_event_n, stop\_event_n\}$).

To exemplify the induction process, from a training set to temporal rule templates, let us consider the following model $\tilde{M} = \{s_1, \ldots, s_{100}\}$. According to the procedure for training set selection (see Sect. 2.2.1.2), in a first step we must indicate the sequences q_i, $i = 1..k$, of predictor variables and the sequence q_c of class values. The information on the sequences q_i is extracted from the structure of the states from the model $\tilde{M}$, i.e. given the state $s_i = (v_1, v_2, v_3)$, $0 \leq i \leq 100$, we define $q_{ji} = v_j$, $j = 1..3$. Therefore, q_1 is the sequence of the event names, q_2 is the sequence of the mean values and q_3 is the sequence of the standard error values. As there is no predefined classification (unsupervised mode), the sequence q_c is defined as $q_{ci} = q_{1i}, i = 1..100$. The next step consists in defining the parameters t_0, t_p and h, which are set to $t_0 = 100$, $t_p = 96$ and $h = 3$ (the methodology for finding the optimal value of the parameter h is based on the analysis of the classification errors, as it was described in Sect. 2.2.1.3). Concerning the tuples from the training set,

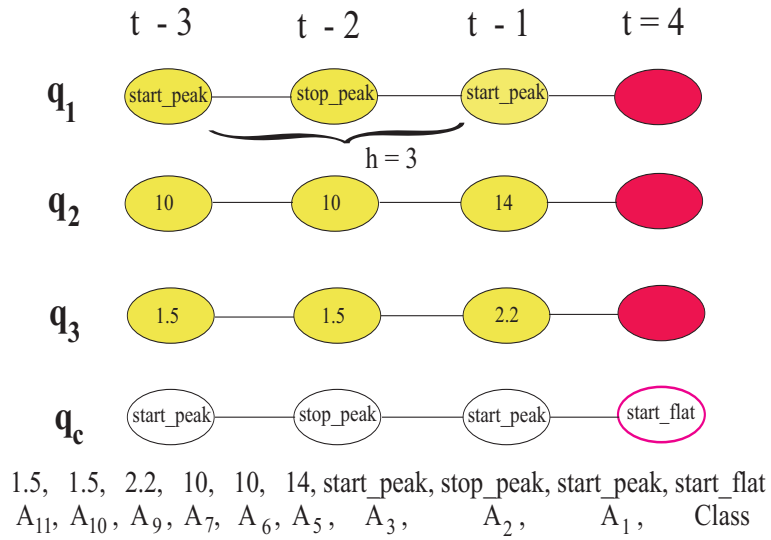


Figure 7: Graphical representation of the last tuple of the training set based on states from Table 1 and defined by the parameters $t_0 = 100$, $t_p = 96$ and $h = 3$ (including the list of corresponding attributes)

there is a minor difference compared with the procedure from Sect. 2.2.1.2 – the sequence of class values (q_c) being the same as one of the sequence of the predictor variables (q_1), we can not include in a tuple which contains the class value q_{ct} the predictor values q_{1t}, q_{2t} and q_{3t}. In other words, we can not accept that the same event occurred at time t to be simultaneously on the left and on the right side of a temporal rule. As we can see in Fig. 7, a tuple contains now $k \cdot h = 9$ predictor values instead of $k(h + 1) = 12$, and there is no attribute having an index i such that i *modulo* $(h + 1)$ is equal to 0. Suppose now that one of the rule generated by C4.5 algorithm using the previous defined training set has the form

```
A3={start_peak}, A7 < 11, A1={start_peak} -> class {start_valley}.
```

By convention, the event in the head of the rule occurs always at time moment $t = 0$, so an event from the body of the rule, corresponding to the attribute A_i, occurs at time moment $-(i \ modulo \ 4)$. By applying this observation and the convention on how to use symbol variables in a temporal rule, we obtain the following temporal rule template

$$X_{-3}(y_1 = start\_peak) \wedge X_{-3}(y_2 < 11) \\ \wedge X_{-1}(y_1 = start\_peak) \mapsto X_0(y_1 = start\_valley)$$

The induction process is repeatedly applied on different models $\tilde{M}_i$ of the time structure

Table 3: Different temporal rule templates extracted from two models $\tilde{M}$ using the induction process (example)

Model	Temporal Rule Templates
$s_1 \ldots s_{100}$	$X_{-3}(y_1 = start\_peak) \wedge X_{-3}(y_2 < 11) \wedge$ $\wedge X_{-1}(y_1 = start\_peak) \mapsto X_0(y_1 = start\_valley)$ $X_{-2}(y_1 = start\_peak) \wedge X_{-2}(y_3 < 1.1) \wedge$ $\wedge X_{-1}(y_1 = stop\_flat) \mapsto X_0(y_1 = start\_valley)$ .....
$s_{300} \ldots s_{399}$	$X_{-2}(y_1 = peak) \wedge X_{-2}(y_3 < 1.1) \wedge$ $\wedge X_{-2}(y_2 \geq 12.3) \mapsto X_0(y_1 = start\_valley)$ $X_{-4}(y_1 = stop\_peak) \wedge X_{-3}(y_1 = start\_flat) \wedge X_{-3}(y_2 >= 3.2) \wedge$ $\wedge X_{-3}(y_3 < 0.4) \wedge X_{-1}(y_1 = stop\_flat) \mapsto X_0(y_1 = start\_peak)$ .....

M, which will generate in the end different sets of local temporal rule templates (see Table 3). It is possible to obtain the same template from two different sets, but with different confidence, or templates with the same head, but with different bodies. As we already mentioned, the meta-rules inference process (fundamental principle and practical application) will be described in Chapter 6.

3.4 Summary

In this chapter we developed a formalism based on first-order temporal logic, which allows us to define the main concepts used in temporal data mining (*event, temporal rule, constraint, support, confidence*) in a formal way. The language L on which the temporal logic is based contains a restricted set of connectives (the logical connective "and" and the temporal connective "next"), but sufficient for an abstract view of temporal rules. Furthermore the symbol variables included in L permit the definition of formal models (called *templates*) for events, constraint formulae and temporal rules. Semantically, the set of local symbols (the interpretation depends on the state at which are evaluated) contains only predicate symbols and variable symbols.

The notion of *consistent linear time structure* allows us to introduce the notion of *general interpretation*. These two important concept express the fact that the structure for which the first-order temporal logic is defined represents a homogenous model and therefore the conclusions (or inferences) based on a finite model of this time structure are consistent. Moreover, if a finite model contains complete states (i.e. all the formulae in L can be evaluated), we may define estimators for the measures of support and confidence. The fact that the support of any formula is a real value between zero and one allows us to develop some concepts "borrowed" from the probability theory, as *almost true formula* or *almost false formula*, and to prove some lemmas concerning the expression of the confidence for a temporal rule (template).

Finally, we showed, using an "in house" example, how the main steps of the methodology are supported by the formalism: how to establish the language L, how to define the domain D, how to construct a linear time structure M, how to set the interpretation of local symbols and how to select a finite model having complete states.

CHAPTER IV

TEMPORAL RULES AND TIME GRANULARITY

The formalism described in Chapter 3 is developed around a time model for which the events are those that describe the system evolution (event-based temporal logics). Each formula expresses what the system does at each event. Events are referred to other events, and so on: this results in specifying relationships of precedence and cause-effect among events. But the real systems are systems whose components (events) have dynamic behavior regulated by very different - even by orders of magnitude - time granularities. Analyzing such systems (hereinafter granular systems) means to approach theories, methodologies, techniques and tools that make use of granules (or groups, classes, clusters of a universe) in the process of problem solving. Granular computing (the label which covers this approach) is a way of thinking that relies on our ability to perceive the real world under various grain sizes, to abstract and consider only those things that serve our present interest, and to switch among different granularities. By focusing on different levels of granularities, one can obtain various levels of knowledge, as well as inherent knowledge structure. Granular computing is essential to human problem solving, and hence has a very significant impact on the design and implementation of intelligent systems [Yao and Zhong, 1999, Yao, 2000, Zadeh, 1998, Lin and Louie, 2002].

The notions of granularity and abstraction are used in many subfields of artificial intelligence. The granulation of time and space leads naturally to temporal and spatial granularities. They play an important role in temporal and spatial reasoning [Euzenat, 1995, Hornsby, 2001, Stell and Worboys, 1998]. Based on granularity and abstraction, many authors studied some fundamental topics of artificial intelligence, such as, for example, knowledge representation [Zhang and Zhang, 1992], theorem proving [Giunchglia and

Walsh, 1992], search [Zhang, 2003], planning Knoblock [1993], natural language understanding [Mani, 1998], intelligent tutoring systems [McCalla et al., 1992], machine learning [Saitta, 1998], and data mining [Han et al., 1993].

Despite the widespread recognition of its relevance in the fields of formal specifications, knowledge representation and temporal databases, there is a lack of a systematic framework for time granularity. Hobbs [1985] proposed a formal characterization of the general notion of granularity, but gives no special attention to time granularity. Clifford and Rao [1988] provide a set-theoretic formalization of time granularity, but they do not attempt to relate the truth value of assertions to time granularity. Extensions to existing languages for formal specifications, knowledge representation and temporal databases to support a limited concept of time granularity are proposed in Roman [1990], Evans [1990], Ciapessoni et al. [1993]. Finally, Bettini et al. [1998a,b,c] provide a formal framework for expressing data mining tasks involving time granularities, investigate the formal relationships among event structures that have temporal constraints, define the pattern-discovery problem with these structures and study effective algorithms to solve them.

The purpose of this chapter is to extend our formalism to include the concept of time granularity. We define the process for which a given structure of time granules μ (called temporal type) induces a first-order linear time structure M_μ on the basic (or absolute) linear time structure M. The major change for the temporal logic based on M_μ is at the semantic level: for a formula p, the interpretation do not assign a meaning of truth (one of the values $\{true, false\}$), but a degree of truth (a real value from $[0, 1]$). Consequently, we can give an answer to the following question: if the temporal type μ is *finer than* temporal type ν, what is the relationship between the interpretations of the same formula p in the linear time structures M_μ and M_ν. We also study the variation process for the set of satisfiable events (degree of truth equal one) during the transition between two time structures with different granularity. By an extension at the syntactic and semantic level we define a mechanism of aggregation for events, that reflects the following intuitive phenomenon: in a coarser

world, not all events inherited from a finer world are satisfied, but in exchange there are new events which become satisfiable.

4.1 The Granularity Model

We start with the concept of a temporal type to formalize the notion of time granularities, as described in Bettini et al. [1998a].

Definition 4.1 *Let $(\mathcal{T}, <)$ (index) be a linearly ordered temporal domain isomorphic to a subset of the integers with the usual order relation, and let $(\mathcal{A}, <)$ (absolute time) be a linearly ordered set. Then, a temporal type on $(\mathcal{T}, \mathcal{A})$ is a mapping μ from $\mathcal{T}$ to $2^{\mathcal{A}}$ such that*

1. *$\mu(i) \neq \emptyset$ and $\mu(j) \neq \emptyset$, where $i < j$, imply that each element in $\mu(i)$ is less than all the elements in $\mu(j)$, and*

2. *for all $i < j$, if $\mu(i) \neq \emptyset$ and $\mu(j) \neq \emptyset$, then $\forall k, i < k < j$ implies $\mu(k) \neq \emptyset$.*

Each set $\mu(i)$, if non-empty, is called a granule of μ. Property (1) says that granules do not overlap and that the order on indexes follows the order on the corresponding granules. Property (2) disallows an empty set to be the value of a mapping for a certain index value if a lower index and a higher index are mapped to non-empty sets.

When considering a particular application or formal context, we can specialize this very general model along the following dimensions:

- choice of the index set $\mathcal{T}$,
- choice of the absolute time set $\mathcal{A}$,
- restrictions on the structure of granules,
- restrictions on the temporal types by using relationships.

We call the resulting formalization a temporal type system. Consider some possibilities for each of the above four dimensions. Convenient choices for the index set are natural numbers, integers, and any finite subset of them. The choice for absolute time is typically between dense and discrete. In general, if the application imposes a fixed basic granularity, then a discrete absolute time in terms of the basic granularity is probably the appropriate choice. However, if one is interested in being able to represent arbitrary finer temporal types, a dense absolute time is required. In both cases, specific applications could impose left/right boundedness on the absolute time set. The structure of ticks could be restricted in several ways:

(1) disallow types with *gaps* in a granule,

(2) disallow types with *non-contiguous* granules,

(3) disallow types whose granules do *not cover all* the absolute time, or

(4) disallow types with *nonuniform* granules (only types with granules having the same size are allowed).

4.1.1 Relationships and formal properties

Following Bettini et al. [1998a], we define a number of interesting relationships among temporal types.

Definition 4.2 *Let be μ and ν be temporal types on $(\mathcal{T}, \mathcal{A})$.*

- Finer-than*: μ is said to be finer than ν, denoted $\mu \preccurlyeq \nu$, if for each $i \in \mathcal{T}$, there exists $j \in \mathcal{T}$ such that $\mu(i) \subseteq \nu(j)$.*

- Groups-into*: μ is said to group into ν, denoted $\mu \trianglelefteq \nu$, if for each non-empty granule $\nu(j)$, there is a subset S of $\mathcal{T}$ such that $\nu(j) = \bigcup_{i \in S} \mu(i)$.*

- Subtype: *μ is said to be a subtype of ν, denoted $\mu \sqsubseteq \nu$, if for each $i \in \mathcal{T}$, there exists $j \in \mathcal{T}$ such that $\mu(i) = \nu(j)$.*

- Shifting*: μ and ν are said to be shifting equivalent, denoted $\mu_1 \rightleftharpoons \mu_2$, if $\mu \sqsubseteq \nu$ and $\nu \sqsubseteq \mu$.*

When a temporal type μ is finer than a temporal type ν, we also say that ν is *coarser* than μ. The *finer-than* relationship formalizes the notion of finer partitions of the absolute time. By definition, this relation is reflexive, i.e. $\mu \preccurlyeq \mu$ for each temporal type μ. Furthermore, the finer-than relation is obviously transitive. However, if no restrictions are given, it is not antisymmetric, and hence it is not a partial order. Indeed, $\mu \preccurlyeq \nu$ and $\nu \preccurlyeq \mu$ do not imply $\mu = \nu$, but only $\mu \rightleftharpoons \nu$. Considering the *groups-into* relation, $\mu \trianglelefteq \nu$ ensures that for each granule of μ there exists a set of granules of ν covering exactly the same span of time. The relation is useful, for example, in applications where attribute values are associated with time granules; sometimes it is possible to obtain the value associated with a granule of ν from the values associated with the granules of μ whose union covers the same time. The groups-into relation has the same two properties as the finer-than relation, but generally $\mu \preccurlyeq \nu$ does not imply $\mu \trianglelefteq \nu$ or viceversa. The *subtype* relation intuitively identifies a type corresponding to subsets of granules of another type. Similar to the two previous relations, subtype is reflexive and transitive, and satisfies $\mu \sqsubseteq \nu \Rightarrow \mu \preccurlyeq \nu$. Finally, *shifting* is clearly an equivalence relation. Concerning this last relation, an equivalent, more useful and practical definition, is:

Definition 4.3 *Two temporal types μ_1 and μ_2 are said to be* shifting equivalent *(denoted $\mu_1 \rightleftharpoons \mu_2$) if there is a bijective function $h : \mathcal{T} \to \mathcal{T}$ such that $\mu_1(i) = \mu_2(h(i))$, for all $i \in \mathcal{T}$.*

In the following we consider only temporal type systems which satisfy the restriction that no pair of different types can be shifting equivalent, i.e.

$$\mu_1 \rightleftharpoons \mu_2 \Rightarrow \mu_1 = \mu_2. \tag{4.2}$$

For this class of systems, the three relationships $\preccurlyeq, \trianglelefteq$ and $\sqsubseteq$ are reflexive, transitive and antisymmetric and, hence, each relationship is a partial order. Therefore, for the relation

we are particulary interested in, *finer-than*, there exists a unique least upper bound of the set of all temporal types, denoted by $\mu_\top$, and a unique greatest lower bound, denoted by $\mu_\bot$. These top and bottom elements are defined as follows: $\mu_\top(i) = \mathcal{A}$ for some $i \in \mathcal{T}$ and $\mu_\top(j) = \emptyset$ for each $j \neq i$, and $\mu_\bot(i) = \emptyset$ for each $i \in \mathcal{T}$. Moreover, for each pair of temporal types μ_1, μ_2, there exist a unique least upper bound $\overline{(\mu_1, \mu_2)}$ and a unique greatest lower bound $\underline{(\mu_1, \mu_2)}$ of the two types, with respect to $\preccurlyeq$. We formalize this result in the following theorem, proved by Bettini et al. [1998a]:

Theorem 4.1 *Any temporal type system having an infinite index, and satisfying (4.2), is a lattice with respect to the finer-than relationship.*

Consider now temporal types for which the index set and the absolute time set are isomorphic with the set of positive natural numbers, i.e. $\mathcal{A} = \mathcal{T} = \mathbb{N}$. If we impose to any such temporal type μ the condition

$$\forall i \in \mathbb{N},\ 0 < \#\mu(i) \tag{4.3}$$

then we can prove that the condition (4.2) is a consequence of the condition (4.3).

Lemma 4.1 *If μ_1, μ_2 are temporal types on $(\mathbb{N}, \mathbb{N})$ satisfying 4.3, then $\mu_1 \rightleftharpoons \mu_2 \Rightarrow \mu_1 = \mu_2$.*

Proof: Before we start, we introduce the following notation: given two non-empty sets S_1 and S_2 of elements in $\mathcal{A}$, $S_1 \ll S_2$ holds if each number in S_1 is strictly less than each number in S_2 (formally, $S_1 \ll S_2$ if $\forall x \in S_1\, \forall y \in S_2\, (x < y)$). Moreover, we say that a set $\mathbb{S}$ of non-empty sets of elements in $\mathcal{A}$ is *monotonic* if for each pair of sets S_1 and S_2 in $\mathbb{S}$ either $S_1 \ll S_2$ or $S_2 \ll S_1$.

The relation $\mu_1 \rightleftharpoons \mu_2$ is equivalent with the existence of a bijection function $h : \mathbb{N} \to \mathbb{N}$ such that $\mu_1(i) = \mu_2(h(i))$, for all i. We will prove by induction that $h(i) = i$, which is equivalent with $\mu_1 = \mu_2$.

- $i = 1$: suppose that $h(1) > 1$. If $a = \min(\mu_1(1))$ – the existence of a is ensured by the condition 4.3 – then $\mu_1(1) = \mu_2(h(1)) \Rightarrow a \in \mu_2(h(1))$. Because $1 < h(1)$ we have

$\mu_2(1) \ll \mu_2(h(1))$ (according to Definition 4.1) and so there is $b \in \mu_2(1)$ such that $b < a$. The inequality $1 < h(1)$ implies $h^{-1}(1) > 1$, and so $\mu_2(1) = \mu_1(h^{-1}(1)) \gg \mu_1(1)$. But the last inequality ($\gg$) is contradicted by the existence of $b \in \mu_1(h^{-1}(1))$ which is smaller than $a \in \mu_1(1)$. In conclusion, $h(1) = 1$.

- $i = n + 1$: from the induction hypothesis we have $h(i) = i, \forall i \leq n$. Supposing that $h(n + 1) \neq n + 1$, then the only possibility is $h(n + 1) > n + 1$. This last relation implies also $h^{-1}(n + 1) > n + 1$. Using a similar rationing as in the previous case (it's sufficient to replace 1 with $n + 1$), we obtain

$$\mu_1(n + 1) \ll \mu_1(h^{-1}(n + 1)) = \mu_2(n + 1) \ll \mu_2(h(n + 1)) = \mu_1(n + 1)$$

where each of the set from this relation are non-empty, according to 4.3. The contradiction of the hypothesis, in this case, means that $h(n + 1) = n + 1$ and, by induction principle, that $h(i) = i, \forall i \in \mathbb{N}$. ■

Therefore, the set of temporal types satisfying (4.3) (denoted $\mathcal{G}_0$) is a lattice with respect to the finer-than relationship. The temporal type system $\mathcal{G}_0$ is not closed, because $\underline{(\mu_1, \mu_2)}$ is not always in $\mathcal{G}_0$. By adding a supplementary condition to the temporal types from $\mathcal{G}_0$,

$$\forall i \in \mathbb{N}, \mu^{-1}(i) \neq \emptyset \tag{4.4}$$

where $\mu^{-1}(i) = \{j \in \mathbb{N} : i \in \mu(j)\}$, it can be shown that this temporal type system (denoted $\mathcal{G}_1$) is a closed system and has a unique greatest lower bound, $\mu_\perp(i) = i$, $\forall i \in \mathbb{N}$, but no least upper bound $\mu_\top$. Furthermore, the condition (4.4) is a sufficient condition for the equivalence of the relationships *finer-than* and *groups-into*, according to the following lemma:

Lemma 4.2 *If μ and ν are temporal types on $(\mathbb{N}, \mathbb{N})$ which satisfy the conditions (4.3) and (4.4) – in other words, are of type $\mathcal{G}_1$ – then $\mu \preccurlyeq \nu \Leftrightarrow \mu \trianglelefteq \nu$.*

Proof Let be $\mu \in \mathcal{G}_1, \nu \in \mathcal{G}_1$.

- $\mu \preccurlyeq \nu$: let $j_0 \in \mathbb{N}$. The condition (4.4) means that for all $k \in \nu(j_0)$, $\mu^{-1}(k) \neq \emptyset$ and so $S = \bigcup_{k\in\nu(j_0)}\{\mu^{-1}(k)\} \neq \emptyset$. It is evident, according to Definition 4.1, that the relation *finer-than* implies that for each $i \in \mathbb{N}$ there is a **unique** $j \in \mathbb{N}$ such that $\mu(i) \subseteq \nu(j)$. Consequently, if $\mu \preccurlyeq \nu$ and $\mu(i) \cap \nu(j) \neq \emptyset$ then $\mu(i) \subseteq \nu(j)$. Therefore, for all $i \in S$, $\mu(i) \subset \nu(j_0)$ which implies $\bigcup_{i\in S} \mu(i) \subset \nu(j_0)$ (a). At the same time, $\forall k \in \nu(j_0)$ we have $k \in \mu\left(\mu^{-1}(k)\right)$ which implies $\nu(j_0) \subseteq \bigcup_{i\in S} \mu(i)$ (b). From (a) and (b) we have $\nu(j_0) = \bigcup_{i\in S} \mu(i)$, which implies $\mu \trianglelefteq \nu$.

- $\mu \trianglelefteq \nu$: let $i_0 \in \mathbb{N}$ and let $k \in \mu(i_0)$. According to (4.4), there exists $j = \nu^{-1}(k)$. Because $\mu \trianglelefteq \nu$ there is a set S such that $\nu(j) = \bigcup_{i\in S} \mu(i)$. Because the sets $\mu(i), i \in S$ are disjunct and $k \in \nu(j) \cap \mu(i_0)$ we have $i_0 \in S$. Therefore, for each i_0 there is $j \in \mathbb{N}$ such that $\mu(i_0) \subseteq \nu(j)$, which implies $\mu \preccurlyeq \nu$. ■

4.2 Linear Granular Time Structure

If $M = (S, x, \mathbf{I})$ is a first-order linear time structure, then let the absolute time $\mathcal{A}$ be given by the sequence x, by identifying the time moment i with the state $s_{(i)}$ (on the i^{th} position in the sequence). If μ is a temporal type from $\mathcal{G}_1$, then the temporal granule $\mu(i)$ may be identified with the set $\{s_j \in S \mid j \in \mu(i)\}$. Therefore, the temporal type μ induces a new sequence, x_μ, defined as $x_\mu : \mathbb{N} \to 2^S$, $x_\mu(i) = \mu(i)$. (*Remark*: In the following the set $\mu(i)$ will be considered, depending of the context, either as a set of natural numbers, or as a set of states).

Consider now the linear time structure derived from M, $M_\mu = (2^S, x_\mu, \mathbf{I}^\mu)$. To be well defined, we must give the interpretation $\mathbf{I}^\mu_{\mu(i)}$, for each $i \in \mathbb{N}$. As for a fixed i the set $\mu(i)$ is a finite sequence of states, it defines (if all the states are complete states) a model $\tilde{M}_{\mu(i)}$ for M. Therefore the estimated general interpretation $I_{G(\tilde{M}_{\mu(i)})}$ is well defined and we consider, by definition, that for all temporal free formula p in L,

$$\mathbf{I}^\mu_{\mu(i)}(p) = I_{G(\tilde{M}_{\mu(i)})}(p) = \text{supp}(p, \tilde{M}_{\mu(i)}) \tag{4.5}$$

This interpretation is extended to any temporal formula in L according to the rule:

$$\mathbf{I}^{\mu}_{\mu(i)}(X_{k_1}p_1 \wedge \ldots \wedge X_{k_n}p_n) = \frac{1}{n}\sum_{j=1}^{n}\mathbf{I}^{\mu}_{\mu(i+k_j)}(p_j) \tag{4.6}$$

where p_i are temporal free formulae and $k_i \in \mathbb{Z}, i = 1 \ldots n$.

Definition 4.4 *If $M = (S, x, \boldsymbol{I})$ is a first-order linear time structure and μ is a temporal type from $\mathcal{G}_1$, then the linear granular time structure induced by μ on M is the triple $M_\mu = (2^S, x_\mu, \boldsymbol{I}^\mu)$, where $x_\mu : \mathbb{N} \to 2^S$, $x_\mu(i) = \mu(i)$ and $\boldsymbol{I}^\mu$ is a function that associates with almost each set of states $\mu(i)$ an interpretation $\boldsymbol{I}^\mu_{\mu(i)}$ according to the rules (4.5) and (4.6).*

Of a particular interest is the linear granular time structure induced by the greatest lower bound temporal type $\mu_\perp(i) = i$. In this case, $M_{\mu_\perp} = (S, x, \mathbf{I}^{\mu_\perp})$, where the only difference from the initial time structure M is at the interpretation level: for $p = P(t_1, \ldots, t_n)$ a formula in L, if the interpretation $\mathbf{I}_s(p)$ is a function defined on D^n with values in $\{true, false\}$ – giving so the meaning of truth – the interpretation $\mathbf{I}^{\mu_\perp}_s(p)$ is a function defined on D^n with values in $[0, 1]$ – giving so the degree of truth. The relation linking the two interpretations is given by $\mathbf{I}_s(p) = true$ if and only if $\mathbf{I}^{\mu_\perp}_s(p) = 1$. Indeed, supposing the state $s_{(i)}$ is a complete state, it defines the model $\tilde{M}_i = (i, s_{(i)})$ and we have, for p a temporal free formula,

$$\mathbf{I}^{\mu_\perp}_{\mu_\perp(i)}(p) = \mathrm{supp}(p, \tilde{M}_i) = \begin{cases} 1, & \text{if } \mathbf{I}_{s(i)}(p) = true, \\ 0, & \text{if } \mathbf{I}_{s(i)}(p) = false \end{cases} \tag{4.7}$$

For a formula $\pi = X_{k_1}p_1 \wedge \ldots \wedge X_{k_n}p_n$, we have $\mathbf{I}_{s_i}(\pi) = true$ iff $\forall j \in \{1 \ldots n\}, i + k_j \models p_j$, which is equivalent with

$$\mathrm{supp}(p_1, \tilde{M}_{i+k_1}) = \ldots = \mathrm{supp}(p_n, \tilde{M}_{i+k_n}) = 1 \Leftrightarrow \frac{1}{n}\sum_{j=1}^{n}\mathbf{I}^{\mu_\perp}_{\mu_\perp(i+k_j)}(p_j) = \mathbf{I}^{\mu_\perp}_{\mu_\perp(i)}(\pi) = 1.$$

Consequently, the linear granular time structure $M_{\mu_\perp}$ can be seen as an extension, at the interpretation level, of the classical linear time structure M.

4.2.1 Linking two Granular Time Structures

All the granular time structures induced by a temporal type have in common interpretations which take values in [0, 1] if applied on predicate symbols in L. This observation allows us to establish the relation linking the interpretations $\mathbf{I}^{\mu}$ and $\mathbf{I}^{\nu}$, from two linear granular time structures induced by μ and ν, when there exists a relationship *finer-than* between these two temporal types. According to the lemma 4.2, for each $i \in \mathbb{N}$ there is a subset $N_i \subset \mathbb{N}$ such that $\nu(i) = \bigcup_{j \in N_i} \mu(j)$. If p is a temporal free formula in L, then the interpretation $\mathbf{I}^{\nu}$ for p at $\nu(i)$ is the weighted sum of the interpretations $\mathbf{I}^{\mu}_{\mu(j)}(p)$, where $j \in N_i$. We formalize this result in the following theorem:

Theorem 4.2 *If μ, ν are temporal types from $\mathcal{G}_1$, such that $\mu \preccurlyeq \nu$, and $\boldsymbol{I}^{\mu}, \boldsymbol{I}^{\nu}$ are the interpretations from the induced linear time structures M_μ and M_ν on M, then for each $i \in \mathbb{N}$,*

$$\boldsymbol{I}^{\nu}_{\nu(i)}(p) = \frac{1}{\#\nu(i)} \sum_{j \in N_i} \#\mu(j) \boldsymbol{I}^{\mu}_{\mu(j)}(p), \tag{4.8}$$

where N_i is the subset of $\mathbb{N}$ which satisfies $\nu(i) = \bigcup_{j \in N_i} \mu(j)$ and p is a temporal free formula in L.

Proof: The formula p being a temporal free formula, we have $w(p) = 0$. According to Def. 4.4 and Def. 3.18, we have

$$\mathbf{I}^{\nu}_{\nu(i)}(p) = \mathrm{supp}(p, \tilde{M}_{\nu(i)}) = \frac{\#\{j \in \nu(i) \mid j \models p\}}{\#\nu(i)} \quad (a)$$

On the other hand, because $\nu(i) = \bigcup_{j \in N_i} \mu(j)$, we have also

$$\frac{1}{\#\nu(i)} \sum_{j \in N_i} \#\mu(j) \mathbf{I}^{\mu}_{\mu(j)}(p) = \frac{1}{\#\nu(i)} \sum_{j \in N_i} \#\mu(j) \mathrm{supp}(p, \tilde{M}_{\mu(j)}) =$$
$$= \frac{1}{\#\nu(i)} \sum_{j \in N_i} \#\{k \in \mu(j) \mid k \models p\} = \frac{\#\{j \in \nu(i) \mid j \models p\}}{\#\nu(i)} \quad (b)$$

From (a) and (b) we obtain (4.8). ■

If we consider $\mu = \mu_\perp$ then $\#\mu(j) = 1$, for all $j \in \mathbb{N}$ and $\mathbf{I}^{\mu}_{\mu(j)}(p) = \text{supp}(p, \tilde{M}_j)$. Therefore,

$$\mathbf{I}^{\nu}_{\nu(i)}(p) = \frac{1}{\#\nu(i)} \sum_{j \in \nu(i)} \text{supp}(p, \tilde{M}_j) = \frac{1}{\#\nu(i)} \#\{j \in \nu(i) \mid j \models p\} = \text{supp}(p, \tilde{M}_{\nu(i)}) = \mathbf{I}_{G(\tilde{M}_{\nu(i)})}(p)$$

result which is consistent with the definition 4.4. But the significance of the theorem 4.2 is revealed in a particular context. Firstly, let $\mathcal{G}_2$ be the subset of $\mathcal{G}_1$, obtained when the condition (4.3) is replaced by the stronger condition (4.3'), $\#\mu(i) = c_\mu$, where $c_\mu \in \mathbb{N}$. If $\mu, \nu \in \mathcal{G}_2$ and $\mu \preccurlyeq \nu$, it can be shown that $\#N_i = \frac{c_\nu}{c_\mu}, \forall i \in \mathbb{N}$ and so the relation (4.8) becomes

$$\mathbf{I}^{\nu}_{\nu(i)}(p) = \frac{1}{\#N_i} \sum_{j \in N_i} \mathbf{I}^{\mu}_{\mu(j)}(p). \tag{4.9}$$

Generally speaking, consider three worlds, W_1, W_2 and W_3 – defined as sets of granules of information – where W_1 is finer than W_2 which is finer than W_3. Suppose also that the conversion between granules from two different worlds is given by a constant factor. If the **independent** part of information in each granule is transferred from W_1 to W_2 and then the world W_1 is "lost", the theorem 4.2 under the form (4.9) affirms that it is possible to transfer the independent information from W_2 to W_3 and to obtain the same result as for the transfer from W_1 to W_3.

Example 4.1 : Consider a linear time structure M (here, the world W_1) and a temporal free formula p such that, for the first six time moments, we have $i \models p$ for $i \in \{1, 3, 5, 6\}$. The concept of independence, in this example, means that the interpretation of p in the state s_i does not depend on the interpretation of p in the state s_j. Let be $\mu, \nu \in \mathcal{G}_2$, $\mu \preccurlyeq \nu$, with $\mu(i) = \{2i-1, 2i\}$ and $\nu(i) = \{6i-5, \ldots, 6i\}$. Therefore, $\nu(1) = \mu(1) \cup \mu(2) \cup \mu(3)$. According to the definition 4.4, $\mathbf{I}^{\mu}_{\mu(1)}(p) = \text{supp}(p, \{1, 2\}) = 0.5$, $\mathbf{I}^{\mu}_{\mu(2)}(p) = \text{supp}(p, \{3, 4\}) = 0.5$, $\mathbf{I}^{\mu}_{\mu(3)}(p) = \text{supp}(p, \{5, 6\}) = 1$, whereas $\mathbf{I}^{\nu}_{\nu(1)}(p) = \text{supp}(p, \{1, .., 6\}) = 0.66$. If the linear time structure M is "lost", the temporal types μ and ν are "lost" too (we don't know the absolute time $\mathcal{A}$ given by M). But if we know the induced time structure M_μ (world W_2) and the relation between μ and ν

$$\nu(k) = \mu(3k-2) \cup \mu(3k-1) \cup \mu(3k), \ \forall k \in \mathbb{N}$$

then we can completely deduce the time structure M_ν (world W_3). As an example, according to (4.9), $\mathbf{I}^\nu_{\nu(1)}(p) = \frac{1}{3}\sum_{i=1}^{3}\mathbf{I}^\mu_{\mu(i)}(p) = 0.66$. The condition about a constant conversion factor between granules is necessary because of the size of granules, as they appear in expression 4.8, are "lost" when the time structure M is "lost".

The theorem 4.2 is not effective for temporal formulae (which can be seen as the dependent part of the information of a temporal granule). In this case we will show that the interpretation, in the coarser world, of a temporal formula with a given time window is linked with the interpretation, in the finer world, of a similar formula but having a larger time window.

Theorem 4.3 *If μ and ν are temporal types from $\mathcal{G}_2$ such that $\mu \preccurlyeq \nu$ and $\boldsymbol{I}^\mu, \boldsymbol{I}^\nu$ are the interpretations from the induced linear time structures M_μ and M_ν on M, then for each $i \in \mathbb{N}$,*

$$\boldsymbol{I}^\nu_{\nu(i)}(p \wedge Xq) = \frac{1}{k}\sum_{j\in N_i}\boldsymbol{I}^\mu_{\mu(j)}(p \wedge X_k q) \tag{4.10}$$

where $k = c_\nu/c_\mu$, $\nu(i) = \bigcup_{j\in N_i}\mu(j)$ and p, q are temporal free formulae in L.

Proof: If $\mu, \nu \in \mathcal{G}_2$ such that $\#\mu(i) = c_\mu$ and $\#\nu(i) = c_\nu$, for all $i \in \mathbb{N}$, it is easy to show that the sets N_i satisfying $\nu(i) = \bigcup_{j\in N_i}\mu(j)$ have all the same cardinality, $\#N_i = c_\nu/c_\mu = k$ and contain successive natural numbers, $N_i = \{j_i, j_i+1, \ldots, j_i+k-1\}$. From the relations (4.6) and (4.9) we have:

$$\begin{aligned}
\mathbf{I}^\nu_{\nu(i)}(p \wedge Xq) &= \frac{1}{2}\left(\mathbf{I}^\nu_{\nu(i)}(p) + \mathbf{I}^\nu_{\nu(i+1)}(q)\right) = \frac{1}{2}\left(\frac{1}{\#N_i}\sum_{j\in N_i}\mathbf{I}^\mu_{\mu(j)}(p) + \frac{1}{\#N_{i+1}}\sum_{j\in N_{i+1}}\mathbf{I}^\mu_{\mu(j)}(q)\right) = \\
&= \frac{1}{2}\left(\frac{1}{k}\sum_{j=j_i}^{j_i+k-1}\mathbf{I}^\mu_{\mu(j)}(p) + \frac{1}{k}\sum_{j=j_i+k}^{j_i+2k-1}\mathbf{I}^\mu_{\mu(j)}(q)\right) = \frac{1}{2k}\left(\sum_{j=j_i}^{j_i+k-1}\left(\mathbf{I}^\mu_{\mu(j)}(p) + \mathbf{I}^\mu_{\mu(j+k)}(q)\right)\right) = \\
&= \frac{1}{2k}\left(\sum_{j=j_i}^{j_i+k-1}2\mathbf{I}^\mu_{\mu(j)}(p \wedge X_k q)\right) = \frac{1}{k}\sum_{j\in N_i}\mathbf{I}^\mu_{\mu(j)}(p \wedge X_k q)\ \blacksquare.
\end{aligned}$$

If we define the operator $zoom_k$ over the set of formulae in L as

$$zoom_k(X_{k_1}p_1 \wedge \ldots \wedge X_{k_n}p_n) = X_{k\cdot k_1}p_1 \wedge \ldots \wedge X_{k\cdot k_n}p_n$$

then an obvious corollary of this theorem is

Corollary 4.1 *If μ and ν are temporal types from $\mathcal{G}_2$ such that $\mu \preccurlyeq \nu$ and $\boldsymbol{I}^{\mu}, \boldsymbol{I}^{\nu}$ are the interpretations from the induced linear time structures M_μ and M_ν on M, then for each $i \in \mathbb{N}$,*

$$\boldsymbol{I}^{\nu}_{\nu(i)}(X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n) = \frac{1}{k} \sum_{j \in N_i} \boldsymbol{I}^{\mu}_{\mu(j)}(zoom_k(X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n)) \tag{4.11}$$

where $k = c_\nu / c_\mu$, $\nu(i) = \bigcup_{j \in N_i} \mu(j)$, $k_i \in \mathbb{N}$ and $p_i, i = 1..n$ are temporal free formulae in L.

According to this corollary, if we know the degree of truth of a temporal rule (template) in the world W_1, we can say nothing about the degree of truth of the same rule in the world W_2, coarser than W_1. The information is only transferred from the temporal rule $zoom_k(H)$ in W_1 (which has a time window greater than $k-1$) to the temporal rule H in W_2, where k is the coefficient of conversion between the two worlds. Consequently, all the information related to temporal formulae having a time window less than k is lost during the transition to the coarser world W_2.

4.2.2 The Consistency Problem

The importance of the concepts of consistency, support and confidence, (see Chapter 3), for the process of information transfer between worlds with different granularity may be highlighted by analyzing the analogous expressions for a linear granular time structure M_μ.

Definition 4.5 *Given L and a linear granular time structure M_μ on M, we say that M_μ is a consistent granular time structure for L if, for every formula p, the limit*

$$supp(p, M_\mu) = \lim_{n \to \infty} \frac{\sum_{i=1}^{n} \boldsymbol{I}^{\mu}_{\mu(i)}(p)}{n} \tag{4.12}$$

exists. The notation $supp(p, M_\mu)$ denotes the support (degree of truth) of p under M_μ.

A natural question concerns the inheritance of the consistency property from the basic linear time structure M by the induced time structure M_μ. The answer is formalized in the following theorem.

Theorem 4.4 *If M is a consistent linear time structure and $\mu \in \mathcal{G}_2$ then the granular time structure M_μ is also consistent.*

Proof M being a consistent time structure, for each formula p in L there is the limit of the sequence $x(p)_n = n^{-1}\#\{i \leq n \,|\, i \models p\}$ and $\lim_{n\to\infty} x(p)_n = \mathrm{supp}(p, M)$. At the same time, $\mu \in \mathcal{G}_2$ implies $\#\mu(i) = k$ for all $i \in \mathbb{N}$ and $\mu(i) = \{k(i-1)+1, k(i-1)+2, \ldots, ki\}$. Consider the following two cases:

- *p temporal free formula* : We have

$$\frac{\sum_{i=1}^n \mathbf{I}^\mu_{\mu(i)}(p)}{n} = \frac{\sum_{i=1}^n \mathrm{supp}(p, M_{\mu(i)})}{n} = \\ = \frac{\sum_{i=1}^n \frac{\#\{j\in\mu(i)\,|\,j\models p\}}{\#\mu(i)}}{n} = \frac{\sum_{i=1}^n \#\{j \in \mu(i) \,|\, j \models p\}}{kn} = \\ = \frac{\#\{i \in \bigcup_{i=1}^n \mu(i) \,|\, i \models p\}}{kn} = \frac{\#\{i \leq kn \,|\, i \models p\}}{kn} = x(p)_{kn}$$

Therefore, there exists the limit $\lim_{n\to\infty} \frac{\sum_{i=1}^n \mathbf{I}^\mu_{\mu(i)}(p)}{n} = \lim_{n\to\infty} x(p)_{kn}$ and therefore we have

$$\mathrm{supp}(p, M_\mu) = \mathrm{supp}(p, M) \text{ for } p \text{ temporal free formula} \tag{4.13}$$

- *temporal formula $\pi = X_{k_1}p_1 \wedge \ldots \wedge X_{k_m}p_m$*: We have

$$\frac{\sum_{i=1}^n \mathbf{I}^\mu_{\mu(i)}(X_{k_1}p_1 \wedge \ldots \wedge X_{k_m}p_m)}{n} = \frac{\sum_{i=1}^n \left(m^{-1}\sum_{j=1}^m \mathbf{I}^\mu_{\mu(i+k_j)}(p_j)\right)}{n} = \\ = \frac{1}{m}\frac{\sum_{i=1}^n\sum_{j=1}^m \mathrm{supp}(p_j, M_{\mu(i+k_j)})}{n} = \frac{1}{mn}\sum_{j=1}^m\sum_{i=1}^n \mathrm{supp}(p_j, M_{\mu(i+k_j)}) = \\ = \frac{1}{mn}\sum_{j=1}^m\sum_{i=1}^n \frac{\#\{h \in \mu(k_j+i) \,|\, h \models p_j\}}{k} = \frac{1}{mn}\sum_{j=1}^m \frac{\#\{h \in \bigcup_{i=1}^n \mu(k_j+i) \,|\, h \models p_j\}}{k} = \\ = \frac{1}{mnk}\sum_{j=1}^m \left(\#\{h \leq k(k_j+n) \,|\, h \models p_j\} - \#\{h \leq kk_j \,|\, h \models p_j\}\right) = \\ = \frac{1}{mnk}\sum_{j=1}^m \left(k(k_j+n)x(p_j)_{k(k_j+n)} - kk_j x(p_j)_{kk_j}\right) = \\ = \frac{1}{m}\sum_{j=1}^m \frac{k_j+n}{n}x(p_j)_{k(k_j+n)} - \frac{1}{m}\sum_{j=1}^m \frac{k_j}{n}x(p_j)_{kk_j}$$

By tacking $n \to \infty$ in the last relation, we obtain

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} \mathbf{I}^{\mu}_{\mu(i)}(X_{k_1}p_1 \wedge \ldots \wedge X_{k_m}p_m)}{n} =$$

$$= \lim_{n\to\infty}\left(\frac{1}{m}\sum_{j=1}^{m}\frac{k_j+n}{n}x(p_j)_{k(k_j+n)} - \frac{1}{m}\sum_{j=1}^{m}\frac{k_j}{n}x(p_j)_{kk_j}\right) =$$

$$= \frac{1}{m}\sum_{j=1}^{m}\lim_{n\to\infty}\frac{k_j+n}{n}x(p_j)_{k(k_j+n)} - \frac{1}{m}\sum_{j=1}^{m}\lim_{n\to\infty}\frac{k_j}{n}x(p_j)_{kk_j} =$$

$$= \frac{1}{m}\sum_{j=1}^{m}\lim_{n\to\infty}x(p_j)_{k(k_j+n)} = \frac{1}{m}\sum_{j=1}^{m}\mathrm{supp}(p_j, M)$$

and so we have

$$\mathrm{supp}(X_{k_1}p_1 \wedge \ldots \wedge X_{k_m}p_m, M_\mu) = \frac{1}{m}\sum_{j=1}^{m}\mathrm{supp}(p_j, M) \tag{4.14}$$

From 4.2.2 and 4.14 results the conclusion of the theorem ■.

The implications of Theorem 4.4 are extremely important. According to Corollary 3.5, the confidence of a temporal rule (template) may be expressed using only the support measure if the linear time structure M is consistent. Therefore, considering that by definition the confidence of a temporal rule (template) $\mathcal{H}$, $H_1 \wedge \ldots \wedge H_m \mapsto H_{m+1}$, giving a consistent granular time structure M_μ, is

$$conf(\mathcal{H}, M_\mu) = \begin{cases} \frac{\mathrm{supp}(H_1\wedge\ldots\wedge H_m\wedge H_{m+1}, M_\mu)}{\mathrm{supp}(H_1\wedge\ldots\wedge H_m, M_\mu)} & \text{if } \mathrm{supp}(H_1 \wedge \ldots \wedge H_m, M_\mu) > 0, \\ 0 & \text{if not} \end{cases} \tag{4.15}$$

we can deduce, by applying Theorem 4.4, that the confidence of $\mathcal{H}$, for a granular time structure M_μ induced on a consistent time structure M by a temporal type $\mu \in \mathcal{G}_2$, is independent of μ. In other words,

The property of consistency is a sufficient condition for the independence of the measures of support and of confidence, during the process of information transfer between worlds with different granularities, but all derived from an absolute world using constant conversion factors.

4.2.3 Event Aggregation

All the deduction processes made until now were conducted to obtain an answer to the following question: how is the degree of truth of a formula p changing if we pass from a linear time structure with a given granularity to a coarser one. And we proved that we can give a proper expression if we impose some restrictions on the temporal types which induce these time structures. But there is another phenomenon which follows the process of transition between two real worlds with different time granularities: new kinds of events appear, some kinds of events disappear.

Definition 4.6 *An event type (denoted $E[t]$) is the set of all temporal atoms from L having the same name (or head).*

All the temporal atoms of a given type $E[t]$ are constructed using the same symbol predicate and we denote by $N[t]$ the arity of this symbol. Consider $E(t, t_2, \ldots, t_n) \in E[t]$ (where $n = N[t]$). According to the definition 3.2, a term $t_i, i \in \{2, .., n\}$ has the form $t_i = f(t_{i1}, \ldots, t_{ik_i})$. Suppose now that for each index i the function symbol f from the expression of t_i belongs to a family of function symbols with different arities, denoted $\mathcal{F}_i[t]$ (so different sets for different event types $E[t]$ and different index i). This family has the property that the interpretation for each of its member is given by a real functions which

- is applied on a variable number of arguments, and
- is invariant in the order of the arguments.

A good example of a such real function is a statistical function, e.g. $\text{mean}(x_1, \ldots, x_n)$. Based on the set $\mathcal{F}_i[t]$ we construct the set of terms expressed as $f_k(c_1, \ldots, c_k)$, where f_k is a $k-$ary function symbol from $\mathcal{F}_i[t]$ and $c_i, i = 1..k$ are constant symbols. We denote such a set as $T_i[t]$. Consider now the operator $\oplus$ defined on $T_i[t] \times T_i[t] \to T_i[t]$ such that

$$f_n(c_1, .., c_n) \oplus f_m(d_1, .., d_m) = f_{n+m}(c_1, .., c_n, d_1, .., d_m)$$

Of course, because the interpretation of any function symbol from $\mathcal{F}_i[t]$ is invariant in the order of arguments, we have

$$f_n(c_1, \ldots, c_n) \oplus f_m(d_1, \ldots, d_m) = f_n(c_{\sigma(1)}, \ldots, c_{\sigma(n)}) \oplus f_m(d_{\varphi(1)}, \ldots, c_{\varphi(n)})$$

where σ (respectively φ) is a permutation of the set $\{1, \ldots, n\}$ (respectively $\{1, \ldots, m\}$). Furthermore, it is evident that the operator $\oplus$ is commutative and associative.

We introduce now a new operator (denoted $\boxplus$) defined on $E[t] \times E[t] \to E[t]$, such that, for $E(t, t_2, .., t_i, .., t_n) \in E[t]$, $E(t, t'_2, .., t'_i, .., t'_n) \in E[t]$ we have:

$$E(t, t_2, \ldots, t_i, \ldots, t_n) \boxplus E(t, t'_2, \ldots, t'_i, \ldots, t'_n) = E(t, t_1 \oplus t'_1, \ldots, t_i \oplus t'_i, \ldots, t_n \oplus t'_n) \quad (4.16)$$

Once again, it is evident that the operator $\boxplus$ is commutative and associative. Therefore, we can apply this operator on a subset $\mathcal{E}$ of temporal atoms from $E[t]$ and we denote the result as $\boxplus_{e_i \in \mathcal{E}} e_i$.

If $M = (S, x, \mathbf{I})$ is a linear time structure, for each event type $E[t]$ we define the subset $E[t]_M$ satisfying the condition:

$$E[t]_M = \{e \in E[t] \mid \exists s_i \in x \text{ such that } \mathbf{I}_{s_i}(e) = true\} \quad (4.17)$$

In a similar manner we define $E[t]_{\tilde{M}}$, where $\tilde{M} = (\tilde{T}, \tilde{x})$ is a model for M:

$$E[t]_{\tilde{M}} = \{e \in E[t] \mid \exists s_i \in \tilde{x} \text{ such that } \mathbf{I}_{s_i}(e) = true\} \quad (4.18)$$

In other words, $E[t]_M$ is the set of all temporal events of type $E[t]$ which are satisfied by M, whereas $E[t]_{\tilde{M}} \subseteq E[t]_M$ is the set of events from $E[t]$ satisfied by $\tilde{M}$. If we consider now M_μ, the linear time structure induced by the temporal type μ on M, the definition of $E[t]_{M_\mu}$ is derived from (4.17) only by changing the condition $\mathbf{I}_{s_i}(e) = true$ in $\mathbf{I}^{\mu}_{\mu(i)}(e) = 1$. Of course, only for $\mu = \mu_\perp$ we have $E[t]_M = E[t]_{M_\mu}$ (we proved that $\mathbf{I}_{s_i}(p) = true \Leftrightarrow \mathbf{I}^{\mu_\perp}_{\mu_\perp(i)}(p) = 1$). Generally $E[t]_M \supset E[t]_{M_\mu}$, which is a consequence of the fact that a coarser world satisfies less temporal events than a finer one.

Example 4.2 : If M is a linear time structure such that for the event $e \in E[t]$ we have $i \models e$ if and only if i is odd, and μ is a temporal type given by $\mu(i) = \{2i-1, 2i\}$, then it is obvious that $e \in E[t]_M$ but $e \notin E[t]_{M_\mu}$ (for all $i \in \mathbb{N}, \mathbf{I}^{\mu}_{\mu(i)}(e) = \text{supp}(e, \{2i-1, 2i\}) = 0.5$).

At the same time a coarser world may satisfy new events, representing a kind of aggregation of local, "finer" events.

Definition 4.7 *If μ is a temporal type from $\mathcal{G}_1$, we call the aggregate event of type t induced by the granule $\mu(i)$ (denoted $e[t]_{\mu(i)}$) the event obtained by applying the operator $\boxplus$ on the set of events of type t which are satisfied by the model $\tilde{M}_{\mu(i)}$, i.e.*

$$e[t]_{\mu(i)} = \boxplus_{e_i \in E[t]_{\tilde{M}_{\mu(i)}}} e_i \tag{4.19}$$

If an event $e \in E[t]$ is not satisfied by M (or $e \notin E[t]_M$), it is obvious that according to (4.5) $\mathbf{I}^{\mu}_{\mu(i)}(e) = 0$, for all μ and all $i \in \mathbb{N}$. Therefore, the relation (4.5) is not appropriate to give the degree of truth for $e[t]_{\mu(i)}$. Before giving the rule expressing the interpretation for an aggregate temporal atom, we impose the following restriction on M: there is a one-to-one relationship between the set of events satisfied by M and the set of states, or

$$\exists\, h : \bigcup_t E[t]_M \to S,\ h(e) = s \text{ where } \mathbf{I}_s(e) = true \tag{4.20}$$

We define the interpretation of the aggregate event induced by $\mu(i_0)$, evaluated at $\mu(i)$, as:

$$\mathbf{I}^{\mu}_{\mu(i)}(e[t]_{\mu(i_0)}) = \sum_{e_j \in \mathcal{E}} \mathbf{I}^{\mu}_{\mu(i)}(e_j) \tag{4.21}$$

where $\mathcal{E} = E[t]_{\tilde{M}_{\mu(i_0)}}$. The restriction (4.20) is sufficient to assure that $\mathbf{I}^{\mu}_{\mu(i)}(e[t]_{\mu(i_0)}) \leq 1$, for all $i, i_0 \in \mathbb{N}$. Indeed, let $e_1, \ldots, e_n$ be the events from $\mathcal{E}$. If $h(e_j) = s_j$, $j = 1..n$, then consider the sets $A_j = \{k \in \mu(i) \,|\, k \models e_j\} = \{s \in \mu(i) \,|\, s = s_j\}$. The function h being injective, the sets A_j are disjoint and therefore $\sum_{j=1}^{n} \#A_j \leq \#\mu(i)$. Consequently, we have

$$\mathbf{I}^{\mu}_{\mu(i)}(e[t]_{\mu(i_0)}) = \sum_{j=1}^{n} \mathbf{I}^{\mu}_{\mu(i)}(e_j) = \sum_{j=1}^{n} \text{supp}(e_j, M_{\mu(i)}) = \frac{1}{\#\mu(i)} \sum_{j=1}^{n} \#A_j \leq \frac{\#\mu(i)}{\#\mu(i)} = 1.$$

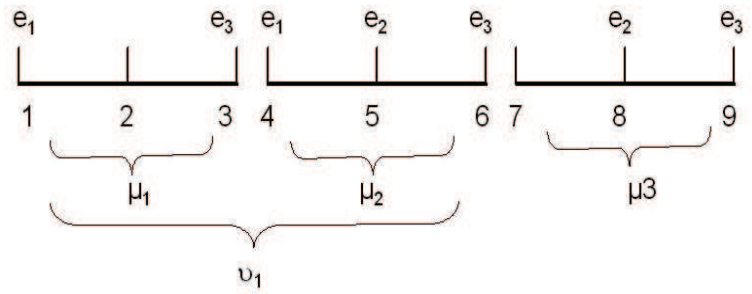


Figure 8: Graphical representation of the first nine states from the time structure M and of the firsts granules of temporal types μ and ν

Furthermore, $e[t]_{\mu(i_0)} \in E[t]_{M_\mu}$ if and only if there is $i \in \mathbb{N}$ such that the sets A_j form a partition of $\mu(i)$ (or equivalently $h^{-1}(\mu(i)) = \mathcal{E}$). Practically, this means that an aggregate event of type t is satisfiable if there is a granule $\mu(i)$ such that if $j \in \mu(i)$ and $j \models e$ then $e \in E[t]$.

Example 4.3 Let be M a linear time structure, $e_1, e_2, e_3 \in E[t]$ such that (see Fig. 8)

$$i \models e_1 \text{ for } i = 3k + 1 \text{ and } k \in \{0, 1, 3, 4, 6, 7, ..\}$$

$$i \models e_2 \text{ for } i = 3k + 2 \text{ and } k \in \{1, 2, 4, 5, 7, 8, ..\}$$

$$i \models e_3 \text{ for } i = 3k + 3 \text{ and } k \in \{0, 1, 2, 3, 4, 5, ..\}$$

(*Remark*: restriction (4.20) means that, if for example $h(e_3) = s_3$, then $s_{(3k)} = s_3$ for all $k \in \mathbb{N}$). Consider two temporal types $\mu, \nu \in \mathcal{G}_2$ such that $\mu(i) = \{3i - 2, 3i - 1, 3i\}$ and $\nu(i) = \{6i - 5, \ldots, 6i\}$. The different aggregate events induced by granules of temporal type μ are:

$$e[t]_{\mu(1)} = e_1 \boxplus e_2,\ e[t]_{\mu(2)} = e_1 \boxplus e_2 \boxplus e_3,\ e[t]_{\mu(3)} = e_2 \boxplus e_3$$

$$\mathbf{I}^{\mu}_{\mu(1)}(e[t]_{\mu(1)}) = 2/3,\ \mathbf{I}^{\mu}_{\mu(2)}(e[t]_{\mu(1)}) = 2/3,\ \mathbf{I}^{\mu}_{\mu(3)}(e[t]_{\mu(1)}) = 1/3$$

$$\mathbf{I}^{\mu}_{\mu(1)}(e[t]_{\mu(2)}) = 2/3,\ \mathbf{I}^{\mu}_{\mu(2)}(e[t]_{\mu(2)}) = 1,\ \mathbf{I}^{\mu}_{\mu(3)}(e[t]_{\mu(2)}) = 2/3$$

$$\mathbf{I}^{\mu}_{\mu(1)}(e[t]_{\mu(3)}) = 1/3,\ \mathbf{I}^{\mu}_{\mu(2)}(e[t]_{\mu(1)}) = 2/3,\ \mathbf{I}^{\mu}_{\mu(3)}(e[t]_{\mu(1)}) = 2/3$$

There is a single aggregate event induced by a granule of temporal type ν, which is $e[t]_{\nu(i)} = e_1 \boxplus e_2 \boxplus e_3$, for all $i \in \mathbb{N}$. But the interpretation of this event (lets call it e_{123}) depends on

the granule $\nu(j)$, according to the rule

$$\mathbf{I}^{\nu}_{\nu(i)}(e_{123}) = \begin{cases} 5/6 & \text{for } i \text{ odd} \\ 4/6 & \text{for } i \text{ even.} \end{cases}$$

Evidently, e_1, e_2 and e_3 are all included in $E[t]_M$ (events satisfiable in M), but none of them are satisfiable in M_μ, because $\mathbf{I}^{\mu}_{\mu(i)}(e_j) = 1/3$, for all $i \in \mathbb{N}$ and $j = 1..3$. Among the aggregated events induced by M_μ, only $e_1 \boxplus e_2 \boxplus e_3 \in E[t]_{M_\mu}$ (there is $i = 2$ such that $\mathbf{I}^{\mu}_{\mu(2)}(e_1 \boxplus e_2 \boxplus e_3)) = 1$). Finally, none of the initial events or aggregate events induced by μ or ν are not satisfied by M_ν.

4.3 Summary

Starting from the inherent behavior of temporal systems – the perception of events and of their interactions is determined, in a large measure, by the temporal scale – we extended the capability of our formalism to "capture" the concept of time granularity. To keep an unitary viewpoint on the meaning of the same formula at different scales of time, we changed the usual definition of the interpretation $\mathbf{I}^{\mu}$ for a formula in the frame of a first-order temporal granular logic: it return the degree of truth (a real value between zero and one) and not only the meaning of truth (*true* or *false*).

The consequence of the definition for $\mathbf{I}^{\mu}$ is formalized in Theorem 4.2 : only the independent information (here, the degree of truth for a temporal free formula) may be transferred without loss between worlds with different granularities. Concerning the temporal rules (scale dependent information), we proved that the interpretation of a rule in a coarser world is linked with the interpretation of a similar rule in a finer world, rule obtained by applying the operator $zoom_k$ on the initial temporal rule.

By defining a similar concept of consistency for a granular time structure M_μ, we could prove that this property is inherited from the basic time structure M if the temporal type μ is of type $\mathcal{G}_2$ (granules with constant size). The major consequence of Theorem 4.4 is

that the confidence of a temporal rule (template) is preserved in all granular time structures derived from a same consistent time structure.

We defined also a mechanism to aggregate events of the same type, that reflects the following intuitive phenomenon: in a coarser world, not all events inherited from a finer world are satisfied, but in exchange there are new events which become satisfiable. To achieve this we extended the syntax and the semantics of L by allowing a "family" of function symbols and by adding two new operators.

In our opinion, the logical next step in our work consists in adding a probabilistic dimension to the formalism. The results in the next chapter confirm that this approach allows a unified framework for the initial formalism and its granular extension, framework in which many of the defined concepts become consequences of the properties of a fundamental stochastic structure.

CHAPTER V

A PROBABILISTIC APPROACH

First-order logic is widely recognized as being a fundamental building block in knowledge representation. However, first-order logic does not have the expressive power to deal with many situations of interest, especially those related with uncertainty [Koller and Halpern, 1996]. If the uncertainty is a fundamental and irreducible aspect of our knowledge about the world, the probability is the most well-understood and widely applied logic for computational scientific reasoning under uncertainty. However, its applicability has been limited by the lack of a coherent semantics for plausible reasoning. A theory in first-order logic assigns definite truth-values only to sentences that have the same truth-value (either true or false) in all interpretations of the theory. The most that can be said about any other sentence is that its truth-value is indeterminate [Laskey, 2004].

5.1 Probabilistic Logics

Among the many proposed logics for plausible inference, probability is the strongest contender as a universal representation for translating among different plausible reasoning logics. There are numerous arguments in favor of probability as a rationally justified calculus for plausible inference under uncertainty [de Finetti, 1974-75, Howson and Urbach, 1993, Jaynes, 2003]. In 1854 already Boole published his work *The Laws of Thought* in which he described, among other things, the key concepts behind the idea of probabilistic inference. These ideas formed the foundations for most of the subsequent probabilistic logics . Until recently, the development of a fully general probabilistic logic was hindered by the lack of modularity of probabilistic reasoning, the intractability of worst-case probabilistic inference, and the difficulty of ensuring that a set of probability assignments specifies a

unique and well-defined probability distribution. Probability is not truth-functional. That is, the probability of a compound expression cannot be expressed solely as a function of the probabilities of its constituent expressions. The number of probabilities required to express a fully general probability distribution over truth-values of a collection of assertions is exponential in the number of assertions, making a brute-force approach to specification and inference infeasible for all but the smallest problems.

Although work relating first-order logic and probability goes back to Carnap [1950], there has been relatively little work on providing formal first-order logics for reasoning about probability. Gaifman [1964] and Scott and Krauss [1966] considered the problem of associating the probabilities with classical first-order statements (which, as pointed out in Bacchus [1988], essentially correspond to putting probabilities on possible worlds). Los [1963] and Fenstad [1967] studied this problem as well, but allowed values for free variables to be chosen according to a probability on a domain. Keisler [1985] investigated an infinitary logic with a measure on the domain, and obtained completeness and compactness results. Feldman [1984] and Harel and Feldman [1984] considered a probabilistic dynamic logic, which extends first-order dynamic logic by adding probabilities. Bacchus [1990] provides a syntax and semantics for a first-order logic for reasoning about chance where the probability is placed on the domain. He also defined the notion of a *belief function* to be the degree of belief in a formula α given a knowledge base β. A very thorough study of probabilistic logics and their properties had been conducted by Halpern et al. [Halpern, 1989, Fagin, Halpern, and Megiddo, 1988, Fagin and Halpern, 1989, Halpern and Pucella, 2003]. The logics they proposed extended their modal logics of knowledge and belief [Fagin, Halpern, Moses, and Vardi, 1995] and it had been shown [Abadi and Halpern, 1994] that they cannot be finitely axiomatized. Another version of probabilistic logic is due to Nilsson [1986]. In his work he described the model theory for probabilistic inference and described a number of methods of computing the probabilities in his model. Nilsson also used possible worlds as a part of his model.

As was pointed out by Halpern [1989], there are two approaches to giving semantics to first-order logics of probability. The first approach puts a probability on the domain, and is appropriate for giving semantics to formulae involving statistical information such as "The probability that a randomly chosen bird flies is greater than 0.9". This approach can be viewed as a statement about what Hacking [1965] calls a *chance setup*, that is, about what one might expect as the result of performing some experiment or trial in a given situation. It can also be viewed as capturing statistical information about the world, (the unique and only possible "real" world). The second approach puts probability on possible worlds, and is appropriate for giving semantics to formulas describing what has been called a *degree of belief* [Bacchus, 1990, Kyburg, 1988], such as "The probability that Tweety (a particular bird) flies is greater than 0.9".

Even if these two approaches can be combined in one framework [Halpern, 1989], most of the logical frameworks for modelling uncertainty in machine learning and artificial intelligence, which incorporates knowledge, probability and time, are based on possible worlds. These frameworks are used, as example, to analyze planning [1] problems and prove the correctness of planning algorithms [Haddawy, 1996, Bacchus and Kabanza, 2000], to manage uncertain information in databases or temporal databases [Lakshmanan et al., 1997] or to develop a model theory, fixpoint theory and proof theory for temporal probabilistic logic programs [Dekhtyar et al., 1999a,b].

In the following we will briefly describe the logical framework proposed by Fagin et al. [1990] and Halpern [1998], which is the closest to our viewpoint. The language contains a fixed infinite set $\Phi = \{p_1, p_2, \ldots\}$ of *primitive propositions* or *basic events*. The set of *propositional formulas* or *events* is the closure of Φ under the Boolean operation $\wedge$ and $\neg$. The notation p denotes a primitive proposition, whereas φ denotes a propositional formula. A *primitive weight term* is an expression of the form $w(\varphi)$, where w is a special

[1] Planning is the process of formulating and choosing a set of actions which when executed would likely achieve a desirable outcome. Actions in a plan may be performed to affect the state of knowledge of the performing agent, to affect the state of the world, or simply for their own sake.

function which can be read as "the probability of". A *weight formula* is a statement of the form $w(\varphi) \geq \alpha$ or $w(\varphi) \leq \alpha$ or $w(\varphi) = \alpha$, where $\alpha \in [0, 1]$. The semantics is defined based on a probability structure space $M = (S, \mathcal{X}, \mu, \pi)$, where $(S, \mathcal{X}, \mu)$ is a probability space (see Appendix A) and π is a function which associates with each state in S a truth assignment on the primitive propositions in Φ. Thus $\pi(s)(p) \in \{\text{true, false}\}$ for each $s \in S$ and $p \in \Phi$. Therefore, for each $p \in \Phi$ the set $p^M = \{s \in S | \pi(s)(p) = \text{true}\}$ can be thought of as the possible worlds where p is true, or the states at which the event p occurs. Using the extension of the truth assignment to propositional formulas one obtains $\phi^M = \{s \in S | \pi(s)(\phi) = \text{true}\}$. If M is a measurable probability structure (i.e. all the sets p^M are measurable), then a formula $w(\phi) = \alpha$ is true if the probability of the set ϕ^M is α:

$$M \models w(\phi) = \alpha \text{ if } \mu(\phi^M) = \alpha.$$

5.2 First Order Probabilistic Temporal Logic

To include probability to our formalism, we extend the language L also at the syntactic and semantic level. Syntactically, we add a special unary operator symbol, $supp$, and a special binary operator symbol, $conf$, which satisfy the following rules:

T4 If p is a formula in L, then $supp(p)$ is a constant.

T5 If p, q are formulae in L, then $conf(p, q)$ is a constant.

Semantically, we first need to add probability to a first order time structure $M = (S, x, \mathbf{I})$. If $S = \{s_0, s_1, \ldots\}$ is a countable[2] set of states, consider $\sigma(S)$ the $\sigma-$algebra generated by S (see Appendix A). The probability measure P on $\sigma(S)$ is defined such that $P(s_i) = p_i > 0$, for all $i \in \mathbb{N}$. Consider now a random variable $\mathbb{X} : S \to \mathbb{R}$ such that the probability $P(\mathbb{X} = s_i) = p_i$ for all $i \in \mathbb{N}$ – this condition assures that the probability systems $(S, \sigma(S), P)$ and $(\mathbb{R}, \mathcal{B}, P_{\mathbb{X}})$ model the same experiment. Such a random variable may be obtained if $\mathbb{X}$ has the (canonical) form $\mathbb{X} = \sum_{i \in \mathbb{N}} x_i \mathbf{1}_{s_i}$, where $\mathbf{1}_{s_i}$ is the indicator function of the basic

[2]There is a one-to-one relation between the elements of S and the set of natural numbers

event $\{s_i\}$ and $x_i \neq x_j$ for $i \neq j$. If $S^{\mathbb{N}} = \{\omega \,|\, \omega = (\omega_1, \omega_2, \ldots, \omega_t, \ldots), \omega_t \in S, t \in \mathbb{N}\}$, then the variable $\mathbb{X}$ induces the stochastic sequence $\psi : S^{\mathbb{N}} \to \mathbb{R}^{\mathbb{N}}$, where $\psi(\omega) = \{\mathbb{X}_t(\omega), t \in \mathbb{N}\}$ and $\mathbb{X}_t(\omega) = \mathbb{X}(\omega_t)$ for all $t \in \mathbb{N}$. The fact that each $\omega \in S^{\mathbb{N}}$ may be uniquely identified with a function $x : \mathbb{N} \to S$ and that $\mathbb{X}$ is a bijection between S and $\mathbb{X}(S)$ allow us to uniquely identify the function x with a single realization of the stochastic sequence. In other words, the sequence $x = (s_{(1)}, s_{(2)}, \ldots, s_{(i)}, \ldots)$ from the structure M can be seen as one of the outcomes of an infinite sequence of experiments, each experiment being modelled by the probabilistic system $(S, \sigma(S), P)$. To each such sequence corresponds a single realization of the stochastic sequence, $\psi(x) = (\mathbb{X}(s_{(1)}), \mathbb{X}(s_{(2)}), \ldots, \mathbb{X}(s_{(i)}), \ldots)$.

Definition 5.1 *Given L and a domain D, a stochastic (first order) linear time structure is a quintuple $\mathbb{M} = (S, P, \mathbb{X}, \psi, \boldsymbol{I})$, where*

- *$S = \{s_1, s_2, \ldots\}$ is a (countable) set of states,*
- *P is a probability measure on the $\sigma-$algebra $\sigma(S)$ such that $P(s_i) = p_i > 0, i \in \mathbb{N}$*
- *$\mathbb{X}$ is a random variable, $\mathbb{X} = \sum_{i \in \mathbb{N}} x_i \boldsymbol{1}_{s_i}$,*
- *ψ is a random sequence, $\psi(\omega) = \{\mathbb{X}(\omega_i)\}_1^\infty$ where $\omega \in S^{\mathbb{N}}$,*
- *$\boldsymbol{I}$ is a function that associates with each state s an interpretation $\boldsymbol{I}_s$ for all symbols from L.*

To each realization of the stochastic sequence ψ, obtained by random drawing of a point in $\mathbb{R}^\infty$ (or equivalently, of a point ω in $S^{\mathbb{N}}$), corresponds a realization of the stochastic structure $\mathbb{M}$. This realization (in the following called "world") is given by the (ordinary) linear time structure $M_\omega = (S, \omega, \mathbf{I})$, which implies that the semantics attached to the symbols of L, described in subsection 3.2.2, is totally effective. The only interpretation which must be still defined is the one for the operators $supp$ and $conf$. Therefore, given M_ω and p, q two formulae, we define:

$$supp(p) = \lim_{n \to \infty} \frac{\#\{i \in \{0, \ldots, n\} | (M_\omega, i) \models p\}}{n} \tag{5.22}$$

and

$$conf(p,q) = \begin{cases} \dfrac{supp(p \wedge q)}{supp(q)} & \text{if } supp(q) > 0, \\ 0 & \text{if } supp(q) = 0. \end{cases} \tag{5.23}$$

In agreement with these definitions, the domain D is extended to $D_e \cup D_f \cup [0,1]$. The existence of the limit in the expression of the *supp* operator is strictly connected with the behavior of the stochastic sequence, which is given by the joint distribution of the coordinates. Therefore, even if the definition of the two operators is based on a single world, M_ω, its correctness is implicitly related to the probability model of all worlds, $\mathbb{M}$. In the literature, the problem of the consequences of the joint distribution law on the semantics, in the framework of a probabilistic first-order logic, was not studied. A first reason is that a majority of the probabilistic logical frameworks have no temporal dimension (most of the studies in the literature) and, in this case, the probability system $(S, \sigma(S), P)$ is sufficient to give a semantics to an expression like $supp(p)$ (e.g. $supp(p) = P(\{s \in S | \mathbf{I}_s(p) = \text{true}\})$). A second reason is that, in the rare cases where a temporal dimension exists, an error is made by considering the single world (or path, or run) M as the whole space and forgetting the stochastic process which stays behind it.

5.2.1 Dependence and the Law of Large Numbers

Much of the largest part of stochastic process theory has to do with the joint distribution of sets of coordinates, under the general heading of dependence. The degree to which random variations of sequence coordinates are related to those of their neighbors in the time ordering, is sometime called the memory of a sequence; in the context of time-ordered observations, one may think in terms of amount of information contained in the current state of the sequence about its previous states. A sequence with no memory is a rather special kind of object, because the ordering ceases to have significance. It is like the outcome of a collection of independent random experiments conducted in parallel, and indexed in an arbitrary manner. Indeed, independence and stationarity are the best-known restrictions on

the behavior of a sequence. But while the emphasis in our framework will mainly be on finding ways to relax these conditions, they remain important because of the many classic theorems in probability and limit theory which are founded on them.

The amount of dependence in a sequence is the chief factor determining how informative a realization of given length can be about the distribution that generated it. At one extreme, the i.i.d. sequence is equivalent to a true random sample. The classical theorems of statistics can be applied to this type of distribution. At the other extreme, it is easy to specify sequences for which a single realization can never reveal the parameters of the distribution, even in the limit as its length tends to infinity. This last possibility is what concerns us most, since we want to know whether averaging operations applied to sequences have useful limiting properties.

Let $\{\mathbb{X}_t\}_1^\infty$ be a stochastic sequence and define $\overline{\mathbb{X}}_n = n^{-1} \sum_{t=1}^n \mathbb{X}_t$. Suppose that $E(\mathbb{X}_t) = \mu_t$ and $n^{-1} \sum_{t=1}^n \mu_t$ converges to μ, with $|\mu| < \infty$; this is trivial in the mean-stationary case in which $\mu_t = \mu$ for all t. In this simple setting, the sequence is said to obey the weak law of large numbers (WLLN) when $\overline{\mathbb{X}}_n$ converges in probability to μ, and the strong law of large numbers (SLLN) when $\overline{\mathbb{X}}_n$ converges almost sure to μ (see Appendix A). The difference between the two forms of the law is concentrated in the convergence model:

- Almost Sure Convergence. In this case, for almost every single realization ω of the stochastic sequence, the sequence $\{\overline{\mathbb{X}}_n(\omega)\}$ converges to μ. The exceptional realizations make up a set whose total probability is zero. This means that it is extremely unlikely, although perhaps not impossible, that one of these realizations will be selected on any random trial.

- Convergence in Probability. This condition does not guarantee convergence in the usual (pointwise) sense on any realization. It simply says that if the m^{th} value of the sequence $\overline{\mathbb{X}}_n(\omega)$ is observed, then for large m the probability is high that the value $\overline{\mathbb{X}}_m(\omega)$ is close to μ. It does not guarantee anything about the terms $\overline{\mathbb{X}}_i(\omega)$ for $i > m$.

To obey the law of large numbers, a sequence must satisfy regularity conditions relating to two distinct factors: the probability of extreme values (limited by bounding absolute moments) and the degree of dependence between coordinates. The necessity of a set of regularity conditions is usually hard to prove (except if the sequences are independent), but various configurations of dependency and boundedness conditions can be shown to be sufficient. These results usually exhibit a trade-off between the two dimensions of regularity; the stronger the moment restrictions are, the weaker dependence restrictions can be, and vice-versa.

Consider now the sequence of the indicator function for an event A (i.e. $\mathbb{X}_t = \mathbf{1}_A$ for all t). In this case, $\mu_t = \mu = P(A)$ and $\overline{\mathbb{X}}_n(\omega) = n^{-1} \sum_{t=1}^{n} \mathbf{1}_A(\omega) = n^{-1}\#\{i \in 1..n \,|\, \mathbb{X}_i(\omega) = 1\}$. If A is the event "the interpretation of the formula p is true", for a given formula p, then the expression for $\overline{\mathbb{X}}_n(\omega)$ may be identified (under some conditions) with the expression which gives, at the limit, the $supp(p)$. Consequently, $supp(p)$ exists (almost sure) if the stochastic sequence $\{\mathbb{X}_t\}_1^\infty$ satisfies the strong law of large numbers. Given a stochastic linear time structure $\mathbb{M} = (S, P, \mathbb{X}, \psi, \mathbf{I})$, the sequence $\{\mathbf{1}_A\}_1^\infty$ is derived from the random sequence ψ and so the necessary conditions for applying SLLN to $\{\mathbf{1}_A\}_1^\infty$ are inherited from the regularity conditions the "basic" stochastic process ψ must satisfy.

5.2.2 The Independence Case

The sequence $\{\mathbb{X}(\omega_i)\}_1^\infty = \{\mathbb{X}_i(\omega)\}_1^\infty$ is independent and identically distributed. Firstly, let p be a temporal free formula in L. On the probabilistic system $(S, \sigma(S), P)$ one define the event $A_p \in \sigma(S), A_p = \{s \in S \,|\, \mathbf{I}_s(p) = true\}$.

Lemma 5.1 *If $\{\mathbb{X}(\omega_i)\}_1^\infty$ are i.i.d. then $\{(\mathbf{1}_{A_p})_i\}_1^\infty = \{\mathbf{1}_{A_p}(\omega_i)\}_1^\infty$ is also i.i.d.*

The proof is elementary and is based on the fact that if the random variables $\mathbb{X}_i(\omega) = \mathbb{X}(\omega_i)$ and $\mathbb{X}_j(\omega) = \mathbb{X}(\omega_j)$ are independent then the random variables $\mathbf{1}_{A_p}(\omega_i)$ and $\mathbf{1}_{A_p}(\omega_j)$ are also independent (Pfeiffer [1989], pg. 223).

As we mentioned, the regularity conditions for SLLN concern the dependence restrictions and the moment restrictions. For the independence case, the Kolmogorov classical version of the SLLN may be applied for the sequence $\{\mathbf{1}_{A_p}(\omega_i)\}_1^\infty$:

Theorem 5.1 *(**Kolmogorov**) If $\{\mathbb{X}_t\}_1^\infty$ is an independent sequence of random variables, $E(\mathbb{X}_t) = \mu$ and $Var(\mathbb{X}_t) \leq \sigma^2$ for all $t \in \mathbb{N}$, then $\overline{\mathbb{X}}_n \to \mu$ a.s.*

Indeed, $Var(\mathbf{1}_{A_p}) = P(A)(1 - P(A)) < 1$. If M_ω is the world defined by $\omega \in S^{\mathbb{N}}$, then

$$\overline{\mathbf{1}_{A_p}}_n(\omega) = \frac{\sum_{t=1}^n \mathbf{1}_{A_p}(\omega_i)}{n} = \\ = \frac{\#\{i \leq n \,|\, \mathbf{1}_{A_p}(\omega_i) = 1\}}{n} = \frac{\#\{i \leq n \,|\, \mathbf{I}_{s_{(i)}}(p) = true\}}{n} = \\ = \frac{\#\{i \leq n \,|\, (M_\omega, i) \models p\}}{n} \quad (5.24)$$

Therefore, we may conclude that

Corollary 5.1 *If the random process ψ from the stochastic first-order linear time structure $\mathbb{M}$ is i.i.d., then for almost all worlds M_ω the interpretation of $supp(p)$, where p is a temporal free formula in L, exists and it is equal with $P(A_p)$.*

Consider now the temporal formula $X_k p$, $k > 0$. For a fixed world M_ω, we have $(M_\omega, i) \models X_k p$ if and only if $(M_\omega, i + k) \models p$. Therefore, the stochastic sequence corresponding to $X_k p$ is given by $\{(\mathbf{1}_{A_{X_k p}})_i\}_1^\infty = \{\mathbf{1}_{A_p}(\omega_{i+k})\}_1^\infty = \{(\mathbf{1}_{A_p})_i\}_{k+1}^\infty$, the last sequence being the one corresponding to the formula p, but without the first k coordinates. Because the approach for $k < 0$ is similar, we may conclude that:

Corollary 5.2 *If the random process ψ from the stochastic first-order linear time structure $\mathbb{M}$ is i.i.d., then for almost all worlds M_ω the interpretation of $supp(X_k p)$, where p is a temporal free formula in L and $k \in \mathbb{N}$, exists and it is equal to $P(A_p)$.*

The last type of formulae we consider is $X_{k_0} p_0 \wedge X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n$, where $p_i, i = 0 \ldots n$ are temporal free formulae and $0 = k_0 \leq k_1 \leq \cdots \leq k_n$. If Tp is an abbreviation for this

formula and M_ω is a fixed world, we have $(M_\omega, i) \models \text{Tp}$ if and only if $(M_\omega, i + k_j) \models p_j$ for all $j = 1..n$. To construct the stochastic sequence corresponding to Tp we first introduce the following transformation:

- If $\mathbb{X}_i(\omega) = \mathbb{X}(\omega_i)$ is the i^{th} coordinate of the stochastic sequence $\psi(\omega)$, then $g_p^k(x)$ denotes the Borel function (see Appendix A) such that

$$g_p^k(\mathbb{X}_i(\omega)) = (g_p^k \circ \mathbb{X}_i)(\omega) = \begin{cases} 1 & \text{if } \omega_{i+k} \in A_p, \\ 0 & \text{if not} \end{cases} \tag{5.25}$$

Therefore, the stochastic sequence for the formula p was obtained by applying on $\{\mathbb{X}_i\}$ the transformation g_p^0, whereas for the formula $X_k p$ one applied the transformation g_p^k. Given the formula Tp, consider the stochastic sequences $\{{}_0\mathbb{G}_i\}_1^\infty = \{g_{p_0}^{k_0}(\mathbb{X}_i)\}_1^\infty, \{{}_1\mathbb{G}_i\}_1^\infty = \{g_{p_1}^{k_1}(\mathbb{X}_i)\}_1^\infty, \ldots, \{{}_n\mathbb{G}_i\}_1^\infty = \{g_{p_n}^{k_n}(\mathbb{X}_i)\}_1^\infty$, corresponding to the formulae $X_{k_0}p_0, X_{k_1}p_1, \ldots, X_{k_n}p_n$. From these sequences we define the stochastic sequence $\{\mathbb{G}_i\}_1^\infty$, $\mathbb{G}_i(\omega) = \prod_{j=0}^n {}_j\mathbb{G}_i(\omega)$. According to the following lemma, $\{\mathbb{G}_i\}$ is the sequence corresponding to the formula Tp.

Lemma 5.2 *$\mathbb{G}_i(\omega) = 1$ if and only if $(M_\omega, i) \models Tp$.*

Proof ${}_j\mathbb{G}_i(\omega) = 1 \Leftrightarrow \omega_{i+k_j} \in A_{p_j} \Leftrightarrow (M_\omega, i + k_j) \models p_j \Leftrightarrow (M_\omega, i) \models X_{k_j}p_j$. Therefore, $\mathbb{G}_i(\omega) = 1 \Leftrightarrow {}_j\mathbb{G}_i(\omega) = 1, j = 0..n, \Leftrightarrow (M_\omega, i) \models X_{k_j}p_j, j = 0..n, \Leftrightarrow (M_\omega, i) \models \text{Tp}$.

Because $g_{p_j}^{k_j}(\mathbb{X}_i) = g_{p_j}^0(\mathbb{X}_{i+k_j})$, the random variable $\mathbb{G}_i$ can be expressed as $h(\mathbb{X}_i, \ldots, \mathbb{X}_{i+k_n})$, where h is a Borel function (a composition between the product function and the $g_{p_j}^{k_j}$ functions). The sequence $\{\mathbb{G}_i\}$ is identically distributed (condition inherited from the sequence $\{\mathbb{X}_i\}$ by applying the function h), but it is not independent (observation which is a consequence of the fact that $\mathbf{1}_{A_p}$ and $\mathbf{1}_{A_q}$ are independent if and only if the events A_p and A_q are independent). In exchange we may prove the following result, by applying Theorem A.2 and Theorem A.3 (see Appendix A):

Lemma 5.3 *For all $i \in \mathbb{N}$ and all $m \in \mathbb{N}, m \geq k_n + 1$, the random variables $\mathbb{G}_i$ and $\mathbb{G}_{i+m}$ are independent.*

Proof Consider the sequence of independent coordinates $\{\mathbb{X}_i, \ldots, \mathbb{X}_{i+k_n}, \ldots, \mathbb{X}_{i+m}, \ldots, \mathbb{X}_{i+m+k_n}\}$. According to Theorem A.3, the random vectors $W_1 = (\mathbb{X}_i, \mathbb{X}_{i+1}, \ldots, \mathbb{X}_{i+k_n})$ and $W_2 = (\mathbb{X}_{i+m}, \mathbb{X}_{i+m+1}, \ldots, \mathbb{X}_{i+m+k_n})$ are independent and so, by Theorem A.2, $\mathbb{G}_i = h(W_1)$ and $\mathbb{G}_{i+m} = h(W_2)$ are independent.

This Lemma affirms that the sequence $\{\mathbb{G}_i\}$ is what is called in stochastic process theory a k_n-dependent sequence, which is a particular case of a mixing sequence. A more detailed description of this notion is presented in Appendix A, Section A.4.1, but as a short summary, we say that a sequence is α-mixing (or strong mixing) if the supremum of the strong mixing coefficient, α_m, which is a measure of the dependence between coordinates separated by a distance m, converge to zero for $m \to \infty$. A consequence of Lemma 5.3 is that α_m is zero for $m \geq k_n + 1$, and evidently $\{\mathbb{G}_i\}$ is a strong mixing sequence. The importance of this result lies in the fact that this kind of dependence is sufficient, under certain conditions, for $\{\mathbb{G}_i\}$ to obey SLLN.

Theorem 5.2 *(Hall and Heyde [1980], pg. 40) Let $\{\mathbb{X}_t\}_1^\infty$ be a α-mixing sequence such that $E(\mathbb{X}_t) = \mu$ and $E(\mathbb{X}_t^2) < \infty, t \geq 1$. Suppose that*

$$\sum_{t=1}^{\infty} b_t^{-2} Var(\mathbb{X}_t) < \infty \text{ and } \sup_n b_n^{-1} \sum_{t=1}^{n} E(|\mathbb{X}_t|) < \infty, \tag{5.26}$$

where $\{b_t\}$ is a sequence of positive constants increasing to ∞. Then

$$b_n^{-1} \sum_{t=1}^{n} \mathbb{X}_t \xrightarrow{a.s.} \mu$$

For the particular case $b_n = n$, the conclusion of the theorem becomes $\overline{\mathbb{X}}_n \xrightarrow{a.s.} \mu$. We will prove that the sequence $\{\mathbb{G}_i\}$ verifies the hypothesis of Theorem 5.2. Indeed, $\mathbb{G}_i = \prod_{j=0}^{n} {}_j\mathbb{G}_i = \prod_{j=0}^{n} g_{p_j}^{k_j}(\mathbb{X}_i) = \prod_{j=0}^{n} g_{p_j}^{0}(\mathbb{X}_{i+k_j})$. According to Theorem A.2, because $\{\mathbb{X}_i, \ldots, \mathbb{X}_{i+k_n}\}$ is an independent class of random variables and $g_{p_j}^0$ are Borel functions, the class of random variables $\{{}_j\mathbb{G}_i\}_{j=0}^n$ is also independent. From the properties of the expectation, $E(\mathbb{G}_i(\omega)) = \prod_{j=0}^{n} E({}_j\mathbb{G}_i(\omega)) = \prod_{j=0}^{n} E(\mathbf{1}_{A_{p_j}}(\omega_{i+k_j})) = \prod_{j=0}^{n} P(A_{p_j})$. The coordinate $\mathbb{G}_i$ being a product of indicator functions, we have $\mathbb{G}_i^2 = \mathbb{G}_i$, so that the condition $E(\mathbb{G}_i^2) = \prod_{j=0}^{n} P(A_{p_j}) < \infty$ is

also verified. For the variance we have $Var(\mathbb{G}_i) = E(\mathbb{G}_i^2) - E(\mathbb{G}_i)^2 = E(\mathbb{G}_i)(1 - E(\mathbb{G}_i)) < 1$, so $\sum_{i=1}^{\infty} i^{-2} Var(\mathbb{G}_i) < \sum_{i=1}^{\infty} i^{-2} < \infty$. And for the last condition, $n^{-1} \sum_{i=1}^{n} E(|\mathbb{G}_i|) < n^{-1} \sum_{i=1}^{n} 1 = 1 < \infty$. Therefore, Theorem 5.2 is verified and so $\overline{\mathbb{G}_n} \xrightarrow{a.s.} E(\mathbb{G}_i)$. In conclusion

Corollary 5.3 *If the random process ψ from the stochastic first-order linear time structure $\mathbb{M}$ is i.i.d., then for almost all worlds M_ω the interpretation of supp(Tp), where Tp is a temporal formula $X_{k_0} p_0 \wedge X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n$, with $p_i, i = 0 \ldots n$ temporal free formulae, exists and it is equal to $\prod_{j=0}^{n} P(A_{p_j})$.*

Finally, based on Corollaries 5.1-5.3, we can prove the following fundamental theorem:

Theorem 5.3 ***(Independence and Consistency)*** *If the random process ψ from the stochastic first-order linear time structure $\mathbb{M} = (S, P, \mathbb{X}, \psi, \boldsymbol{I})$ is i.i.d., then almost all worlds $M_\omega = (S, \omega, \boldsymbol{I}_s)$ are consistent linear time structures.*

But the property of independence for the random process ψ, even if it assures the property of consistency for linear time structures M_ω, creates another problem for the temporal data mining methodology. Indeed, what we try to discover are temporal rules expressing a dependence between the event occurred at time t and the events occurred before time t. It is easy to show that the independence implies the correlation between the body and the head of the temporal rule to be zero. The question is how much do we have to relax the independence condition to still conserve the property of consistency.

5.2.3 The Mixing Case

Since mixing is not so much a property of the sequence $\{\mathbb{X}_i\}$ as of the sequence of σ-fields generated by $\{\mathbb{X}_i\}$, it holds for any random variables measurable on those σ-fields. More generally, we have the following important implication:

Theorem 5.4 *(Davidson [1994], pg. 210) Let $\mathbb{Y}_i = g(\mathbb{X}_i, \mathbb{X}_{i-1}, \ldots, \mathbb{X}_{i-k})$ be a Borel function, for finite k. If $\mathbb{X}_i$ is α-mixing (respectively ϕ-mixing) of size $-\varphi$, then $\mathbb{Y}_i$ is also.*

This theorem is the key to prove that ψ α-mixing is a sufficient condition for consistency. Indeed, the previously defined functions $g_{p_j}^{k_j}$ and $h = \prod_{j=0}^{n} g_{p_j}^{k_j}$ are Borel transformations. Consequently, the sequence $\{g_p^0(\mathbb{X}_t)\}$ (corresponding to temporal free formula p), the sequence $\{g_p^k(\mathbb{X}_t)\}$ (corresponding to temporal formula $X_k p$) and the sequence $\{h(\mathbb{X}_t)\}$ (corresponding to temporal formula $X_{k_0} p_0 \wedge X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n$) are also α-mixing, of the same size as $\{\mathbb{X}_i\}$. The following step is to verify if the hypotheses of Theorem 5.2, for $b_n = n$, are satisfied for these three sequences. It is easy to show that a sufficient condition for (5.26) to hold, in the case of an identical distributed sequence with positive coordinates, is $\mathbb{X} \leq B$, with B a positive constant (e.g. $\mathbb{X} \leq B \Rightarrow \mathbb{X}^2 \leq B^2 \Rightarrow E(\mathbb{X}^2) \leq B^2 \Rightarrow Var(\mathbb{X}) \leq B^2 \Rightarrow \sum_{n=1}^{\infty} n^{-2} Var(\mathbb{X}) \leq B^2 \sum_{n=1}^{\infty} n^{-2} < \infty$). For the first two sequences, $\{g_p^0(\mathbb{X}_t)\}$ and $\{g_p^k(\mathbb{X}_t)\}$, as the coordinates are indicator function $\mathbf{1}_{A_p}$, the sufficient condition is fulfilled. For the last sequence, because the coordinates are product of indicator functions $\mathbf{1}_{A_{p_j}}$, the sufficient condition is also fulfilled. Therefore the conclusion of Theorem 5.2 holds, implying that for any formula p in L the $supp(p)$ exists (but, unlike in the independent case, we can not give an exact expression for the support of a temporal formula like Tp). This result is formalized in the following theorem.

Theorem 5.5 ***(Mixing and Consistency)*** *If the random process ψ from the stochastic first-order linear time structure $\mathbb{M} = (S, P, \mathbb{X}, \psi, \boldsymbol{I})$ is α-mixing, then almost all worlds $M_\omega = (S, \omega, \boldsymbol{I}_s)$ are consistent linear time structures.*

Let $\mathbb{H}$ be the temporal rule $H_1 \wedge H_2 \wedge \ldots \wedge H_n \mapsto H_{n+1}$. It is evident that the rule $X_k\mathbb{H}$ has the same support as the rule $\mathbb{H}$, for any $k \in \mathbb{Z}$, so consider the following canonical form for $\mathbb{H}$, $X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n \wedge X_{k_{n+1}} p_{n+1}$, where $0 = k_1 \leq k_2 \leq \ldots \leq k_n < k_{n+1}$ and $H_{n+1} = p_{n+1}$. If ψ is i.i.d., a consequence of Corollary 5.3 is that the confidence of the rule $\mathbb{H}$ is $P(A_{p_{n+1}})$.

Indeed,

$$conf(\mathbb{H}) = conf(X_{k_{n+1}} p_{n+1}, X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n) =$$
$$= \frac{supp(X_{k_1} p_1 \wedge \ldots \wedge X_{k_{n+1}} p_{n+1})}{supp(X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n)} = \frac{\prod_{j=1}^{n+1} P(A_{p_j})}{\prod_{j=1}^{n} P(A_{p_j})} = P(A_{p_{n+1}})$$

If ψ is α-mixing, we can obtain only an upper bound for the confidence of the temporal rule. To facilitate the notation, let $\mathbb{A}$ denote the event $\{g_{p_{n+1}}^{k_{n+1}}(\mathbb{X}) = 1\}$ and $\mathbb{B}$ the event $\{g_{p_0}^{k_0}(\mathbb{X}) = 1, \ldots, g_{p_n}^{k_n}(\mathbb{X}) = 1\}$.

Lemma 5.4 *If ψ is α-mixing, the confidence of the temporal rule (template) $\mathbb{H}$ having the form $X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n \wedge X_{k_{n+1}} p_{n+1}$, with $0 = k_1 \leq k_2 \leq \ldots \leq k_n < k_{n+1}$, satisfies the relation*

$$conf(\mathbb{H}) \leq \frac{\alpha_1}{P(\mathbb{B})} + P(\mathbb{A}).$$

Proof According to Definition 3.11 and to relation (5.23) we have

$$conf(\mathbb{H}) = conf(X_{k_{n+1}} p_{n+1}, X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n) =$$
$$= \frac{supp(X_{k_1} p_1 \wedge \ldots \wedge X_{k_{n+1}} p_{n+1})}{supp(X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n)} = \frac{E\left(\prod_{j=1}^{n+1} g_{p_j}^{k_j}(\mathbb{X})\right)}{E\left(\prod_{j=1}^{n} g_{p_j}^{k_j}(\mathbb{X})\right)} = \frac{P(\mathbb{B} \cap \mathbb{A})}{P(\mathbb{B})} = P(\mathbb{A}/\mathbb{B})$$

For any $i \in \mathbb{N}$, the event $\mathbb{B} \in \sigma(\mathbb{X}_i, \mathbb{X}_{i+1}, \ldots, \mathbb{X}_{i+k_n}) \subseteq \mathcal{X}_1^{i+k_n}$ whereas the event $\mathbb{A} \in \sigma(\mathbb{X}_{i+k_{n+1}}) \subseteq \mathcal{X}_{i+k_{n+1}}^{\infty}$ (see Section A.4.1). The process ψ being α-mixing, we have

$$|P(\mathbb{B} \cap \mathbb{A}) - P(\mathbb{B})P(\mathbb{A})| \leq \sup_i \alpha(\mathcal{X}_1^{i+k_n}, \mathcal{X}_{i+k_{n+1}}^{\infty}) = \alpha_{k_{n+1}-k_n}.$$

Therefore,

$$P(\mathbb{A}/\mathbb{B}) = \frac{P(\mathbb{B} \cap \mathbb{A})}{P(\mathbb{B})} - \frac{P(\mathbb{A})P(\mathbb{B})}{P(\mathbb{B})} + P(\mathbb{A}) \leq$$
$$\leq \frac{|P(\mathbb{B} \cap \mathbb{A}) - P(\mathbb{B})P(\mathbb{A})|}{P(\mathbb{B})} + P(\mathbb{A}) \leq \frac{\alpha_{k_{n+1}-k_n}}{P(\mathbb{B})} + P(\mathbb{A}) \leq \frac{\alpha_1}{P(\mathbb{B})} + P(\mathbb{A}),$$

where the last inequality comes from the monotonicity of the sequence α_n.

5.2.4 The Near Epoch Dependence Case

The mixing concept has a serious drawback from the viewpoint of applications in stochastic limit theory, in that a function of a mixing sequence (even an independent sequence) that depends on an infinite number of coordinates of the sequence is not generally mixing. Let $\mathbb{X}_i = g(\ldots, \mathbb{V}_{i-1}, \mathbb{V}_i, \mathbb{V}_{i+1}, \ldots)$, where $\mathbb{V}_i$ is a vector of mixing processes. The idea is that although $\mathbb{X}_i$ may not be mixing, if it depends almost entirely on the "near epoch" of $\{\mathbb{V}_i\}$ it will often have properties permitting the application of limit theorems, including SLLN. Near-epoch dependence (see Definition A.4.2, Appendix A) is not an alternative to a mixing assumption; it is a property of the mapping from $\{\mathbb{V}_i\}$ to $\{\mathbb{X}_i\}$, not of the random variables themselves.

The main key we applied in the previous cases is the property of a Borel transformation g to inherit the type of dependence (the independence or the mixing dependence) from the initial sequence. For the near-epoch dependence this property is achieved only if the function g satisfies additional conditions and only for particular L_q-NED sequences. Concretely, let $g(x) : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$ a Borel function and consider the metric on $\mathbb{R}^n$, $\rho(\boldsymbol{x}^1, \boldsymbol{x}^2) = \sum_{i=1}^n |x_i^1 - x_i^2|$ for measuring the distance between points $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$. If g satisfies

(i) g continuous

(ii) $|g(\boldsymbol{X}^1) - g(\boldsymbol{X}^2)| \leq M\rho(\boldsymbol{X}^1, \boldsymbol{X}^2)$ a.s., where $\boldsymbol{X}^1$, $\boldsymbol{X}^2$ are random vectors from $\mathbb{R}^n$

then the following theorem holds:

Theorem 5.6 *(Davidson [1994], pg. 269) Let $\mathbb{X}_{ji}$ be L_2-NED of size $-a$ on$\{\mathbb{V}_i\}$ for $j = 1..n$, with constants d_{ji}. If g satisfies the conditions (i)-(ii), then $\{g(\mathbb{X}_{1i}, \ldots, \mathbb{X}_{ni})\}$ is also L_2-NED on $\{\mathbb{V}_i\}$ of size $-a$, with constants a finite multiple of* $\max_i\{d_{ji}\}$.

Suppose the process $\psi = \{\mathbb{X}_i\}$ is L_2-NED of size $-a$ on $\{\mathbb{V}_i\}$. As we have already seen in the previous cases, for p a temporal free formula, the corresponding sequence is $\{g_p^0(\mathbb{X}_i)\}$. The function $g_p^0(\cdot)$, as defined in (5.25), don't satisfy the condition (i). But it is possible to define

a function $\tilde{g}_p$ which takes the value one for the arguments $x \in \mathbb{X}(A_p) = \{\mathbb{X}(s) : s \in A_p\}$, the value zero for the arguments $x \in \{\mathbb{X}(s) : s \in S - A_p\}$ and to be continuous for $x \in \mathbb{R}$. Because $g_p^0(\mathbb{X}_i(\omega)) = \tilde{g}_p(\mathbb{X}_i(\omega)) \in \{0, 1\}$, it is possible (the support of $\mathbb{X}$ being a discrete set) to choose the constant M_p such that $|\tilde{g}_p(x) - \tilde{g}_p(y)| \leq M_p|x - y|$, for any $x, y \in \mathbb{X}(S)$. Therefore, the conditions of Theorem 5.6 are verified and so $\{\tilde{g}_p(\mathbb{X}_i)\} = \{\mathbf{1}_{A_p}\}$ is also L_2-NED of size $-a$ on $\{\mathbb{V}_i\}$.

For the temporal formula $X_k p$, the corresponding sequence is $\{g_p^k(\mathbb{X}_i)\} = \{g_p^0(\mathbb{X}_{i+k})\}$. According to Theorem A.6, $\mathbb{X}_{i+k}$ is also L_2-NED, so using the same argument as in the previous paragraph, $\{\tilde{g}_p(\mathbb{X}_{i+k})\}$ is L_2-NED. Finally, consider the temporal formula Tp, expressed as $X_{k_0} p_0 \wedge X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n$, where $p_i, i = 0 \ldots n$ are temporal free formulae and $0 = k_0 \leq k_1 \leq \cdots \leq k_n$. The corresponding sequence is

$$\left\{\prod_{j=0}^{n} g_{p_j}^{k_j}(\mathbb{X}_i)\right\} = \left\{\prod_{j=0}^{n} g_{p_j}^{0}(\mathbb{X}_{i+k_j})\right\} = \left\{\prod_{j=0}^{n} \tilde{g}_{p_j}(\mathbb{X}_{i+k_j})\right\} = \{h(\mathbb{X}_i, \ldots, \mathbb{X}_{i+k_n})\} = \{h(\mathbb{X}'_{i0}, \ldots, \mathbb{X}'_{in})\},$$

where $\mathbb{X}'_{ij} = \mathbb{X}_{i+k_j}$. Theorem A.6 assures that $(\mathbb{X}'_{i0}, \ldots, \mathbb{X}'_{in})$ are all L_2-NED. Concerning the transformation h, it satisfies (i) as being a product of continuous functions and satisfies (ii) because, denoting $\boldsymbol{X}_i = (\mathbb{X}_i, \ldots, \mathbb{X}_{i+k_n})$,

$$|h(\boldsymbol{X}_i^1) - h(\boldsymbol{X}_i^2)| = \left|\prod_{i=0}^{n} \tilde{g}_{p_j}(\mathbb{X}_{i+k_j}^1) - \prod_{i=0}^{n} \tilde{g}_{p_j}(\mathbb{X}_{i+k_j}^2)\right| \leq$$

$$\leq \sum_{j=0}^{n} \left|\tilde{g}_{p_j}(\mathbb{X}_{i+k_j}^1) - \tilde{g}_{p_j}(\mathbb{X}_{i+k_j}^2)\right| \leq \sum_{j=0}^{n} M_{p_j} \rho(\mathbb{X}_{i+k_j}^1, \mathbb{X}_{i+k_j}^2) \leq M\rho(\boldsymbol{X}_i^1, \boldsymbol{X}_i^2).$$

The first inequality comes from the fact that $|\prod_i x_i - \prod_i y_i| \leq \sum_i |x_i - y_i|$ if $x_i, y_i \in \{0, 1\}$ and the second inequality is the condition (ii) for the transformations $\tilde{g}_{p_j}$. Therefore, Theorem 5.6 holds and so the sequence corresponding to the temporal formula Tp is L_2-NED. In conclusion,

Corollary 5.4 *If ψ is L_2-NED then for any formula in L the corresponding sequence is also L_2-NED.*

The following step is to establish the sufficient condition for the application of SLLN to a L_q-NED sequence. The concept of near-epoch dependence, as a mapping from $\{\mathbb{V}_i\}$ to

$\{\mathbb{X}_i\}$, acquires importance when $\{\mathbb{V}_i\}$ is a mixing process, because then $\{\mathbb{X}_i\}$ inherits certain useful characteristics permitting the application of limit theorems. In Davidson and de Jong [1997] are summarized the up-to-date strong laws for dependent heterogeneous processes, including NED sequences. For the above mentioned dependence, there are different version for SLLN, due to the multiple parameters involved: the mixing size $-a$, the NED size $-b$, the NED order q, the maximum order of existing moments $q_{\max}$ and, in addition, the rates of increase of the sequences $\| \mathbb{X}_i - \mu_i \|_q$ and a_n. We consider the following form for the limit theorem, which includes the case $q = 2$.

Theorem 5.7 *(Davidson and de Jong [1997], pg. 7) Let a sequence $\{\mathbb{X}_i\}$ with means $\{\mu_i\}$ be L_q-NED, $1 \leq q \leq 2$, of size $-b$, with respect to constants $d_i \ll \| \mathbb{X}_i - \mu_i \|_q$, on a sequence $\{\mathbb{V}_i\}$ which is α-mixing of size $-a$. If $a_n / \sqrt{n} \uparrow \infty$ as $n \to \infty$, and*

$$\frac{\| \mathbb{X}_n - \mu_i \|_q^{2-q/2}}{a_n} = O(n^{\epsilon}), \tag{5.27}$$

where

$$\epsilon < 1/2 - 1/q + \min\{-1/2, \min\{bq/2, a/2\} - 1\} \tag{5.28}$$

then $a_n^{-1} \sum_{i=1}^{n} (\mathbb{X}_i - \mu_i) \to 0$, a.s.

For $q = 2$ and $a_n = n$, the condition (5.27) becomes $\| \mathbb{X}_n - \mu_n \|_2 = O(n^{\epsilon+1})$ or $\sqrt{Var(\mathbb{X}_n)} = O(n^{\epsilon+1})$. As for any formula p in L, the corresponding L_2-NED sequence has bounded coordinates ($\mathbb{X}_i \in \{0, 1\}$), $\sqrt{Var(\mathbb{X}_n)} = O(n^0)$. In the same time, the condition (5.28) becomes $\epsilon < \min\{-1/2, \min\{b, a/2\} - 1\}$ or $\epsilon \leq -1$ (limit attained when $a, b \downarrow 0$). Therefore the condition (5.27) is satisfied and so $\{\mathbb{X}_i\}$ obeys the SLLN. (*Remark*: If in Theorem 5.7 we set $a = \infty$ then $\{\mathbb{X}_i\}$ becomes a pure α-mixing process, whereas for $b = \infty$, $\{\mathbb{X}_i\}$ is a L_2-NED function of an independent process.)

Therefore, as in the previous cases, we may conclude that

Theorem 5.8 ***(Near-Epoch Dependence and Consistency)*** *If the random process ψ from the stochastic first-order linear time structure $\mathbb{M} = (S, P, \mathbb{X}, \psi, \boldsymbol{I})$ is L_2-NED on an α-mixing*

sequence, then almost all worlds $M_\omega = (S, \omega, \boldsymbol{I}_s)$ are consistent linear time structures.

5.3 Consistency of Granular Time Structure

While in Chapter 4 we could prove that the consistency of the granular time structure M_μ is inherited from the time structure M if the temporal type μ satisfies certain conditions (see Theorem 4.4), we will show that this fundamental property is derived, under a probabilistic framework, from the regularity conditions of the "basic" stochastic process ψ.

If $\psi = \{\mathbb{X}_i\}$, $i = 1 \ldots \infty$, is a sequence and μ a temporal type, then one denotes $\mathbf{X}_{\mu(i)}$ the random vector $(\mathbb{X}_{j_1}, \ldots, \mathbb{X}_{j_k})$, where $j_1 < \cdots < j_k$ are all the indices from $\mu(i)$. The random sequence induced by μ on ψ is simply $\mu[\psi] = \{\mathbf{X}_{\mu(i)}\}_{i=1}^{\infty}$. Similarly, if $\omega \in S^{\mathbb{N}}$ then $\omega_{\mu(i)} = (\omega_{j_1}, \ldots, \omega_{j_k})$ and $\mu[\omega] = \{\omega_{\mu(i)}\}_1^{\infty}$. According to Def. 4.4 and Def. 5.1, a stochastic granular time structure is defined as:

Definition 5.2 *If $\mathbb{M} = (S, P, \mathbb{X}, \psi, \boldsymbol{I})$ is a stochastic (first-order) linear time structure and μ is a temporal type from $\mathcal{G}_0$, then the stochastic granular time structure induced by μ on $\mathbb{M}$ is the quintuple $\mathbb{M}_\mu = (2^S, P, \mathbb{X}, \mu[\psi], \boldsymbol{I}^\mu)$, where $\boldsymbol{I}^\mu$ is given by (4.5) and (4.6)*

Practically, the random process $\mu[\psi]$ from the stochastic granular time structure M_μ is a sequence of random vectors obtained by grouping the coordinates of the process ψ according to the mapping μ. To each realization of the stochastic sequence ψ, obtained by a random drawing of a point ω in $S^{\mathbb{N}}$, corresponds a realization of the stochastic structure $\mathbb{M}$ (i.e. the time structure $M_\omega = (S, \omega, \mathbf{I})$) and a corresponding realization of the stochastic structure $\mathbb{M}_\mu$ (i.e. the granular time structure $M_{\mu[\omega]} = (2^S, \mu[\omega], \mathbf{I}^\mu)$). In the following one establishes the expression linking the interpretation $\mathbf{I}^\mu$ of a given formula in L with the random process $\mu[\psi]$. For this we introduce the function $\mathcal{S}$ defined by $\mathcal{S}(\mathbf{X}_{\mu(i)}) = (\#\mu(i))^{-1} \sum_{j \in \mu(i)} \mathbb{X}_j$. If $\{\mathbb{X}_i\}$ are identical distributed, with $E(\mathbb{X}_i) = \gamma$, then it is evident that $E(\mathcal{S}(\mathbf{X}_{\mu(i)}) = \gamma$, for all $i \in \mathbb{N}$. Consider the following two situations:

- *Temporal free formula*: According to 4.5 and to 5.24,

$$\mathbf{I}^{\mu}_{\mu[\omega](i)}(p) = \text{supp}(p, \tilde{M}_{\mu[\omega](i)}) = \mathcal{S}\left((\mathbf{1}_{A_p})_{\mu[\omega](i)}\right). \tag{5.29}$$

  While the sequence corresponding to the interpretation of the temporal free formula p is $\{(\mathbf{1}_{A_p})_i\}$ (under the time structure M_ω), the same sequence, but under the granular time structure $M_{\mu[\omega]}$, is represented by the arithmetic mean of the vectors from $\mu[(\mathbf{1}_{A_p})_i]$, i.e. $\left\{\mathcal{S}\left((\mathbf{1}_{A_p})_{\mu[\omega](i)}\right)\right\}_{i=1}^{\infty}$.

- *Temporal formula*: According to 4.6 and 5.29, for a temporal formula $X_{k_1}p_1 \wedge \ldots \wedge X_{k_n}p_n$ we have

$$\mathbf{I}^{\mu}_{\mu[\omega](i)}(X_{k_1}p_1 \wedge \ldots \wedge X_{k_n}p_n) = \frac{1}{n}\sum_{j=1}^{n}\mathbf{I}^{\mu}_{\mu[\omega](i+k_j)}(p_j) = \frac{1}{n}\sum_{j=1}^{n}\mathcal{S}\left((\mathbf{1}_{A_{p_j}})_{\mu[\omega](i+k_j)}\right), \tag{5.30}$$

  which represents the arithmetic mean of the vectors with the indices $i + k_j$, $j = 1..n$, from the sequences $\mu[(\mathbf{1}_{A_{p_1}})_i], \ldots, \mu[(\mathbf{1}_{A_{p_n}})_i]$.

The utility of the expressions 5.29 and 5.30 is due to the fact that, according to 4.12, if the corresponding sequence for a given formula in L obeys to SLNN, then the support of this formula exists. By analogy with the study of the degree of dependence allowed for the random process ψ, the following cases are analyzed.

5.3.1 The Independence Case

If ψ is a i.i.d. process, then according to Lemma 5.1, for p a temporal free formula, the sequence $\{\mathbf{1}_{A_p}\}_1^{\infty}$ is also i.i.d. By applying Theorem A.3, the vectors $(\mathbf{1}_{A_p})_{\mu(i)}$ are independent, and consequently, according to Theorem A.2 and to the fact that the function $\mathcal{S}$ is a Borel transformation, the sequence $\left\{\mathcal{S}\left((\mathbf{1}_{A_p})_{\mu[\omega](i)}\right)\right\}_{i=1}^{\infty}$ is independent. Therefore, the classical Kolmogorov theorem may be applied and so the support of the formula p, under the granular time structure $M_{\mu[\omega]}$, exists almost sure. For the temporal formula $X_{k_1}p_1 \wedge \ldots \wedge X_{k_n}p_n$, similar considerations assure that, for a fixed i, the random variables $\mathcal{S}\left((\mathbf{1}_{A_{p_1}})_{\mu[\omega](i+k_1)}\right), \ldots, \mathcal{S}\left((\mathbf{1}_{A_{p_n}})_{\mu[\omega](i+k_n)}\right)$ are independent. The sequence corresponding to

the temporal formula (see 5.30) is not independent, but k_n-dependent, and so the conditions of Theorem 5.2 are satisfied. By consequence this sequence obeys the law of large numbers, i.e. the support of the temporal formula exists. We can even obtain the exact expression of the support, which is

$$\text{supp}(X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n, M_{\mu[\omega]}) = \frac{1}{n} \sum_{j=1}^{n} P(A_{p_j}).$$

In conclusion, we have the following theorem:

Theorem 5.9 *If the random process ψ from the stochastic first-order linear time structure $\mathbb{M} = (S, P, \mathbb{X}, \psi, \boldsymbol{I})$ is i.i.d., then almost all granular time structures induced by a temporal type $\mu \in \mathcal{G}_0$, $M_{\mu[\omega]} = (2^S, \mu[\omega], \boldsymbol{I}^{\mu})$, are consistent.*

Remark: This result is stronger than those obtained in Theorem 4.4, where the temporal type has to satisfy a more restricted condition, i.e. $\mu \in \mathcal{G}_2$. This is explained by the fact that in a probabilistic framework we can apply fundamental results which go beyond a simple algebraic manipulation. If we follow this idea, we can prove that the conclusion of the previous theorem remains true even if we replace the function giving the interpretation of a temporal formula (in this case, the arithmetic mean) with any Borel transformation.

5.3.2 The Mixing Case

If ψ is α-mixing then it is evident that any subsequence of ψ is also α-mixing. The following result, necessary for our rationing, is a consequence of the fact that mixing is a property of σ-fields generated by $\{\mathbb{X}_i\}$.

Lemma 5.5 *Consider $\mathbb{X}_i$ an α-mixing sequence of size $-\varphi$ and let be k sequences ${}_j\mathbb{Y}_i$ obtained by applying on $\{\mathbb{X}_i\}$ the measurable functions $g_j(\mathbb{X}_t, \ldots, \mathbb{X}_{t-\tau_j})$, $j = 1..k$. Then the sequence ${}_1\mathbb{Y}_{i_1}, {}_2\mathbb{Y}_{i_2}, \ldots, {}_k\mathbb{Y}_{i_k}, {}_1\mathbb{Y}_{i_{k+1}}, \ldots$, obtained by tacking successively from each sequence ${}_j\mathbb{Y}_i$ coordinates with indices in an increasing order, is also α-mixing of size $-\varphi$.*

The utility of this lemma is due to the fact that the granules of a temporal type from $\mathcal{G}_0$ have a variable size, and so we can not apply a single measurable function $g(\cdot)$, with a fixed

number of parameters, on $\{\mathbf{1}_{A_p}\}$. By considering for each effective size $k \in \mathbb{N}$ the function $g_k(x_1, \ldots, x_k) = k^{-1} \sum x_i$ and applying Lemma 5.5 on $\{\mathbf{1}_{A_p}\}$ one obtains that $\mathcal{S}\left((\mathbf{1}_{A_p})_{\mu[\omega](i)}\right)$, p a temporal free formula, is $\alpha-$mixing. Concerning a temporal formula $X_{k_1} p_1 \wedge \ldots \wedge X_{k_n} p_n$, by applying n times Lemma 5.5 for the $\alpha-$mixing sequences $\{\mathbf{1}_{A_{p_j}}\}$, $j = 1..n$, we obtain the $\alpha-$mixing sequences $\mathcal{S}\left((\mathbf{1}_{A_{p_j}})_{\mu[\omega](i)}\right)$, $j = 1..n$. From these sequences one extracts the subsequence $\mathcal{S}\left((\mathbf{1}_{A_{p_1}})_{\mu[\omega](i+k_1)}\right), \ldots, \mathcal{S}\left((\mathbf{1}_{A_{p_n}})_{\mu[\omega](i+k_n)}\right)$, $i \in \mathbb{N}$ (which is $\alpha-$mixing, according to the same Lemma), on which we apply the function $g_n(\cdot)$. The resulted sequence is again $\alpha-$mixing, according to Theorem 5.4. Finally, the corresponding sequence for any formula in L is $\alpha-$mixing, bounded by the interval $[0, 1]$, thus fulfilling the conditions of Theorem 5.2. In consequence, we can affirm that

Theorem 5.10 *If the random process ψ from the stochastic first-order linear time structure $\mathbb{M} = (S, P, \mathbb{X}, \psi, \boldsymbol{I})$ is $\alpha-$mixing, then almost all granular time structures induced by a temporal type $\mu \in \mathcal{G}_0$, $M_{\mu[\omega]} = (2^S, \mu[\omega], \boldsymbol{I}^\mu)$, are consistent.*

5.3.3 The Near Epoch Dependence Case

The results in this section are obtained only for ψ being L_2-NED on $\{\mathbb{V}_i\}$ an α-mixing sequence and μ a temporal type from $\mathcal{G}_2$. According to Corollary 5.4, any sequence $\{\mathbf{1}_{A_p}\}$ is also L_2-NED on the same sequence $\{\mathbb{V}_i\}$. If $\#\mu(i) = k$ then it is easy to show that the function $g_k(\cdot)$ is continuous and satisfies the uniform Lipschitz condition. Therefore, according to Theorem 5.6, the sequence corresponding to the temporal free formula p, $\mathcal{S}\left((\mathbf{1}_{A_p})_{\mu[\omega](i)}\right)$, is also L_2-NED on $\{\mathbb{V}_i\}$. The same theorem, applied to the sequence of vectors $\left(\mathcal{S}\left((\mathbf{1}_{A_{p_1}})_{\mu[\omega](i+k_1),}\right) \ldots, \mathcal{S}\left((\mathbf{1}_{A_{p_n}})_{\mu[\omega](i+k_n)}\right)\right)$, all L_2-NED on $\{\mathbb{V}_i\}$, and for the Lipschitz function $g_n(\cdot)$, assures that the sequence $\frac{1}{n} \sum_{j=1}^{n} \mathcal{S}\left((\mathbf{1}_{A_{p_j}})_{\mu[\omega](i+k_j)}\right)$ is L_2-NED on $\{\mathbb{V}_i\}$. Therefore, for any formula in L the corresponding sequence is L_2-NED on the α-mixing sequence $\{\mathbb{V}_i\}$. Furthermore, these sequences fulfil the conditions of Theorem 5.7 for $q = 2$ and so obey the strong law of large numbers. In consequence, we can affirm that

Theorem 5.11 *If the random process ψ from the stochastic first-order linear time structure*

$\mathbb{M} = (S, P, \mathbb{X}, \psi, \boldsymbol{I})$ *is* L_2*-NED on an* $\alpha-$*mixing sequence, then almost all granular time structures induced by a temporal type* $\mu \in \mathcal{G}_2$, $M_{\mu[\omega]} = (2^S, \mu[\omega], \boldsymbol{I}^{\mu})$, *are consistent.*

Remark: For the near-epoch dependence case we were forced to impose a stronger restriction to the temporal type μ (constant size and total coverage) to compensate the higher degree of dependence of the stochastic process ψ.

5.4 Summary

To the natural question "Is there a theoretical framework in which the consistency property for a time structure $M = (S, x, \mathbf{I})$ is the objective consequence of a deeper property?", we tried to give an answer by extending our formalism with a probabilistic model. By providing a probability system $(S, \sigma(S), P)$ to the set of states S, we could define a stochastic linear time structure, $\mathbb{M} = (S, P, \mathbb{X}, \psi, \mathbf{I})$ such that to each realization of the stochastic sequence ψ, obtained by random drawing of a point ω in $S^{\mathbb{N}}$, corresponds an (ordinary) linear time structure $M_\omega = (S, \omega, \mathbf{I})$. The key for the consistency question is the fact that, as we proved, the existence of the support for a given formula p in L is equivalent with the property of a particular stochastic sequence to obey to the strong law of large numbers. As the sequence corresponding to formula p is constructed, using appropriate transformations, from the stochastic sequence ψ, we studied the necessary conditions for ψ which assures the applicability of SLLN.

To obey the law of large numbers, a sequence must satisfy regularity conditions relating to two distinct factors: the probability of extreme values (limited by bounding absolute moments) and the degree of dependence between coordinates. In our case, because the absolute moments of the sequence corresponding to a formula p are bounded by 0 and 1, the only factor we could change was the degree of dependence. And for all considered cases – the independence case, the mixing case (the degree of dependence converges to zero if the distance between variables converges to ∞) and near-epoch dependence case (a function of a mixing sequence with an infinite number of parameters) – we succeeded to

show that the linear time structure $M_\omega = (S, \omega, \mathbf{I})$ is consistent almost sure (i.e. the set of points $\omega \in S^{\mathbb{N}}$ for which M_ω is not consistent has the probability zero).

In the last section of this chapter we showed that the consistency problem for granular time structures, as defined in Chapter 4, may be solved in an analogous manner in our probabilistic framework. Even in this more complex situation, implying sequences of random vectors, we could prove that for all previous enumerated cases of dependence, the linear granular time structure $M_{\mu[\omega]} = (2^S, \mu[\omega], \mathbf{I}^\mu)$ is consistent almost sure.

CHAPTER VI

TEMPORAL META-RULES

As we mentioned in Section 2.2.2, the second step of the phase two of the methodology for temporal rule extraction is an inference process designed to obtain temporal meta-rules. A temporal meta-rule is a temporal rule template in accordance with Definition 3.4, but supposed to have a small variability of the estimated confidence among different models. Therefore, a temporal meta-rule may be applied with the same confidence in any state, complete or incomplete. To obtain such temporal rules, we apply strategies which cut irrelevant relational atoms, according with some criterions, from the implication clauses of temporal rule templates obtained during the first induction process. The strategies and the criterions are derived from the process of rules' generalization, applied by the C4.5 system.

The process of inferring temporal meta-rules is related to a new approach in data mining, called *higher order mining*, i.e. mining from the results of previous mining runs. According to this approach, the rules generated by the first induction process are first order rules and those generated by the second inference process (i.e. temporal meta-rules) are higher order rules. The formalism described in Chapter 3 does not impose what methodology to use to discover first order temporal rules. As long as these rules satisfy the syntactic form described in Definition 3.4, the strategy (including algorithms, criterions, statistical methods) developed to infer temporal meta-rules might be applied.

6.1 Lower Confidence Limit Criterion

Suppose that for a given model $\tilde{M}$ we dispose of a set of temporal rules templates, extracted from the corresponding classification tree. It is very likely that some temporal rules

templates contain implication clauses that are irrelevant, i.e. after their deletion, the general interpretation of the templates remain unchanged (*Remark*: in the following, by the notion "implication clause" we consider a relational atom prefixed by the temporal connective X_{-k}). In the frame of a consistent linear time structure M, it is obvious that we cannot delete an implication clause from a temporal rule template (denoted TR) if the resulting template (noted TR^-) has a lower confidence. But for a given model $\tilde{M}$, we calculate an estimate, $conf(TR, \tilde{M})$, of the confidence $conf(TR, M)$. Supposing that it is possible to establish a confidence interval for $conf(TR, M)$, the following approach can be applied : we accept to delete an implication clause from TR if and only if the lower confidence limit of $conf(TR^-, \tilde{M})$ is greater than the lower confidence limit of $conf(TR, \tilde{M})$.

Establishing a confidence interval for a parameter means implicitly that we are working inside a probabilistic model. In the previous chapter we have shown how we can "immerse" a first-order temporal logic in a probabilistic framework. In the following we consider that the linear time structure M is a realization of a stochastic time structure $\mathbb{M}$, for which the stochastic process ψ is either independent, or α-mixing, or L_2-NED. Therefore, M is an almost sure consistent time structure.

The degree of dependence for the process ψ determines how the confidence interval for the parameter $conf(TR, M)$ is calculated. The simplest situation is for the independence case, where the classical central limit theorem (see Appendix A, section A.5) can be applied to all sequences corresponding to a given formula. Therefore, because the estimator $conf(TR, \tilde{M})$ is the ratio $\#A/\#B$ (see Def. 3.20), a confidence interval for this value is constructed using a normal distribution depending on $\#A$ and $\#B$ (more precisely, the normal distribution has mean $\pi = \#A/\#B$ and variance $\sigma^2 = \pi(1-\pi)/\#B$). The lower limit of the interval is $L_\alpha(A, B) = \pi - z_\alpha\sigma$, where z_α is a quantile of the normal distribution for a given confidence level α. In the following, $L_\alpha(TR, \tilde{M})$ will denote the lower bound of the confidence interval for $conf(TR, \tilde{M})$, having the coverage $1 - \alpha$.

The problem becomes more difficult for ψ a proper dependent process (α-mixing or

L_2-NED), because even this degree of dependence permits the application of law of large numbers, it is not a strong enough assumption to yield a central limit theorem. Moreover, the conditions that a NED function of strong mixing process must satisfy to apply CTL are complicated and very difficult to verify in practice (see Davidson [1994]). And if these conditions are satisfied, the convergence rate is so slow that using the normal distribution as approximation in the expression for confidence interval bounds induces a lower accuracy.

The solution to this problem comes from a new approach in computational statistics, called the bootstrap approach. It is a method for estimating the distribution of an estimator or test statistic by resampling own's data or a model estimated from the data [Efron and Tibshirani, 1993, Davison and Hinkley, 1997]. Under conditions that hold in a wide variety of applications, the bootstrap provides approximations to distributions of statistics, coverage probabilities of confidence intervals, and rejection probabilities of tests that are at least as accurate as the approximations of first-order asymptotic distribution theory. The methods that are available for implementing the bootstrap and the improvements in accuracy that it achieves relative to first-order asymptotic approximations depend on whether the data are a random sample from a distribution or a time series. If the data are a random sample, then the bootstrap can be implemented by sampling the data randomly with replacement or by sampling a parametric model of the distribution of the data. For dependent data, the data generating process is often not fully specified and so there exists no unique natural way for resampling. The resampling should be carried out in such a way that the dependence structure should be captured. The most popular bootstrap methods for dependent data are block, sieve, local, wild and Markov bootstrap and subsampling. They all are nonparametric procedures [Buhlmann, 2002, Härdle et al., 2003, Politis, 2003]. In our opinion, the most adequate resampling method for sequences derived from the stochastic process ψ is the block bootstrap, which has turned out to be a very powerful method for dependent data [Liu and Singh, 1992, Politis and Romano, 1994]. It does not achieve the accuracy of the bootstrap for i.i.d. data but it outperforms the subsampling. It works reasonably well

under very weak conditions on the dependency structure and no specific assumption must be made on the structure of the data generating process.

Once the bootstrap resampling time structure models $\tilde{M}_k$, $k = 1..R$, are generated, the bootstrap estimators of confidence $\varphi_k = conf(TR, \tilde{M}_k)$ are calculated and a confidence interval for $conf(TR, M)$, using a confidence level α, is determined. There exist two basic approaches for the construction of confidence regions, one based on bootstrap asymptotic pivots and the other based on bootstrap percentiles. We will denote the lower bound of the bootstrap confidence interval for $conf(TR, \tilde{M})$, having the accuracy α, as $\widetilde{L_\alpha}(TR, \tilde{M})$.

The algorithm which generalizes a single temporal rule template TR, by deleting a single implication clause, may calculate the lower bound of the confidence interval either using the normal approximation, or using the bootstrap approach. The version for the normal approximation is presented in the following:

Algorithm 2 ***1-delete (normal approximation)***

Step 1 *Let $TR = H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$ be a temporal rule template. Let $\aleph = \bigcup_j \{C_j\}$, where C_j are all the implication clauses that appear in the body of the template. Rewrite TR, by an abuse of notation, as $\aleph \mapsto H_{m+1}$. If $n = \#\aleph$, denote by $C_1, \ldots, C_n$ the list of all implication clauses from $\aleph$.*

Step 2 *For each $i = 1, \ldots, n$ do*

$\aleph^- = \aleph - C_i, \quad TR_i^- = \aleph^- \mapsto H_{m+1}$

$A = \{i \in \tilde{T} | i \Rightarrow \aleph \wedge H_{m+1}\}, B = \{i \in \tilde{T} | i \Rightarrow \aleph\}$

$A^- = \{i \in \tilde{T} | i \Rightarrow \aleph^- \wedge H_{m+1}\}, B^- = \{i \in \tilde{T} | i \Rightarrow \aleph^-\}$

$conf(TR, \tilde{M}) = \#A/\#B, \ conf(TR_i^-, \tilde{M}) = \#A^-/\#B^-$

If $L_\alpha(A, B) \leq L_\alpha(A^-, B^-)$ then store TR_i^-

Step 3 *Keep only the generalized temporal rule template TR_i^- for which $L_\alpha(A^-, B^-)$ is maximal.*

The core of the algorithm is the **Step 2**, where the sets used to estimate the confidence of the initial rule template, TR, and of the generalized rule template, TR^-, i.e. A, B, A^- and B^-, are calculated. The complexity of this algorithm is linear in n (or $O(n)$).

Of course, more than one implication clause may be deleted from TR, which justifies the necessity of the following definition.

Definition 6.1 *(**Lower Confidence Limit**) Giving $\tilde{M}$ a consistent time structure model and TR a temporal rule template, the temporal meta-rule inferred from TR according to the lower confidence limit criterion (or LCL) is the temporal rule template TR_{LCL} with a maximum set of implication clauses deleted from TR and having the maximum lower confidence limit greater than $L_\alpha(TR, \tilde{M})$.*

An algorithm designed to find the largest subset of implication clauses that can be deleted will have an exponential complexity. A first solution is to use an exhaustive search when the number of implication clauses is small and some near-optimal approaches (greedy search, simulated annealing, etc.) when it is not. Another solution is to use the Algorithm 2 in successive steps until no more deletions are possible, but without having the guarantee that we will get the global maximum.

As example, consider the first temporal rule template from Table 3 and suppose that $\#A = 20$ and $\#B = 40$. Therefore, the estimate $conf(TR, \tilde{M})$ of the true confidence has the value 0.5, and the lower bound of the confidence interval for $\alpha = 0.95$ is $L_{0.95}(20, 40) = 0.345$. Looking at Table 4 – obtained by analyzing a first application of Algorithm 2 – we find two implication clauses which could be deleted (the first and the second) with a maximum $L_\alpha(A^-, B^-)$ given by the second clause. As a remark, by deleting the first implication clause, the resulting temporal rule template has an estimate of the confidence (0.489) less than of the original rule template (0.5), but a lower bound of the confidence interval (0.349) greater than $L_{0.95}(TR, \tilde{M})$. This case justifies the use of the confidence interval limits rather than the estimator $conf(TR, \tilde{M})$ during the inference process. If we

Table 4: Parameters calculated in Step 2 of the Algorithm 2 by deleting one implication clause from the template $X_{-3}(y_1 = start\_peak) \wedge X_{-3}(y_2 < 11) \wedge X_{-1}(y_1 = start\_peak) \mapsto X_0(y_1 = start\_valley)$

Deleted implication clause	$\#A^-$	$\#B^-$	$co(TR_i^-, \tilde{M})$	$L_\alpha(A^-, B^-)$
$X_{-3}(y_1 = start\_peak)$	24	49	0.489	0.349
$X_{-3}(y_2 < 11)$	30	50	0.60	0.464
$X_{-1}(y_1 = start\_peak)$	22	48	0.458	0.317

apply again the Algorithm 2 on the template

$$X_{-3}(y_1 = start\_peak) \wedge X_{-1}(y_1 = start\_peak) \mapsto X_0(y_1 = start\_valley)$$

(denoted TR^-), we find that no other implication clause can be deleted, i.e. TR^- is the temporal meta rule according to the criterion LCL inferred from the temporal rule template

$$X_{-3}(y_1 = start\_peak) \wedge X_{-3}(y_2 < 11) \wedge$$
$$X_{-1}(y_1 = start\_peak) \mapsto X_0(y_1 = start\_valley).$$

For the situation where the confidence interval is calculated using the bootstrap approach, the only changes we must perform in the inference process of LCL temporal meta-rules is at an algorithmic level. As we can see, the version of the algorithm using bootstrap methods (Algorithm 3) contains a supplementary step, **Step 1'**, where the lower bound of the bootstrap confidence interval for $conf(TR, \tilde{M})$ is calculated. All the resamples time structure models $\tilde{M}_j$, $j = 1..R$, generated in **Step 1'**, are used in **Step 2** to obtain the lower bound of the bootstrap confidence interval of the meta-rule TR^-, i.e. $\widetilde{L_\alpha}(TR^-, \tilde{M})$.

Algorithm 3 ***1-delete (bootstrap approach)***

Step 1 *Let $TR = H_1 \wedge \cdots \wedge H_m \mapsto H_{m+1}$ be a temporal rule template. Let $\aleph = \bigcup_j \{C_j\}$ the set of all implication clauses that appear in the body of the template. Rewrite TR as $\aleph \mapsto H_{m+1}$. If $n = \#\aleph$, denote by $C_1, \ldots, C_n$ the list of all implication clauses from $\aleph$.*

Step 1' *If $\tilde{M} = (\tilde{T}, \tilde{x})$ is a time structure model, then generate R resamples $\tilde{M}_j = (\tilde{T}_j, \tilde{x}_j)$, $j = 1..R$, by applying the block bootstrap resampling method on the sequence $\tilde{x}$.*

For each $j = 1 \ldots R$ *do*

$A_j = \{k \in \tilde{T}_j \,|\, k \Rightarrow \aleph \wedge H_{m+1}\}$, $B_j = \{k \in \tilde{T}_j \,|\, k \Rightarrow \aleph\}$, $\varphi_j = conf(TR, \tilde{M}_j) = \#A_j / \#B_j$

If $\varphi_{(i)}$ *represents the value on the* i^{th} *position in the ordered sequence* $\{\varphi_j\}$ *then the lower bound of the bootstrap confidence interval is* $\widetilde{L_\alpha}(TR, \tilde{M}) = \varphi_{R\cdot\alpha/2}$

Step 2 *For each* $i = 1, \ldots, n$ *do*

$\aleph^- = \aleph - C_i, \quad TR_i^- = \aleph^- \mapsto H_{m+1}$

For j=1,. . . ,R

$A_j^- = \{k \in \tilde{T}_j \,|\, k \Rightarrow \aleph^- \wedge H_{m+1}\}, B_j^- = \{k \in \tilde{T}_j \,|\, k \Rightarrow \aleph^-\}$

$\vartheta_j = conf(TR_i^-, \tilde{M}_j) = \#A_j^- / \#B_j^-$

$\widetilde{L_\alpha}(TR^-, \tilde{M}) = \vartheta_{R\cdot\alpha/2}$

If $\widetilde{L_\alpha}(TR, \tilde{M}) \leq \widetilde{L_\alpha}(TR_i^-, \tilde{M})$ *then store* TR_i^-

Step 3 *Keep only the generalized temporal rule template* TR_i^- *for which* $\widetilde{L_\alpha}(TR_i^-, \tilde{M})$ *is maximal.*

6.2 Minimum Description Length Criterion

Suppose now that we dispose of two models, $\tilde{M}_1 = (\tilde{T}_1, \tilde{x}_1)$ and $\tilde{M}_2 = (\tilde{T}_2, \tilde{x}_2)$, and for each model we have a set of temporal rule templates with the same implicated clause H (sets denoted S_1, respectively S_2). Let S be a subset of the union $S_1 \cup S_2$. If $TR_j \in S$, $j = 1, \ldots, n$, $TR_j = H_{j1} \wedge \cdots \wedge H_{jm_j} \mapsto H$, then consider the sets

$$A_j = \{i \in \tilde{T}_1 \cup \tilde{T}_2 \,|\, i \Rightarrow H_{j1} \wedge \ldots \wedge H_{jm_j} \wedge H\}, \; \mathbf{A} = \bigcup_j A_j,$$

$$B_j = \{i \in \tilde{T}_1 \cup \tilde{T}_2 \,|\, i \Rightarrow H_{j1} \wedge \ldots \wedge H_{jm_j}\}, \; \mathbf{B} = \bigcup_j B_j,$$

$$\mathbf{C} = \{i \in \tilde{T}_1 \cup \tilde{T}_2 \,|\, i \Rightarrow H\}.$$

The performance of the subset S can be summarized by the number of *false positives* (time instants where the implication clauses of each template from S are true, but not the clause H) and the number of *false negatives* (time instants where the clause H is true, but none of the implication clauses of the templates from S). Practically, the number of *false positives* is $fp = \#(\mathbf{B} - \mathbf{A})$ and the number of *false negatives* is $fn = \#(\mathbf{C} - \mathbf{B})$. The worth of the subset S of temporal rule templates is assessed using the Minimum Description Length Principle (MDLP)[Rissanen, 1978, Quinlan and Rivest, 1989]. This provides a basis for offsetting the accuracy of a theory (here, a subset of templates) against its complexity. The principle is simple: a Sender and a Receiver have both the same models $\tilde{M}_1$ and $\tilde{M}_2$, but the states of the model of the Receiver are incomplete states (the interpretation of the implicated clause cannot be calculated). The sender must communicate the missing information to the Receiver by transmitting a theory together with the exceptions to this theory. He may choose either a simple theory with a great number of exceptions or a more complex theory with fewer exceptions. The MLD Principle states that the best theory will minimize the number of bits required to encode the total message consisting of the theory together with its associated exceptions. This is a particular instantiation of the MLDP, called two-part code version, which states that, among the set of candidate hypotheses $\mathcal{H}$, the best hypothesis to explain a set of data is one which minimizes the sum of the length, in bits, of the description of the hypothesis, and the length, in bits, of the description of the data encoded with the help of the hypothesis (which usually amounts to specifying the errors the hypothesis makes on the data). In the case where there are different hypotheses for which the sum attains its minimum, we select one with a minimum description length.

The following encoding schema is an approximation, since it attempts to find a lower limit on the number of bits in any encoding rather than choosing a particular encoding. The general ideas may be summarized as:

- To encode a temporal rule template from S, we must specify each of its implication clauses (the implicated clause being the same for all rules, there is no need to encoded

it). Because the order of the implication clauses is not important, the number of required bits must be reduced by $\kappa \log_2(m!)$, where m is the number of implication clauses and κ is a constant depending on encoding procedure.

- The number of bits required to encode the set S is the sum of encoding length for each template from S reduced by $\kappa \log_2(n!)$ (the order of the n templates from S is not important).

- The exceptions are encoded by indicating the sets *false positive* and *false negative*. A case covered by S is a state x_i from $\tilde{T}_1 \cup \tilde{T}_2$ for which there is at least a temporal rule $TR_j \in S$ such that $i \models H_{j1} \wedge \ldots \wedge H_{jm_j}$. Therefore, the set of cases covered by S is the set $\boldsymbol{B}$, whereas the set of uncovered cases is $\tilde{T}_1 \cup \tilde{T}_2 - \boldsymbol{B}$. If $b = \#\mathbf{B}$ and $N = \#(\tilde{T}_1 + \tilde{T}_2)$ then the number of bits required is

$$\kappa \log_2 \left(\binom{b}{fp} \right) + \kappa \log_2 \left(\binom{N-b}{fn} \right),$$

because we have $\binom{b}{fp}$ possibilities to choose the *false positives* among the cases covered by the rules from S and $\binom{N-b}{fn}$ possibilities to choose the *false negatives* among the uncovered cases.

The total number of bits required to encode the message (the theory represented by the set S of temporal rule templates and the exceptions representing the errors these rules make on data) is then equal to *theory bits + exceptions bits.* The set $S \subseteq S_1 \cup S_2$ for which this sum attains its minimum represents the set of temporal meta-rules inferred from $S_1 \cup S_2$, according to the following definition.

Definition 6.2 *(**Minimum Description Length**) Consider $k \geq 2$ time structure models $\tilde{M}_i$, $i = 1..k$ and, for H a given short constraint formula, let be S_i the set of temporal rule templates which are satisfied by $\tilde{M}_i$ and which imply the clause H. The set of temporal meta-rules inferred from $\boldsymbol{S} = \bigcup_{i=1}^{k} S_i$ according to the minimum description length criterion (or MDL) is the subset of $\boldsymbol{S}$ which minimizes the total encoding length.*

An algorithm designed to extensively search this subset S has an exponential complexity, but in practice (and especially when $\#\boldsymbol{S} > 10$) we may use different non-optimal strategies (hill-climbing, genetic algorithms, simulated annealing), having a polynomial complexity.

For a practical implementation of an encoding procedure in the frame of our formalism, we will employ a concept from the theory of probability, i.e. the entropy. Given a finite set S, the entropy of S is defined as $\mathcal{I}(S) = -\sum_{v \in S} freq(v) \cdot \log_2(freq(v))$, where $freq(v)$ means the frequency of the element v in S. This measure attains its maximum when all frequencies are equal. Consider now a model $\tilde{M}$, characterized by the states $s_1, \ldots, s_n$, where each state s_i is defined by a m-tuple $(v_{i_1}, \ldots, v_{i_m})$ (see Section 3.3 on how a tuple from the database of events is identified with a state s). Based on these states consider the sets A_j, $j = 1..m$, where $A_j = \bigcup_{i=1..n}\{v_{i_j}\}$ (see Figure 9). Let TR be a temporal rule template obtained by the first induction process from the model $\tilde{M}$ and let be $X_{-k}(y_j \, \rho \, c)$ an implication clause from this template, with $j \in \{1 \ldots m\}$ and ρ a relational symbol. We define the encoding length for $X_{-k}(y_j \, \rho \, c)$ to be $\mathcal{I}(A_j)$. The encoding length of a temporal rule template having k implication clauses is then equal with $\log_2(k)$ plus the sum of encoding length for each clause, reduced by $\log_2(k!)$ (order is not important), but augmented with $\log_2(m \cdot w(TR))$, where $w(TR)$ is the time window of the template. The last quantity $(\log_2(m \cdot h_{max}))$ expresses the encoding length of the maximum number of implication clause

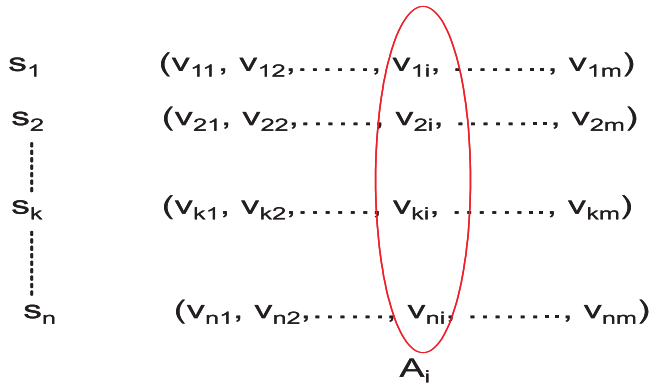


Figure 9: Graphical representation of the sets A_i

a temporal rule may have, which is evidently $m \cdot w(TR)$. Thus the minimum description length principle will favour, for identical number and equal encoding length of implication clauses, temporal rules with a smaller temporal dimension. Finally, the encoding length of q temporal rule templates is $\log_2(q)$ plus the sum of encoding length for each template, reduced by $\log_2(q!)$ (order is not important), whereas the encoding length of the exceptions is given by $\log_2\left(\binom{b}{fp}\right) + \log_2\left(\binom{N-b}{fn}\right)$.

As an example, consider the set of temporal rule templates from Table 3 having as implicated clause $X_0(y_1 = start\_valley)$. To facilitate the notation, we denote with TR_1, TR_2 and TR_3 the three concerned templates, written in this order in the mentioned Table. Therefore $S_1 = \{TR_1, TR_2\}$, $S_2 = \{TR_3\}$ and states used to calculate the entropy of the sets A_j, $j = 1..3$ are $\{s_1, \ldots, s_{100}, s_{300}, \ldots, s_{399}\}$. The encoding length for each subset $S \subseteq S_1 \cup S_2$ is presented in the last column of Table 5. It's values are the sum of the template encoding length (second column) and the exceptions encoding length (third column). As an observation, even if the set $\{TR_1, TR_2\}$ has more templates than the set $\{TR_3\}$, the encoding length for the two templates (14.34) is less than the encoding length of the last template (17.94). The conclusion to be drawn by looking at the last column of Table 5 is that the temporal meta rules, according to the MDL criterion and inferred from the set $\{TR_1, TR_2, TR_3\}$ (based on the states $\{s_1, \ldots, s_{100}\}$, $\{s_{300}, \ldots, s_{399}\}$) is the subset $S = \{TR_1, TR_2\}$.

Table 5: The encoding length of different subsets of temporal rule templates having as implicated clause $X_0(y_1 = start\_valley)$, based on states $\{s_1, \ldots, s_{100}\}$ and $\{s_{300}, \ldots, s_{399}\}$

Subset S	Templates length	Exceptions length	Total length
$\{TR_1\}$	8.88	70.36	79.24
$\{TR_2\}$	7.48	66.64	74.12
$\{TR_3\}$	17.94	67.43	85.37
$\{TR_1, TR_2\}$	14.34	46.15	60.49
$\{TR_1, TR_3\}$	24.82	41.2	66.02
$\{TR_2, TR_3\}$	23.42	38.00	61.42
$\{TR_1, TR_2, TR_3\}$	31.72	30.43	62.15

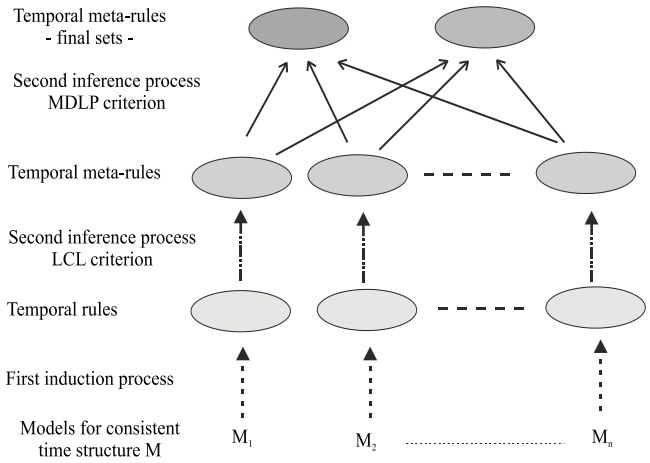


Figure 10: Graphical representation of the second inference process

Because the two definitions of temporal meta-rules differ not only in criterion (LCL, respectively MLD), but also in the number of initial models (one, respectively at least two), the second inference process is applied in two steps. During the first step, temporal meta-rules are inferred from each set of temporal rule templates based on a single model. During the second step, temporal meta-rules are inferred from each set of temporal rules created during step one and having the same implicated clause (see Fig. 10).

There is another reason to first apply the LCL criterion: the resulting temporal meta-rules are less redundant concerning the set of implication clauses and so the encoding procedures, used by MLD criterion, don't need an adjustment against this effect, as it was mentioned in the literature [Quinlan and Rivest, 1989].

6.3 Summary

The second inference process of the methodology described in Chapter 2 is related to a new approach in data mining, called *higher order mining*, i.e. mining from the results of previous mining runs. According to this approach, the rules generated by the first induction process are first order rules and those generated by the second inference process (i.e temporal meta-rules) are higher order rules.

Depending on the number of models at the input, the inference process is applied based

on two different criterions. If a single model is considered, then a temporal meta-rule is inferred from a first order rule template TR according to lower confidence limit criterion. From an algorithmic viewpoint, this means to delete a maximum number of implication clauses from TR and keeping, in the same time, the lower bound of the confidence interval for the new rule greater than the same measure of the initial rule. Using the probabilistic framework developed in Chapter 5, we proposed two approaches to calculate these bounds:

- for temporal data with a weak degree of dependence, an approach based on normal approximation and supported by the central limit theorem,
- for temporal data with a stronger degree of dependence, an approach based on bootstrap methods, using block bootstrap resampling and confidence intervals based on bootstrap percentiles.

If several models are considered, then a set of temporal meta-rules are inferred from the set of temporal rules (which are satisfied by at least one models and which implies all the same clause) according to the minimum description length criterion. From an algorithmic viewpoint, this means to find the subset S of rules such that the sum of the encoding length (in bits) of the rules from S and the encoding length of the exceptions (errors of these rules under all models) is minimal.

An important remark is that the second inference process, developed in the framework of a probabilistic first-order temporal logic, does not impose which methodology must be used to discover first order temporal rules. As long as these rules may be expressed according to Definition 3.4 , the strategy (here including algorithms, criterions, statistical methods), developed to infer temporal meta-rules may be applied.

CHAPTER VII

CONCLUSIONS

Data mining can 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. Temporal data mining is an important extension as it has the capability of mining activities rather than just states and, thus, inferring 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, the ability to suggest causes and effects that are overlooked when the temporal component is ignored or treated as a simple numerical attribute.

Among the different ways to represent knowledge as structured patterns, (decision tables, decision trees, decision rules, instant based representation, neural networks, Markov chains, etc.), the form we considered the most adequate for temporal/sequential data is the temporal rule. This choice is justified by the following considerations:

- Rules have a long history as a knowledge representation paradigm in cognitive modelling and artificial intelligence.
- Rules are inherently discrete in nature, and so are particularly well suited to modelling discrete and categorical-valued variables.
- Rules can be relatively easy for humans to interpret (at least relatively small sets of rules are), and have been found to be a useful paradigm for learning interpretable knowledge from data in machine learning search.

Our goal was to develop a methodology for extracting such rules, using techniques used in artificial intelligence/machine learning and statistics. This approach seemed to be very

important for us, because, as pointed out by Smyth [2001], there is a long and successful tradition of "marrying" ideas, theories, and techniques developed relatively independently within computer science and within statistics (graph-based models [Lauritzen and Spiegelhalter, 1988, Pearl, 1988], latent (hidden) variable models [Dunmur and Titterington, 1999, Hinton and Sejnowski, 1999], decision trees [Morgan and Sonquist, 1963, Quinlan, 1993] boosting algorithms [Freund and Schapire, 1997, Friedman et al., 2000]). Naturally, since computer science is a much younger discipline than statistics, the field of statistics has a much broader scope (in the context of learning from data). For example, there are large areas of data analysis such as spatio-temporal modelling, repeated measures/longitudinal data, and so forth, where machine learning has not had any appreciable impact. On the other hand, there are areas where a computational approach to learning has added concepts to data analysis that are relatively unrelated to anything in statistics, as Vapnik's theory of generalization based on margins [Vapnik, 1998]. By citing Padhraic Smyth (2001), "the future success of data mining will depend critically on our ability to integrate techniques for modelling and inference from statistics into the mainstream of data mining practice".

Mining paradigm \ Data type		Timestamped objects		
		Value	Event	Mining result
Apriori-like discovery	Ordering	Template-based mining	Sequence mining	Sequence maintenance
	No ordering		Discovery of temporal association rules	Rule maintenance and evolution
Classification	Ordering	Discovery of common trends	Sequence classification	
	No ordering		Temporal extensions to classification	

Figure 11: A Taxonomy of Temporal Mining Concepts [Roddick and Spiliopoulou, 2002]

According to the taxonomy of temporal knowledge discovery provided by Roddick and Spiliopoulou [2002] (see Fig. 11), temporal data mining research is categorized across three dimensions: *Datatype*, *Mining paradigm* and *Ordering*. Along the *Datatype* axis, the

methodology we proposed (see Chapter 2) has the great advantage to consider each type of timestamped objects:

- *Values*, represented by raw data and on which we apply a discretisation phase and a features extraction phase,
- *Events*, as the result of the first phase and on which we apply an induction process to extract local temporal rules, and
- *Mining results*, represented by sets of local rules and on which we apply an inference process to extract temporal meta-rules.

Along the *Mining paradigm* axis, our methodology can be seen as a combination between *Apriory-like discovery* and *Classification*. Indeed, the classification tree approach we applied to extract rules from the sequence of events (never used before in a consistent manner, as far as we were been able to ascertain) is clearly related to the classification set of methods. On the other hand, the resulted temporal rules are rather similar with temporal association rules, related in the above taxonomy with Apriory-like set of methods. Concerning the last axis, *Ordering*, we must remark again that our methodology is compatible with both ordered and non-ordered data. For the algorithm which "builds" classification trees (considered at the low level), there is no order in data, but for the algorithm which adds the temporal dimension to a rule (considered at a higher level), the data is ordered in time. Our contribution consisted in the development of a procedure for training set construction, which allows to capture the order of events for a given time window and to encode this order inside the index set of attributes.

The five primary components of a data mining algorithm, as described in Hand et al. [2001], are

- **The task**: categorization of types of data mining algorithms, as exploratory data analysis, pattern search, descriptive modelling and predictive modelling.

- **The Model Structure**: determining the underlying structure or functional forms that we seek in the data, as decision trees, Gaussian mixtures, association rules, and linear regression models.

- **The Score Function**: judging the quality of a fitted model or pattern on the basis of observed data, e.g., squared error for regression, classification error for classification, and so forth.

- **The Optimization Algorithm**: optimizing the score function and searching over different model and pattern structures.

- **The Data Management Strategy**: handling data access efficiently during the search/optimization

Within this framework, it is obvious that Chapter 2 and a major part of Chapter 6, which contain the detailed description of the methodology, cover especially the two primary computational components (optimization and data management). In this part of the thesis there is a model representation (decision tree) and a score function (error rate) used implicitly, but we considered as absolutely necessary to develop, in a second part of the thesis (Chapters 3, 4, 5) a more general model structure, allowing an abstract view of temporal rules, and based on first-order linear temporal logic. This choice is justified by the discrete structure of temporal data and by the temporal ontology we adopted (linearly ordered discrete instants). From our viewpoint, our major contribution is the definition of the concept of consistency for a linear time structure, which, even it seems straightforward as definition, has profound implications on the inference process of temporal rules. Indeed, consistency allows to define (in a consistent manner), for each formula in our language L, a "degree of truth" (or *general interpretation*) as a summary of the states where the formula was evaluated as true and, by consequence, to support inferences based on finite models. The only similar concept found in the literature is the one defined by Bacchus et al. [1996], in the context of a statistical knowledge base KB, and derived from the definition of "degree of

belief" ($Pr_\infty(\varphi|KB)$) for a formula φ. But contrary to our approach, the limit in the expression of $Pr_\infty(\varphi|KB)$ is taken along the size of the domain D where φ is defined and not along the number of states.

The sequence of states from a linear time structure (those on which the temporal logic is based), and its necessary properties to assures the existence of the support measure, suggests naturally the concept of a stochastic process. Consequently, a whole chapter (Chap. 5) is dedicated to the consequences of the "immersion" of the temporal logic formalism in a probabilistic framework. The most important result proved here concern again the consistency concept: the equivalence between the property of a particular time structure to be consistent and the property of a particular random sequence to obey the strong law of large numbers. This result demanded a laborious rationing especially for the cases which reflect faithfully the reality of temporal data: the existence of a certain degree of dependence between events over time (or, in a statistical language, an α-mixing dependence or a near-epoch dependence for the stochastic process).

The time, or more exactly, the scale of time is another direction in which the temporal logic formalism was extended. The concepts of temporal type, time granularity, *finer-than* relationship, event aggregation, function symbol family, are those on which the results from Chapter 4 are based. Using the notion of estimated support of a formula under a time structure model, we succeeded to define a granular temporal logic. In this model representation, the interpretation of a formula returns a value in the interval [0, 1], expressing the degree of truth. The most important theorems concern the mechanism of information transfer (here, the interpretation function) between worlds with different granularities (here, granular time structures). Once again, the concept of consistency proved to be fundamental: it ensures the preservation of the confidence of a temporal rule in all worlds derived from the same consistent, absolute world.

Finally, the same concept assures also the theoretical foundation for the process of temporal meta-rules inference, described in Chapter 6 and related to the higher order mining

approach. A first version of the inference process is based on the estimation of the bounds of a confidence interval, for the parameter *confidence* of a temporal rule template, satisfied by a finite time structure model. A second version is based on the minimum description length principle and concerns sets of temporal rules, satisfied by different finite models and implying the same short constraint formula. From a computational viewpoint, this chapter also treats the algorithms implementing the inference process and the possible solutions for an optimal application of these algorithms.

7.1 Future Work

It is obvious that a researcher can never say that he exhausted all the possible consequences of a given research problem. From this viewpoint, we want to make some remarks, all representing possible starting points for future works.

- *Interestingness*: We have not approached the difficult problem of deciding which temporal rules are of interest. The two metrics defined in our formalism, the support and the confidence, are interestingness metrics which can be misleading in some applications. As pointed out by Silberschatz and Tuzhilin [1996], not all patterns that are statistically dominant are of interest. Typically, background knowledge about the implication clauses and the implicated clause have a great influence in the interestingness of the rule and the discovery systems need to make it easy for the user to use such application-dependent criteria. As the large majority of interestingness measures are purely statistical criteria, (contingency tables, χ^2 scores, J-measures, cross-entropy), of great interest would be the analyze of the applicability of these measures in the framework of a stochastic linear temporal logic, and especially for dependency cases (as only the independence case was considered in literature).

- *Homogeneity hypothesis*: Even adopting a probabilistic framework for our formalism, a fundamental question remains : how to determine if a given linear time structure is consistent ? If the stochastic process ψ which concretize a fixed time structure

is i.i.d. then we can apply statistical tests of independence. But if the same process is α-mixing or L_2-NED dependent then these conditions are in general difficult to check. However, if the process follows a stationary Markov chain, then geometric ergodicity (for which there are techniques for checking) implies absolute regularity, which in turn implies strong mixing conditions. Another subsidiary question is how to determine if we are passing from a consistent model (as example, ψ i.i.d.) to another consistent model (as example, ψ α-mixing). In our opinion, the only feasible approach to this problem is the development of methods and procedure for detecting the change points in the model and, from a practical viewpoint, the analysis of the evolution of support/confidence of temporal meta-rules seems a very promising starting point.

- *Temporal scales*: When we defined the procedure for training set construction an implicit assumption was made: the events with the same index, from each sequence of predictor variables, start at the same time moment. This assumption is equivalent with the use of the same time scale for all sequences. If different scales (or time granularities) are applicable to different sequences – a situation often met in practice – then the possibility to encode the time in the set of attributes indexes is lost and the rules can not be transformed in temporal rules. Although we did not perform a deeper analyze of this situation, we think that the ideas developed in Chapter 4, concerning the mechanism of event aggregation, provide sufficient arguments for the use of a particular time scale, the least-upper bound temporal type, under the *finer-than* relationship, of the initial time scales.

"*Un jour ou l'autre, le temps vous donnera raison*"

(**Le Temps**)

APPENDIX A

THEORY OF STOCHASTIC PROCESSES

A random experiment is an action or observation whose outcome is uncertain in advance of its occurrence. Tosses of a coin, spin of a roulette wheel and observations of the price of a stock are familiar examples.

Definition A.1 *The basic space Ω is the set whose elements ω are the possible outcomes of the experiment.*

Definition A.2 *Let be Ω the basic space of an experiment. Let $\pi_A(\cdot)$ be a proposition; $\pi_A(\omega)$ is a proposition about ω which can be true or false. Then the event A is defined as the set $\{\omega : \pi_A(\omega)$ is true$\}$. The event A occurs iff the element ω selected is an element of the set A.*

Remark: When the outcome ω is identified completely, many different events may have occurred. This merely means that ω may belong to many subsets of the basic space.

Definition A.3 *A $\sigma-$algebra ($\sigma-$field) $\mathcal{X}$ is a class of subsets of Ω satisfying*

a) $\Omega \in \mathcal{X}$.

b) If $A \in \mathcal{X}$ *then* $A^c \in \mathcal{X}$.

c) If $\{A_n, n \in \mathbb{N}\}$ *is a sequence of* $\Omega-$*sets, then* $\bigcup_{n=1}^{\infty} A_n \in \mathcal{X}$.

A.1 Probability Spaces

If C is a collection of sets from Ω, the intersection of all $\sigma-$algebra containing C is called the $\sigma-$algebra generated by C, customarily denoted $\sigma(C)$. Given Ω a basic space and a class

of events $\mathcal{X}$ having the structure of a $\sigma-$algebra of subsets of Ω, the probability measure on $(\Omega, \mathcal{X})$ is a function $P : \mathcal{X} \to [0, 1]$ satisfying the following set of axioms:

A) $P(A) \geq 0$, for all $A \in \mathcal{X}$.

B) $P(\Omega) = 1$.

C) Countable additivity: for a disjoint collection $\{A_j \in \mathcal{X}, j \in \mathbb{N}\}$, $P(\bigcup_{j=1}^{\infty} A_j) = \sum_{j=1}^{\infty} P(A_j)$.

Definition A.4 *A probability system is a triple* $(\Omega, \mathcal{X}, P)$ *where* Ω *is a basic space,* $\mathcal{X}$ *is a* $\sigma-$*algebra on* Ω *and* P *is a probability measure on* $\mathcal{X}$.

The *conditional probability* of an event B given A is defined as $P(B|A) = P(A \cap B)/P(A)$, for $A, B \in \mathcal{X}$ and $P(A) > 0$. $P(\cdot|A)$ satisfies the probability axioms as long as P does and $P(A) > 0$. Events A and B are said to be *dependent* when $P(B|A) \neq P(B)$. A pair of events $A, B \in \mathcal{X}$ is said to be *independent* if $P(A \cap B) = P(A)P(B)$. A collection of events C is said to be *totally* independent if

$$P\left(\bigcap_{A \in \mathcal{I}} A\right) = \prod_{A \in \mathcal{I}} P(A)$$

for every subset $\mathcal{I} \subseteq C$.

A.2 Random Variables

Definition A.5 *The class of* `Borel` *sets on the real line (denoted* $\mathcal{B}$*) is the* $\sigma-$*algebra generated by the class of semi-infinite intervals of the form* $(-\infty, t]$ *for all* $t \in \mathbb{R}$.

Definition A.6 *If* g *is a real valued function of a single real variable, it is a* `Borel function` *iff the inverse image of every Borel set is a Borel set.*

Definition A.7 *Given* $(\Omega, \mathcal{X}, P)$ *a probability system, a real-valued function* $X : \Omega \to \mathbb{R}$ *is called a (real) random variable iff* $X^{-1}(B) \in \mathcal{X}$ *for all* $B \in \mathcal{B}$.

A random variable induces a probability measure on $\sigma-$algebra $\mathcal{B}$, denoted P_X, under the rule $\forall B \in \mathcal{B}, P_X(B) = P(X^{-1}(B))$. Furthermore, the class of inverse images of Borel sets

under the mapping of X is a $\sigma-$algebra of sets and is called the $\sigma-$algebra determined by X (denoted $\sigma(X)$).

Theorem A.1 *(Pfeiffer [1989], pg. 237) Suppose W is a random vector and g is a Borel function whose domain includes the range of W. Then $Z = g(W)$ is a random vector.*

Definition A.8 *A pair X, Y of random variables is (stochastically) independent iff for each pair of events, $E \in \sigma(X)$ and $F \in \sigma(Y)$, E and F are independent.*

Suppose $Z = g(X)$, where g is a Borel function. By Theorem A.1, any event determined by Z is an event determined by X. As a consequence, we have the following two important theorems.

Theorem A.2 *(Pfeiffer [1989], pg. 254) Suppose $\{X_t : t \in T\}$ is an independent class of random vectors. For each $t \in T$, let $Z_t = g_t(X_t)$, where g_t is a Borel function on the codomain of X_t. Then the class $\{Z_t : t \in T\}$ is independent.*

Theorem A.3 *(Pfeiffer [1989], pg. 255) Suppose $W = (X_1, \ldots, X_n)$ and $Z = (Y_1, \ldots, Y_m)$ are random vectors with the indicated coordinate random variables. If the class $\{X_i, Y_j : 1 \leq i \leq n, 1 \leq j \leq m\}$ is independent, then $\{W, Z\}$ is independent.*

The primary analytical tool for representing the probability distribution induced by a real random variable is as simple as it is useful. For each real x, we set the value $F_X(x)$ to be the amount of the probability mass located at or to the left of point x on the real line.

Definition A.9 *Given $(\Omega, \mathcal{X}, P)$ and X a real random variable on Ω, the cumulative distribution function (c.f.d) of X is the function $F_X : \bar{\mathbb{R}} \to [0, 1]$, where*

$$F_X(x) = P_X(-\infty, x) = P(X \leq x), x \in \mathbb{R}.$$

A.3 Expectation

Definition A.10 *The mathematical expectation $E(X)$ of real-valued random variable $X(\omega)$ in a probability space $(\Omega, \mathcal{X}, P)$ is given by*

$$E(X) = \int_{\Omega} X(\omega)\, dP(\omega) = \int_{\mathbb{R}} x\, dF_X(x)$$

provided the integrals exist.

The variance of X is defined as $Var(X) = E(X^2) - E(X)^2$, whereas the covariance of two r.v. X and Y is given by $Cov(X, Y) = E(XY) - E(X)E(Y)$. If A is an event from $\mathcal{X}$ then the random variable $\mathbf{1}_A$ which takes the value $\mathbf{1}_A(\omega) = 1$ for $\omega \in A$ and the value $\mathbf{1}_A(\omega) = 0$ for $\omega \notin A$ is called the indicator function of the set A. Therefore, $E(\mathbf{1}_A) = \int_A \mathbf{1}_A(\omega) dP(\omega) = P(A)$, $Var(\mathbf{1}_A) = P(A)P(\Omega - A)$ and $Cov(\mathbf{1}_A, \mathbf{1}_B) = P(A \cap B) - P(A)P(B)$.

Let X be an integrable r.v. on $(\Omega, \mathcal{X}, P)$ and $\mathbb{G}$ a σ-field contained in $\mathcal{X}$.

Definition A.11 *The conditional expectation (denoted $E(X|\mathbb{G})$) is any integrable, $\mathbb{G}$-measurable random variable having the property*

$$\int_G E(X|\mathbb{G})\, dP = \int_G X\, dP = E(X|G)P(G)$$

for all $G \in \mathbb{G}$.

Intuitively, $E(X|\mathbb{G})$ represents the prediction of $X(\omega)$ made by an observer having the information $\mathbb{G}$, when the outcome ω is realized.

The existence of a moment of order p (the quantity $E(X^p)$) requires the existence of the corresponding absolute moment (the quantity $E(|X|^p)$). If $E(|X|^p) < \infty$ for any real $p > 0$, X is sometimes said to belongs to the set L_p (of functions Lebesgue-integrable to order p), or to be L_p-bounded. Therefore, for $X \in L_p$, the L_p-norm of X is defined as

$$\| X \|_p = (E|X|^p)^{1/p}.$$

A.4 Stochastic Processes

Let $(\Omega, \mathcal{X}, P)$ be a probability space, let $\mathbb{T}$ be any set and let $\mathbb{R}^{\mathbb{T}}$ be the product space generated by tacking a copy of $\mathbb{R}$ for each element of $\mathbb{T}$. Then a *stochastic process* is a measurable mapping $x : \Omega \to \mathbb{R}^{\mathbb{T}}$, where

$$x(\omega) = \{X_\tau(\omega), \tau \in \mathbb{T}\}.$$

$\mathbb{T}$ is called the *index set* and the random variable $X_\tau(\omega)$ is called a *coordinate* of the process. A stochastic process can also be characterized as a mapping from $\Omega \times \mathbb{T}$ to $\mathbb{R}$. However, the significant feature of the given definition is the requirement of joint measurability of the coordinates.

Definition A.12 *A stochastic sequence is a stochastic process whose index set is countable and linearly ordered.*

Looking at the distribution of the sequence as a whole, the simplest treatment is to assume that the joint distribution of the coordinates is invariant with respect to the time index.

Definition A.13 *A random sequence is called* `strictly stationary` *if the sequences* $\{X_t\}_{t=1}^{\infty}$ *and* $\{X_{t+k}\}_{t=1}^{\infty}$ *have the same joint distribution, for every* $k > 0$.

Subject to the existence of particular moments, less restrictive versions of the conditions are also employed. If $\mu_t = E(X)$ and $\gamma_{kt} = Cov(X_t, X_{t+k})$ are well defined, the sequence is called *mean stationary* if $\mu = \mu_t$, and is called *covariance stationary* if $\gamma_{kt} = \gamma_k$, for all t. If the marginal distribution of X_t is the same for any t, the sequence $\{X_t\}$ is said to be *identically distributed*. This concept is different from stationarity. However, when a stochastic sequence is both independent and identical distributed (*i.i.d.*), this suffices for stationarity.

Much the largest part of stochastic process theory has to do with the joint distribution of sets of coordinates, under the general heading of dependence. From the various issues

relating exclusively to the marginal distributions of the coordinates, a special interest is giving to the conditions that limit the random behavior of a sequence as the index tends to infinity. Consider a sequence $\{X_n, n \in \mathbb{N}\}$ of real random variables. For each $\omega \in \Omega$, $\{X_n(\omega), n \in \mathbb{N}\}$ is a sequence of real numbers. Such a sequence may converge for some ω and diverge for others. If one denote D the set of ω for which the sequence diverge, it can be shown that D is a measurable set.

Definition A.14 *A sequence* $\{X_n, n \in \mathbb{N}\}$ *of random variable is said to converge* `almost surely`*, or to converge* `with probability one`*, iff the probability of the divergence set is zero.*

Definition A.15 *A sequence* $\{X_n, n \in \mathbb{N}\}$ *of random variable is said to* `converge in probability` *to random variable X if, for any* $\epsilon > 0$*, the probabilities of the events* $\{\omega : |X_n(\omega) - X(\omega)| < \epsilon\}$ *form real sequence converging to 1.*

A.4.1 Mixing

There are several ways to characterize the dependence between pairs of σ-subfield of events, but the following are the concepts that have been most commonly exploited in limit theory (Davidson [1994]).

Let be $(\Omega, \mathcal{X}, P)$ be a probability space and let $\mathcal{G}, \mathcal{H}$ be σ-subfields on $\mathcal{X}$; then

$$\alpha(\mathcal{G}, \mathcal{H}) = \sup_{G \in \mathcal{G}, H \in \mathcal{H}} |P(G \cap H) - P(G)P(H)|$$

is known as the strong mixing coefficient, and

$$\phi(\mathcal{G}, \mathcal{H}) = \sup_{G \in \mathcal{G}, H \in \mathcal{H}; P(G) > 0} |P(H|G) - P(H)|$$

as the uniform mixing coefficient. These are alternative measures of the dependence between the subfields $\mathcal{G}$ and $\mathcal{H}$. If the subfields $\mathcal{G}$ and $\mathcal{H}$ are independent, then $\alpha(\mathcal{G}, \mathcal{H}) = 0$ and $\phi(\mathcal{G}, \mathcal{H}) = 0$, and the converse is also true in the case of uniform mixing, although not

for strong mixing. Since

$$|P(G \cap H) - P(G)P(H)| \leq |P(H|G) - P(H)| \leq \phi(\mathcal{G}, \mathcal{H})$$

for all $G \in \mathcal{G}, H \in \mathcal{H}$, it is clear that $\alpha(\mathcal{G}, \mathcal{H}) \leq \phi(\mathcal{G}, \mathcal{H})$.

Consider a double infinite sequence $\{X_t, t \in \mathbb{Z}\}$ and define the family of subfields $\{\mathcal{X}_s^t, s \leq t\}$, where $\mathcal{X}_s^t = \sigma(X_s, \ldots, X_t)$ is the smallest σ-field on which the sequence coordinates from times s to t are measurable. A particularly important sub-family is the increasing sequence $\{\mathcal{X}_{-\infty}^t, t \in \mathbb{Z}\}$, which can be though of as, in effect, "the information contained in the sequence up to time t".

For a sequence $\{X_t(\omega)\}_{-\infty}^{\infty}$, let be $\mathcal{X}_{-\infty}^t = \sigma(\ldots, X_{t-1}, X_t)$ and $\mathcal{X}_{t+m}^{\infty} = \sigma(X_{t+m}, X_{t+m+1}, \ldots)$. The sequence is said to be *α-mixing* (or strong mixing) if $\lim_{n\to\infty} \alpha_m = 0$, where

$$\alpha_m = \sup_t \alpha(\mathcal{X}_{-\infty}^t, \mathcal{X}_{t+m}^{\infty}).$$

It is said to be ϕ-mixing (or uniform mixing) if $\lim_{n\to\infty} \phi_m = 0$, where

$$\phi_m = \sup_t \phi(\mathcal{X}_{-\infty}^t, \mathcal{X}_{t+m}^{\infty}).$$

Uniform mixing implies strong mixing, while the converse does not. Since the collections $\mathcal{X}_{-\infty}^t$ and $\mathcal{X}_{t+m}^{\infty}$ are respectively non-decreasing in t and non-increasing in t and m, the sequence $\{\alpha_m\}$ (respectively $\{\phi_m\}$) is monotone. Because these sequences may tend to zero at different rate, we say that a sequence $\{X_t(\omega)\}_{-\infty}^{\infty}$ is α-mixing (ϕ-mixing) of size $-\varphi_0$ if $\alpha_m = O(m^{-\varphi})$ for some $\varphi > \varphi_0$ (and similarly for ϕ_m).

A.4.2 Near-Epoch Dependence

Definition A.16 *(Davidson [1994], pg. 261) For a stochastic sequence $\{V_t\}_{-\infty}^{\infty}$, possibly vector -valued, on a probability space $(\Omega, \mathcal{X}, P)$, let $\mathcal{X}_{t-m}^{t+m} = \sigma(V_{t-m}, \ldots, V_{t+m})$, such that $\{\mathcal{X}_{t-m}^{t+m}\}_0^{\infty}$ is an increasing sequence of σ-fields. If, for $q > 0$, a sequence of integrable r.v.s $\{X_t\}_{-\infty}^{\infty}$ satisfies*

$$\| X_t - E(X_t|\mathcal{X}_{t-m}^{t+m}) \|_q \leq d_t \nu_m,$$

where $\nu_m \to 0$*, and* $\{d_t\}_{-\infty}^{\infty}$ *is a sequence of positive constants,* X_t *is said to be* `near-epoch dependent in` L_q `norm` *(*L_q*-NED) on* $\{V_t\}$*.*

We say that the sequence $\{X_t\}$ is L_q-NED of size $-b$ if $\nu_m = O(m^{-b-\epsilon})$, for $\epsilon > 0$. The role of the sequence $\{d_t\}$ is usually to account for the possibility of trending moments, and when $\| X_t - E(X_t) \|_p$ is uniformly bounded, we can set d_t equal with a finite constant for all t. Moreover, if this constant is chosen such that $d_t \leq 2 \| X_t - E(X_t) \|_q$, we can set $\nu_m \leq 1$ with no loss of generality.

Suppose that $(X_{1t}, \ldots, X_{kt}) = \boldsymbol{X}_t = g(\ldots, V_{t-1}, V_t, V_{t+1}, \ldots)$ is a k-vector of L_q-NED functions, and interest focuses on the scalar sequence $\{\phi_t(\boldsymbol{X}_t)\}$, where $\phi : \mathbb{D} \to \mathbb{R}$, $\mathbb{D} \subseteq \mathbb{R}^k$, is a Borel measurable function. This setup subsumes the important case $k = 1$, in which the question at issue is the effect of nonlinear transformations on the NED property. For the cases of sums and products of pairs of sequences there are specialized results.

THEOREM A.4 *(Davidson [1994], pg 267) Let* X_t *and* Y_t *be* L_q*-NED on* $\{V_t\}$ *of respective sizes* $-a_X$ *and* $-a_Y$*. Then* $X_t + Y_t$ *is* L_q*-NED of size* $-min(a_X, a_Y)$*.*

THEOREM A.5 *(Davidson [1994], pg 268) Let* X_t *and* Y_t *be* L_2*-NED on* $\{V_t\}$ *of respective sizes* $-a_X$ *and* $-a_Y$*. Then* X_tY_t *is* L_1*-NED of size* $-min(a_X, a_Y)$*.*

Also a useful result is the following:

THEOREM A.6 *(Davidson [1994], pg 268) If* X_t *is* L_q*-NED on* $\{V_t\}$*, so is* X_{t+k} *for* $0 < j < \infty$*.*

A.5 Central Limit Theorem

The "normal law of error" is the most famous result in statistics. If a sequence of random variables $\{X_t\}_1^{\infty}$ have means of zero, and the partial sums $\sum_{t=1}^{n} X_t$, $n \in \mathbb{N}$, have variances s_n^2 tending to infinity with n although finite for each finite n, then, subject to rather mild additional condition on the distributions and the sampling process,

$$S_n = \frac{1}{s_n^2} \sum_{t=1}^{n} X_t \xrightarrow{D} N(0, 1),$$

where $\xrightarrow{D}$ means "convergence in distribution", i.e. the distribution function of S_n converge pointwise for each $x \in \mathbb{R}$ to the normal distribution.

The simplest case is where the sequence $\{X_t\}$ is both stationary and independently drawn.

Theorem A.7 ***Lindeberg-Levy*** *(Davidson [1994], pg. 366) If $\{X_t\}$ is an i.i.d. sequence having zero mean and variance σ^2, then $S_n = n^{-1/2} \sum_{t=1}^{n} X_t/\sigma \xrightarrow{D} N(0,1)$.*

The Lindeberg-Levy theorem impose strong conditions, especially for the equality of distributions. The standard result for independent, non-identically distributed sequences is the Lindeberg-Feller theorem, which establishes that a certain condition on the distributions of the summands is sufficient, and in some circumstances also necessary.

Theorem A.8 ***Lindeberg*** *(Davidson [1994], pg. 369). Let the array $\{X_{nt}\}$ be independent with zero mean and variance sequence $\{\sigma_{nt}^2\}$ satisfying $\sum_{t=1}^{n} \sigma_{nt}^2 = E(S_n^2) = 1$. Then, $S_n \xrightarrow{D} N(0,1)$ if*

$$\lim_{n\to\infty} \sum_{t=1}^{n} \int_{\{|X_{nt}|>\epsilon\}} X_{nt}^2 dP = 0, \text{ for all } \epsilon > 0 \text{ (Lindeberg condition)} \tag{A.31}$$

The results concerning the central limit theorem for dependent process are all derived from the following fundamental theorem, due to McLeish [1974].

Theorem A.9 *Let $\{Z_{nt}, t = 1..r_n, n \in \mathbb{N}\}$ denote a zero-mean stochastic array, where r_n is a positive, increasing integer-valued function of n, and let*

$$T_{r_n} = \prod_{t=1}^{r_n} (1 + i\lambda Z_{nt}), \lambda > 0.$$

Then, $S_{r_n} = \sum_{t=1}^{r_n} Z_{nt} \xrightarrow{D} N(0,1)$ if the following conditions hold:

- *T_{r_n} is uniformly integrable,*
- *$E(T_{r_n}) \to 1$ as $n \to \infty$,*

- $\sum_{t=1}^{r_n} Z_{nt}^2 \xrightarrow{pr} 1 \; as \; n \to \infty,$
- $\max_{1 \le t \le r_n} |Z_{nt}| \xrightarrow{pr} 0 \; as \; n \to \infty.$

Bibliography

M. Abadi and J. Y. Halpern. Decidability and expressiveness for first-order logics of probability. *Information and Computation*, 112(1):1–36, 1994.

R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient Similarity Search In Sequence Databases. In D. Lomet, editor, *Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms (FODO)*, pages 69–84, Chicago, Illinois, 1993. Springer Verlag.

R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. S. P. Chen, editors, *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

S. Al-Naemi. A theoretical framework for temporal knowledge discovery. In *Proceedings of International Workshop on Spatio-Temporal Databases*, pages 23–33, Spain, 1994.

J. Allen, H. Kautz, R. Pelavin, and J. Tenenberg. *Reasoning About Plans*. CA: Morgan Kaufmann, 1991.

C. Antunes and A. Oliveiro. Temporal Data Mining: an overview. In *Workshop on Temporal Data Mining, KDD2001*, San Francisco, August 2001.

J. Augusto. *Razonamiento Rebatible Temporal (Defeasible Temporal Reasoning)*. PhD thesis, Departamento de Cs. de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, 1998.

J. C. Augusto. The logical approach to temporal reasoning. *Artificial Intelligence Revue*, 16(4):301–333, 2001.

F. Bacchus. On probability distributions over possible worlds. In *UAI '88: Proceedings of the Fourth Annual Conference on Uncertainty in Artificial Intelligence*, pages 217–226, 1988.

F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge*. MIT Press, Cambridge, Mass, 1990.

F. Bacchus, A. J. Grove, J. Y. Halpern, and D. Koller. From statistical knowledge bases to degrees of belief. *Artif. Intell.*, 87(1-2):75–143, 1996.

F. Bacchus and F. Kabanza. Using temporal logics to express search control knowledge for planning. *Artif. Intell.*, 116(1-2):123–191, 2000.

F. Barber and S. Moreno. Representation of continuous change with discrete time. In *Proceedings of the 4th International Conference on Temporal Representation and Reasoning (TIME97)*, pages 175 – 179, 1997.

Y. Bengio. *Neural Networks for Speech and Sequence Recognition Neural Networks for Speech and Sequence Recognition*. International Thompson Publishing Inc, 1996.

G. Berger and A. Tuzhilin. Discovering Unexpected Patterns in Temporal Data using Temporal Logic. *Lecture Notes in Computer Science*, 1399:281–309, 1998.

D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD Workshop*, pages 359–370, 1994.

C. Bettini, X. S. Wang, and S. Jajodia. A general framework for time granularity and its application to temporal reasoning. *Ann. Math. Artif. Intell.*, 22(1-2):29–58, 1998a.

C. Bettini, X. S. Wang, and S. Jajodia. Mining temporal relationships with multiple granularities in time sequences. *Data Engineering Bulletin*, 21(1):32–38, 1998b.

C. Bettini, X. S. Wang, S. Jajodia, and J.-L. Lin. Discovering frequent event patterns with multiple granularities in time sequences. *IEEE Trans. Knowl. Data Eng.*, 10(2):222–237, 1998c.

J. P. Bigus. *Data Mining with Neural Networks*. McGraw-Hill, 1996.

A. Bochman. Concerted instant-interval temporal semantics I: Temporal ontologies. *Notre Dame Journal of Formal Logic*, 31(3):403 – 414, 1990a.

A. Bochman. Concerted instant-interval temporal semantics II: Temporal valuations and logics of change. *Notre Dame Journal of Formal Logic*, 31(4):581 – 601, 1990b.

G. Boole. The Laws of Though. Macmillan, London, 1854.

L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks/ Cole Advanced Books & Software, 1984.

P. Buhlmann. Bootstraps for time series. *Statist. Science*, 17:52–72, 2002.

S. Card, J. MacKinlay, and B. Shneiderman, editors. *Readings in Information Visualisation*. Morgan Kaufmann, 1999.

R. Carnap. *Logical Foundations of Probability*. University of Chicago Press, Chicago, 1950.

S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext classification using hyperlinks. In *Proceedings of ACM-SIGMOD Int. Conf. Management of Data*, pages 307–318, Seattle, 1998.

X. Chen and I. Petrounias. A Framework for Temporal Data Mining. *Lecture Notes in Computer Science*, 1460:796–805, 1998.

X. Chen and I. Petrounias. Discovering Temporal Association Rules: Algorithms, Language and System. In *Proceedings of the 6th International Conference on Data Engineering*, page 306, San Diego, USA, 2000.

J. Chomicki and G. E. Saake. *Logics for Databases and Informatin Systems*. Kluwer Academic Publisher, Boston, 1998.

J. Chomicki and D. Toman. Temporal Logic in Information Systems. *BRICS Lecture Series*, LS-97-1:1–42, 1997.

E. Ciapessoni, E. Corsetti, A. Montanari, and P. S. Pietro. Embedding time granularity in a logical specification language for synchronous real-time systems. *Sci. Comput. Program.*, 20(1-2):141–171, 1993.

J. Clifford and A. Rao. A simple general structure for temporal domains. In *Temporal Aspects of Information Systems*. Elsevier Science, 1988.

P. Cohen. Fluent Learning: Elucidating the Structure of Episodes. In *Advances in Intelligent Data Analysis*, pages 268–277. Springer Verlang, 2001.

P. Cotofrei and K. Stoffel. Classification Rules + Time = Temporal Rules. In *Lecture Notes in Computer Science, vol 2329*, pages 572–581. Springer Verlang, 2002a.

P. Cotofrei and K. Stoffel. First Order Logic Based Formalism for Temporal Data Mining. In *IEEE ICDM02 Workshop on Foundation of Data Mining and Knowledge Discovery*, 2002b.

P. Cotofrei and K. Stoffel. A Formalism for Temporal Rules. In *Proceedings of the Workshop on Temporal Data Mining, KDD02*, pages 25–37, 2002c.

P. Cotofrei and K. Stoffel. Rule Extraction from Time Series Databases using Classification Trees. In *Proceedings of IASTED International Conference*, pages 327–332, Insbruck, Austria, 2002d.

P. Cotofrei and K. Stoffel. Higher order temporal rules. In *Proceedings of International Conference on Computational Science*, pages 323–332, St.-Petersburg, 2003.

P. Cotofrei and K. Stoffel. From temporal rules to temporal meta-rules. In *Procedings of 6th International Conference Data Warehousing and Knowledge Discovery, DaWaK 2004*, Lecture Notes in Computer Science, vol. 3181, pages 169–178, Zaragoza, Spain, 2004.

P. Cotofrei and K. Stoffel. Temporal granular logic for temporal data mining. In *Proceeding of IEEE International Conference on Granular Computing*, Beijing, China, 2005 (to appear).

D. Cuckierman and J. Delgrande. Towards a formal characterization of temporal repetition with closed time. In *Proceedings of TIME98*, pages 140 – 147. IEEE Computer Society Press, 1998.

G. Das, D. Gunopulos, and H. Mannila. Finding similar time series. In *Principles of Data Mining and Knowledge Discovery*, pages 88–100, 1997.

G. Das, K. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule Discovery from Time Series. In *Proceedings of the* 4^{th} *Conference on Knowledge Discovery and Data Mining*, pages 16–22, 1998.

J. Davidson. *Stochastic Limit Theory*. Oxford University Press, 1994.

J. Davidson and R. de Jong. Strong laws of large numbers for dependent and heterogeneous processes: a synthesis of new and recent results. *Econometric Reviews*, 16(3):251–79, 1997.

A. Davison and D. Hinkley. *Bootstrap Methods and their Applications*. Cambridge University Press, Cambridge, 1997.

B. de Finetti. *Theory of Probability: A Critical Introductory Treatment*. Wiley, New York, 1974-75.

A. Dekhtyar, M. I. Dekhtyar, and V. S. Subrahmanian. Temporal probabilistic logic programs. In *Proceedings of International Conference of Logic Programming*, pages 109–123, 1999a.

M. I. Dekhtyar, A. Dekhtyar, and V. S. Subrahmanian. Hybrid probabilistic programs: Algorithms and complexity. In *UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden*, pages 160–169, 1999b.

A. P. Dunmur and D. M. Titterington. Analysis of latent structure models with multidimensional latent variables. In *Statistics and Neural Networks : Advances at the Interface*, pages 165–194. Oxford University Press, 1999.

B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, London, 1993.

E. A. Emerson. Temporal and Modal Logic. *Handbook of Theoretical Computer Science*, pages 995–1072, 1990.

B. Erickson and P. Sellers. Recognition of patterns in genetic sequences. *Time Warps, String Edits and macromolecules: The Theory and Practice of Sequence Comparison*, 1983.

J. Euzenat. An algebraic approach to granularity in qualitative time and space representation. In *IJCAI (1)*, pages 894–900, 1995.

C. Evans. The macro-event calculus: representing temporal granularity. In *Proceedings of PRICAI*, Japan, 1990.

R. Fagin, J. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Information and Computation*, 87(1):78–128, 1990.

R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

R. Fagin and J. Y. Halpern. Uncertainty, belief, and probability. In *IJCAI*, pages 1161–1167, 1989.

R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. In *Proceedings of Third Annual Symposium on Logic in Computer Science*, pages 410–421, 1988.

C. Faloutsos and et al. A signature technique for similarity-based queries (extended abstract), 1997.

C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *SIGMOD Conference*, pages 419–429, 1994.

Y. A. Feldman. *Probabilistic programming logics*. PhD thesis, Weizmann Institute of Science, 1984.

J. E. Fenstad. Representations of probabilities defined on first order languages. In J. N. Crossley, editor, *Sets, Models and Recursion Theory*, pages 156–172, 1967.

Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

J. H. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–374, 2000.

N. Friedman, K. Murphy, and S. Russel. Learning the structure of dynamic probabilistic networks. In *Proceedings of the* 14^{th} *Conference on Uncertainty in Artificial Intelligence*, pages 139–147. AAAI Press, 1998.

H. Gaifman. Concerning measures in first order calculi. *Israel Journal of Mathematics*, 2: 1–18, 1964.

F. Giunchglia and T. Walsh. A theory of abstraction. *Artificial Intelligence*, 56:323–390, 1992.

D. Q. Goldin and P. C. Kanellakis. On similarity queries for time-series data: Constraint specification and implementation. In *Proceedings of International Conference on Principles and Practice of Constraint Programming*, pages 137–153, 1995.

G. Guimares. Temporal knowledge discovery for multivariate time series with enhanced self-organizing maps. In *Proceedings of the IEEE-INNS-ENNS Int. Joint Conference on Neural Networks*, pages 165–170. IEEE Computer Society, 2000.

I. Hacking. *Logic of Statistical Inference*. Cambridge University Press, 1965.

P. Haddawy. A logic of time, chance, and action for representing plans. *Artif. Intell.*, 80 (1-2):243–308, 1996.

P. Hall and C. Heyde. *Martingale Limit Theory and Its Application*. Probability and Mathematical Statistics. Academic Press, 1980.

J. Halpern. A logical approach to reasoning about uncertainty: a tutorial. In X. Arrazola, K. Korta, and F. J. Pelletier, editors, *Discourse, Interaction, and Communication*, pages 141–155. Kluwer, 1998.

J. Y. Halpern. An analysis of first-order logics of probability. In *IJCAI*, pages 1375–1381, 1989.

J. Y. Halpern and R. Pucella. A logic for reasoning about upper probabilities. *CoRR*, cs.AI/0307069, 2003.

C. Hamblin. Instants and intervals. In F. Haber, J. Fraser, and G. Muller, editors, *The Study of Time*, pages 324 – 328, New York, 1972. Springer Verlag.

J. Han, Y. Cai, and N. Cercone. Data-driven discovery of quantitative rules in databases. *IEEE Transactions on Knowledge and Data Engineering*, 5:29–40, 1993.

J. Han, G. Dong, and Y. Yin. Efficient Mining of Partial Periodic Patterns in Time Series Database. In *Proceedings of International Conference on Data Engineering*, pages 106–115, Sydeny, Australia, 1999.

J. Han, Y. Fu, W. Wang, K. Koperski, and O. Zaiane. Dmql: a data mining query language for relational databases. In *Proceedings of SIGMOD'96 Workshop Researcg Issues on Data Mining and Knowledge Discovery*, pages 250 – 255, Portland, 1996.

J. Han, W. Gong, and Y. Yin. Mining Segment-Wise Periodic Patterns in Time-Related Databases. In *Proceedings of the* 4^{th} *Conference on Knowledge Discovery and Data Mining*, pages 214–218, 1998.

J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, 2001.

D. Harel and Y. Feldman. A probabilistic dynamic logic. *Journal of Computer and System Science*, 28:193–215, 1984.

J. Hartigan and M. Wong. Algorithm 136: A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

G. Hinton and T. Sejnowski, editors. *Unsupervised Learning: Foundations of Neural Computation*. The MIT Press., 1999.

J. Hobbs. Granularity. In *Proceedings of the IJCAI-85*, pages 432 – 435, 1985.

F. Hoppner. Learning Temporal Rules from State Sequences. In *IJCAI Workshop on Learning from Temporal and Spatial Data*, pages 25–31, Seattle, USA, 2001.

F. Hoppner. Discovery of core episodes from sequences. In *Pattern Detection and Discovery*, pages 199–213, 2002.

K. Hornsby. Temporal zooming. *Transactions in GIS*, 5:255–272, 2001.

J. Hosking, E. Pednault, and M. Sudan. A statistical perspective on data mining. *Future generatiion Computer Systems*, 13:117–134, 1997.

C. Howson and P. Urbach. *Scientific Reasoning: The Bayesian Approach*. Open Court, 1993.

W. Härdle, J. Horowitz, and J. Kreiss. Bootstrap methods for time series. *International Statist. Review*, 71:435–459, 2003.

Y. Huang and P. Yu. Adaptive query processing for time series data. In *Proceedings of Knowledge Discovery in Database*, pages 282–286, San Diego, USA, 1999.

E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, UK, 2003.

M. W. Kadous. Learning comprehensible descriptions of multivariate time series. In *International Conference on Machine Learning*, pages 454–463, 1999.

P. Kam and A. W. Fu. Discovering Temporal Patterns for Interval-based Events. *Lecture Notes in Computer Science*, 1874:317–326, 2000.

H. Kamp. Events, instants and temporal reference. In Baurle, editor, *Semantics from Different Points of View*, pages 376 – 417. Springer Verlag, 1979.

K. Karimi and H. Hamilton. Finding Temporal Relations: Causal Bayesian Networks vs. C4.5. In *Proceedings of the 12th International Symposium on Methodologies for Intelligent Systems*, Charlotte, USA, 2000.

G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29:119–127, 1980.

H. J. Keisler. Probability quantifiers. In J. Barwise and S. Feferman, editors, *Model-Theoretic Logics*, Berlin, 1985. Springer-Verlag.

E. Keogh, S. Lonardi, and B. Chiu. Finding Surprising Patterns in a Time Series Database in Linear Time and Space. In *Proceedings of 8th ACM SIKDD International Conference on Knowledge Discovery and Data Mining*, pages 550–556, Edmonton, Canada, 2002a.

E. Keogh and M. J. Pazzani. An Enhanced Representation of Time Series which Allows Fast and Accurate Classification, Clustering and Relevance Feedback. In *Proceedings of the 4th Conference on Knowledge Discovery and Data Mining*, pages 239–243, 1998.

E. J. Keogh, S. Chu, D. Hart, and M. J. Pazzani. Iterative Deepening Dynamic Time Warping for Time Series. In *Proceedings of Second SIAM International Conference on Data Mining*, 2002b.

E. J. Keogh and M. J. Pazzani. Scalling up Dynamic Type Warping to Massive Datasets. In *Proceedings of the* 3^{rd} *European Conference PKDD*, pages 1–11, 1999.

C. Knoblock. *Generating Abstraction Hierarchies: an Automated Approach to Reducing Search in Planning*. Kluwer Academic Publishers, 1993.

D. Koller and J. Y. Halpern. Irrelevance and conditioning in first-order probabilistic logic. In *AAAI/IAAI, Vol. 1*, pages 569–576, 1996.

D. Kozen and R. Parikh. An Elementary Proof of Completeness for PDL. *Theoretical Computational Science*, pages 113–118, 1981.

H. E. Kyburg. Higher order probabilities and intervals. *International Journal of Approximate Reasoning*, 2:195–209, 1988.

V. S. Lakshmanan, N. Leone, R. Ross, and V. S. Subrahmanian. Probview: a flexible probabilistic database system. *ACM Trans. Database Syst.*, 22(3):419–469, 1997. ISSN 0362-5915.

K. B. Laskey. Mebn: A logic for open-world probabilistic reasoning. George Mason University Department of Systems Engineering and Operations Research, 2004.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *J. Roy. Statist. Soc. B*, 50:157 – 224, 1988.

T. Lin and C. Liau, editors. *Foundation of Data Mining and Knowledge Extraction*. Springer-Verlag, 2005 (to appear).

T. Y. Lin and E. Louie. Data mining using granular computing: fast algorithms for finding association rules. *Data mining, rough sets and granular computing*, pages 23–45, 2002.

W. Lin, M. A. Orgun, and G. J. Williams. Temporal Data Mining using Hidden Markov-Local Polynomial Models. In *Proceedings of the* 5^{th} *International Conference PAKDD, Lecture Notes in Computer Science*, volume 2035, pages 324 – 335, 2001.

Y. Lin. A commonsense theory of time. In Lakemeyer and N. (eds.), editors, *Foundations of Knowledge Representation and Reasoning*, pages 216 – 228. Springer Verlag, 1994.

R. Liu and K. Singh. Moving blocks jackknife and bootstrap capture weak dependence. In *Exploring the Limits of the Bootstrap*, pages 225–248. Wiley, New-York, 1992.

H. Loether and D. McTavish. *Descriptive and Inferential Statistics: An introduction*. Allyn and Bacon, 1993.

W. Loh and Y. Shih. Split Selection Methods for Classification Trees. *Statistica Sinica*, 7: 815–840, 1997.

W. Loh and N. Vanichsetakul. Tree-structured classification via generalized discriminant analysis. *Journal of the American Statistical Association*, 83(403):715–725, September 1988.

J. M. Long, E. Irani, and J. Slagle. Automating the discovery of causal relationships in a medical records database. *Knowledge discovery in databases*, pages 465–476, 1991.

J. Los. Remarks on the foundations of probability. In *Proceedings of International Congress of Mathematicians*, pages 225–229, 1963.

H. Lu, J. Han, and L. Feng. Stock movement prediction and n-dimensional inter-transaction association rules. In *Proc. ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 12:1–12:7, Seattle, Washington, June 1998.

D. Malerba, F. Esposito, and F. Lisi. A logical framework for frequent pattern discovery in spatial data. In *Proceedings of 5th Conference Knowledge Discovery in Data*, 2001.

S. Mangaranis. *Supervised Classification with Temporal Data*. PhD thesis, Computer Science Department, School of Engineering, Vanderbilt University, 1997.

I. Mani. A theory of granularity and its application to problems of polysemy and underspecification of meaning. In *Proceedings of the Sixth International Conference Principles of Knowledge Representation and Reasoning*, pages 245–255, 1998.

W. Maniatty and M. Zaki. A requirement analysis of parallel kdd systems. In *3rd Workshop on High Performance Data Mining*, 2000.

H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov.*, 1(3):259–289, 1997.

G. McCalla, J. Greer, and P. Barrie, J.and Pospisil. Granularity hierarchies. *Computers and Mathematics with Applications*, 23:363–375, 1992.

R. McConnell. ψ-s correlation and dynamic time warping: Two methods for tracking ice floes in sar images. *IEEE Transactions on Geoscience and Remote sensing*, 29(6):1004–1012, 1991.

D. McDermott. A temporal logic for reasoning about plans and actions. *Cognitive Science*, 6:101Ű155, 1982.

D. McLeish. Dependent central limit theorems and invariant principles. *Annals of Probability*, 2(4):620–628, 1974.

R. Michalski, I. Brakto, and M. Kubat. *Machine Learning and Data Mining: Methods and Applications*. John Wiley & Sons, New York, 1998.

H. Miller and J. Han. *Geographic Data Mining and Knowledge Discovery*. Taylor and Francis, 2000.

J. Morgan and R. Messenger. Thaid: A sequential analysis program for the analysis of nominal scale dependent variables. Technical report, Institute of Social Research, University of Michigan, Ann Arbor, 1973.

J. N. Morgan and J. A. Sonquist. Problems in the analysis of survey data and a proposal. *J. Am. Stat. Assoc*, 58(415-434), 1963.

N. Nilsson. Probabilistic logic. *AI Journal*, 28:71–87, 1986.

T. Oates, D. Jensen, and P. Cohen. Discovering rules for clustering and predicting asynchronous events. In *Predicting the Future: AI Approaches to Time-Series Problems*, pages 73–79, 1998.

B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic Association Rules. In *Proceedings of International Conference on Data Engineering*, pages 412–421, Orlando, USA, 1998.

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

P. Pfeiffer. *Probability for Applications*. Springer Texts in Statistics. Springer-Verlag, 1989.

G. Piatesky-Shapiro and W. Frawley. *Knowledge Discovery in Databases*. AAAI/MIT Press, 1991.

D. Politis. The impact of bootstrap methods on time series analysis. *Statist. Science*, 18: 219–230, 2003.

D. Politis and J. Romano. The stationary bootstrap. *J. Amer. Statist. Assoc.*, 89:1303–1313, 1994.

J. R. Quinlan. *C4.5: Programa for Machine Learning*. Morgan Kauffmann, San Mateo, California, 1993.

J. R. Quinlan and R. L. Rivest. Inferring decision trees using Minimum Description Length Principle. *Information and Computation*, 3:227–248, 1989.

L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–15, January 1986.

J. Rissanen. Modelling by Shortest Data Description. *Automatica*, 14:465–471, 1978.

J. F. Roddick and M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Trans. Knowl. Data Eng.*, 14(4):750–767, 2002.

J. Rodriguez, C. Alonso, and H. Boström. Learning first order logic time series classifiers: Rules and boosting. In *Proceedings of 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 299–308, 2000.

G.-C. Roman. Formal specification of geographic data processing requirements. *IEEE Trans. Knowl. Data Eng.*, 2(4):370–380, 1990.

B. Russell. On order in time. In *Proceedings of the Cambridge Philosophical Society*, volume 32, pages 216 – 228, 1936.

J.-D. Saitta, L.and Zucker. Semantic abstraction for concept representation and learning. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation*, pages 103–120, 1998.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 43–49, February 1978.

D. Scott and P. Krauss. Assigning probabilities to logical formulas. In J. Hintikka and P. Supper, editors, *Aspects of Inductive Logic*, North-Holland, Amsterdam, 1966.

Y. Shoham. Ten requirements for a theory of change. *New Generation Computing*, 3:467 – 477, 1985.

A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. Knowl. Data Eng.*, 8(6):970–974, 1996.

P. Smyth. Data mining at the interface of computer science and statistics. In *Data Mining for Scientific and Engineering Applications*, pages 35–61. Kluwer, 2001.

M. Spiliopoulou and J. Roddick. Higher order mining: Modelling and mining the results of knowledge discovery. In *Proceedings of the 2nd International Conference on Data Mining, Methods and Databases*, pages 309–320, UK, 2000.

StatSoft, Inc. Electronic statistics textbook, 2004. URL `http://www.statsoft.com/textbook/stathome.html`.

J. Stell and M. Worboys. Stratified map spaces: a formal basis for multi-resolution spatial databases. In *Proceedings of the 8th International Symposium on Spatial Data Handling*, pages 180–189, 1998.

V. S. Subrahmanian. *Principles of Multimedia Database Systems*. Morgan Kaufmann, 1998.

S. Tsumoto. Rule Discovery in Large Time-Series Medical Databases. In *Proceedings of the 3rd Conference PKDD*, pages 23–31. Lecture Notes in Computer Science, 1074, 1999.

R. Turner. *Logic for Artificial Intelligence*. John Wiley & Sons, 1984.

V. Vapnik. *Statistical Learning Theory*. Springer Verlag, 1998.

L. Vila. Ip: An instant-period based theory of time. In R. Rodriguez, editor, *Proceedings of the Workshop on Spatial and Temporal Reasoning in ECAI 94*, 1994.

P. Wolper. On the Relation of Pograms and Computations to Models of Temporal Logic. *Temporal Logic in Specifications*, LNCS 398:75–123, 1989.

Y. Yao. Granular computing: basic issues and possible solutions. In P. Wang, editor, *Proceedings of the 5th Joint Conference on Information Sciences*, pages 186–189, Atlantic City, New Jersey, USA, 2000. Association for Intelligent Machinery.

Y. Yao and N. Zhong. Potential applications of granular computing in knowledge discovery and data mining. In M. Torres, B. Sanchez, and J. Aguilar, editors, *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, pages 573–580, Orlando, Florida, USA, 1999. International Institute of Informatics and Systematics.

B. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proceedings of the Fourteenth International Conference on Data Engineering*, pages 201–208, Orlando, USA, 1998. IEEE Computer Society.

L. A. Zadeh. Information granulation and its centrality in human and machine intelligence. In *Rough Sets and Current Trends in Computing*, pages 35–36, 1998.

B. Zhang and L. Zhang. *Theory and Applications of Problem Solving,*. North-Holland, Amsterdam, 1992.

B. Zhang, L.and Zhang. The quotient space theory of problem solving. In *Proceedings of International Conference on Rough Sets, Fuzzy Set, Data Mining and Granular Computing*, pages 11–15, 2003.

G. Zweig and S. J. Russell. Speech recognition with dynamic bayesian networks. In *AAAI/IAAI*, pages 173–180, 1998.