Department of Computing

PhD candidate: Yip Chi Kin
Chief Supervisor: Prof. Keith Chan
Date: 17-12-2007

Discovering Patterns in Multiple Time-series

Download: http://yipchikin.net/phd/draft.pdf

Abstract

In the past there has been some methodologies for solving time-series data mining. Those previous works of multiple sequences matching mechanisms are complicated and lack of comprehensive application domains, especially in multiple streaming data. Here we deal with these restrictions by introducing a novel methodology for finding multiple time-series patterns. The model is evaluated the noise by synthetic dataset and implemented in real-life application.

The patterns finding are based on the discovery of *synchronous* and *sequential* sequences correlation among themselves, in order to detect intra and extra relationships of each streaming data. Cross-relationship patterns mining are discovered based on the target-oriented, *meta-code* granularities, event-driven and correlation-based approach, and which would present knowledge in causal relations and space-time association rules. This development of patterns finding has naturally led to the exploration of application domains, in which spatio-temporal context data mining could be used. In our experimental case study, we demonstrate meta-code discretization strategy in financial time-series dataset by using primary and secondary meta-code sequences mining. Hence, the significant causal relationships of temporal patterns are extracted underlying more effectively and efficiently in an alternative way (meta-code sequence).

Key words: Multiple Time-series, Meta-code, Causal Relationships, Temporal Relations, Causal fork, Temporal Pattern.

1. Introduction

1.1 Motivation

Multiple time-series analysis is applied in many disciplines. Most of researches about multiple subsequences mining are in traditional regression model [4], [31], [16] or heuristics (neural network model) [10], and also curve approximation within graph-based analysis or subsequences shape matching manner [35], [22], [36]. Although some researchers proposed multi-dimensional sequential pattern mining [29] based on mining frequent subsequences [2] approach, the application domain is limited by massive different data structures and episodes [24], [20], and some different approaches for multiple time-series mining applications [28], [13], [34], [37], [33] as well. In recent years, symbolic approach has been attracted great interest in diverse research fields. Megalooikonomou et al. [23] introduced time-series codebook by vector quantization. Hugueney et al. [12], [14] defined data-driven symbols in time-series transformation, and Lin et al. [18] discretized time-series by breakpoints and comparing them with euclidean distance measure. Mörchen [19] represents temporal data into categories corresponding to symbolic sequences on point-based or interval-based. All those are single series transformation individually, not applied in multi-dimensional or multivariate symbolization as a whole.

Although Mörchen and Ultsch [21] used multivariate data and Allen's relations [1] for time interval patterns mining, Allen's temporal logic (13 relations) is solely based on time intervals and does not include time points. Papapetrou et al. [30] using Freksa's [8] semi-intervals relations to develop their temporal logic (5 relations), and extended to 7-relations [27] for applying in DNA sequences to finding high-regions (H-regions) that are occurred frequently. Here, we focus meta-code symbolization on time code-based multi-sequences methodology to instead of interval relations, in order to simplify temporal pattern mining implementation.

Most of single time-series presents as a streaming data, and multi-dimensional streams could be formalized as spatio-temporal contexts. Unfortunately, most time-series are in different natures with independent numerical states or notations. To solve this problem, we use a time granularities [5] mechanism to discrete events into *meta-code* sequences. Oates and Cohen [25] introduced multi-tokens encoding for structure searching, but we are aimed to find out their causal relationships of each series, and presented to be a temporal pattern. For patterns discovery strategy, the initial ideas come from Reichenbach's macrostatistics [32] and Allen's interval-relation knowledge [1] to fulfill data mining goals.

1.2 Discovering Patterns Methodology

This paper describes *Multiple Meta-code Sequences Miner* (MMSM), a novel framework (cf. figure 1) for acquiring causation and association rules from historical logs of multiple streaming events. The Top-N-certainty temporal patterns which are represented in the form of association rule underlying causal relationships. The pattern mining consists of two main components, sequential and synchronous subsequence miner for solving code-based inputting dataset. Therefore, the essential pre-processing of symbolization multiple time-series is transforming event tokens to meta-codes through relation logic rules, and time-series tokens encoding using time granularities (time-step) discretization according to user-defined target seeking. The philosophy of algorithms is multiple subsequence relationships seeking through causality analysis.

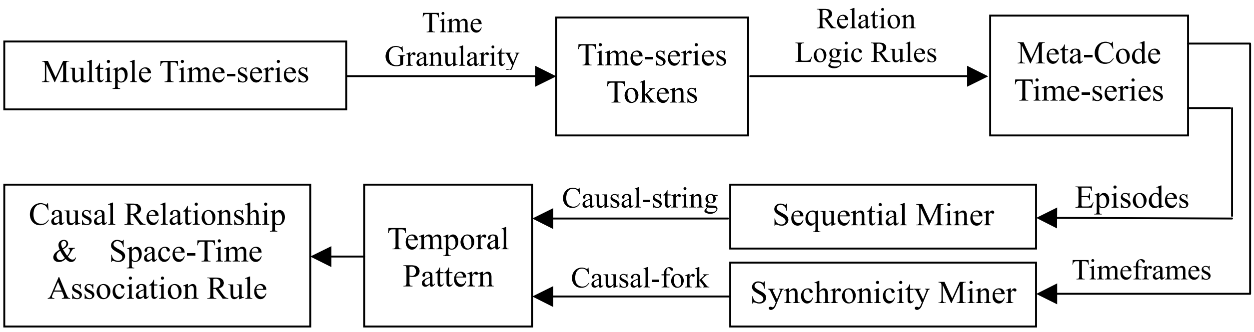


Figure 1. Multiple Meta-code Sequences (MMS) Miner

2. Mining Methodology

2.1 Time Granularities

Through examining time-series visually, a key parameter of temporal scale such as ten minutes, an hour, daily, etc. (named as a time-step) should be determined during discretization. We introduce the concept of temporal granularity very closely to spatio-temporal context. A fundamental of granule in a time-series called temporal cell, which allowed to be expressed at various scales. Each time-step is originally corresponded to the amount of time associated with each time point (time point defined as the smallest unit of time over which the system collects data, e.g., a second); for example, in real world day interval sequence groups into business day, where a business day is Monday through Friday, excluding holidays, and day interval sequence groups into business week as well. Our discretization policy for a time-series by coding of states is as follows. Suppose it is a given sequence s and a subsequence width n. Given $s = (x_1, \ldots, x_n)$, a subsequence of width n on s is a contiguous subsequence $(x_i, \ldots, x_{i+n-1})$. We form from s all synchronous subsequences $s_1, \ldots, s_i$ of width n, where $s_i = \{ x^{\tau}_n \mid x^{\tau} \in \mathbb{N} \text{ and } x^{\tau}_1, \ldots, x^{\tau}_n \mid n \in \mathbb{N} \}$, and nature number $\mathbb{N} = \{0, 1, 2, \ldots\}$. In a s_i time-series is composed of categorical values called tokens taken from a set τ_i. Each granule (***temporal-cell***) in each granularity is indexed by a positive integer to form time-step partition. For example in figure 2, sequence s_1 is composed of tokens drawn from the set $\tau_1 = \{ A_1, A_2, A_3, A_4, A_5 \}$. Here we denoted symbol ‘↨’ as a synchronicity connector, and ‘↳’ as a sequential connector. In figure 2, the events in t-column named ***timeframe***, which is denoted as ↨($A_1B_3C_3D_2$) at time t; or express timeframe as $t_k(A_pB_qC_rD_s)$ at time-step k. In the B-time-series, (B_3↳B_4↳B_2) is named ***episode***, which exists in time t to t+2, this episode is equivalent to $\{t_1(B_3), t_2(B_4), t_3(B_2)\}$ at time-step($t_1t_2t_3$). We express a multiple time-series as { ↨(timeframe$_1$) ↳↨(timeframe$_2$) … ↳↨(timeframe$_n$) } without indicate time-step, but ought to layout wildcards for empty time-step region. Alternative express as { *time-step$_k$*(timeframe$_k$), *time-step$_{k+1}$*(timeframe$_{k+1}$), … , *time-step$_{k+n-1}$*(timeframe$_{k+n-1}$) } with time-steps, where k is the starting time-step, n is the window size.

For instance, four subsequences (cf. figure 2), A-sequence contains 5 different tokens, B-sequence contains 4 tokens, etc. The complete temporal pattern (with shaded block in figure 2) would be presented as:

$$\{\ \updownarrow(A_2B_1C_2D_4) \hookrightarrow \updownarrow(A_3B_4C_4D_1) \hookrightarrow \updownarrow(A_4B_3C_2D_1)\ \}$$

$$\text{or}\quad \{\ t_4(A_2B_1C_2D_4),\ t_5(A_3B_4C_4D_1),\ t_6(A_4B_3C_2D_1)\ \}$$

$$=\ \{\ t_4(C_2B_1A_2D_4),\ t_5(B_4A_3D_1C_4),\ t_6(C_2A_4D_1B_3)\ \}$$

$$\neq\ \{\ t_4(A_2,B_1,C_2,D_4),\ t_5(A_3,B_4,C_4,D_1),\ t_6(A_4,B_3,C_2,D_1)\ \}$$

Time-series of event A	A_1	A_3	A_1	A_2	A_3	A_4	A_3	A_5	…	A_2	…	A_5
Time-series of event B	B_3	B_4	B_2	B_1	B_4	B_3	B_1	B_3	…	B_1	…	B_1
Time-series of event C	C_3	C_2	C_1	C_2	C_4	C_2	C_6	C_4	…	C_5	…	C_6
Time-series of event D	D_2	D_1	D_3	D_4	D_1	D_1	D_2	D_3	…	D_2	…	D_2
Time-step	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7	…	t+k	…	t+n-1

Figure 2. Tokens of multiple time-series from time t to $t+n-1$.

2.2 Meta-code Transformation

The definition of ***meta-code*** is "*Code about Code*" or "*Code about Token*" in this paper. Symbol "⇐" means "Transformation by rules", so formulate definition as "Meta-code ⇐ Token" or "Meta-code ⇐ Code". The framework of meta-code transformation is changing time-series into token sequence, and apply "*Relative Logic Rules*" (RLR) to transform tokens to be a meta-code sequence. Discretization of continuous variables can be considered a category of tokens (or intervals) estimation technique, in order to transform a general time-series be a state sequence. This type of symbolization has been done by many algorithms [25], [7], [15], [12], [11], [17]. Decision making of users is the main purpose of data mining in the real life, but the outcomes of data mining should be predefined some rules based on experts' hypothesis. Here, we predefined relation rules of cross-relationship mining; so-called this strategy is *Relative Logic Rules* (RLR) transformation. For example financial marketing data mining involves human behavior, technology, political aspects, etc. Those constraints are tightly embedded in domain-specific business rules and process with expertise. Discovering interesting patterns should be preprocessing raw data with business rules, regulations and targets of user. Otherwise the outcomes cannot achieve business benefits for supporting realistic and reliable decision-making.

Normally in state time-series (spatio-temporal contexts), a token is a specified discrete data type, which is hidden with same attribute in the time-series. For instance, a subsequence of hidden tokens may be {9,-2,5,0,3,7}, which could be transformed to be a token subsequence {+,–,+,Δ,+,+}, corresponding token set $\tau\{+,-,\Delta\}$. A meta-code usually describes a certain character of codes or tokens via RLR, and they may not be in the same attribute of time-series.

For instance, a category of meta-code M_{code}\{Buy, Sell, Hold\} with token set τ_{price}\{Up, Down\} and τ_{saving}\{High, Low\}. Financial expert setup a rule that: a "BUY" action will be actuated by conditions of saving is high and the price is down. Therefore, in this meta-code transformation the RLR denoted as $M_{code}(\text{Buy}) \Leftarrow \tau_{saving}(\text{High}) \wedge \tau_{price}(\text{Down})$.

There are two classes of data type called *primary* and *secondary* meta-code sequence for implementation of the data mining algorithms and the interpretation of the results. Primary meta-code (e.g. in section 5.1) is simple, Fundamental, factualness and identical attribute with higher contribution in two events, using straightforward RLR excluding extra-relationship. Secondary meta-code (e.g. in section 5.2) could be reduced computation, and RLR is highly context specific on case-by-case basis underlying some predefined extra-relationships.

It is conceivable that this meta-code transformation can be also applied in interval-relation knowledge mining, in order to spread relationship knowledge mining in multiple time-series. According to Allen's interval-based approach to temporal reasoning is presented on the time-line formats or state sequences (cf. figure 3), they were simplified [30] and applied [27] in DNA sequences mining by considering seven types of relations (cf. figure 4). We develop another code-based interval format of presentation to illustrate all the interval-relation knowledge (cf. figure 5).

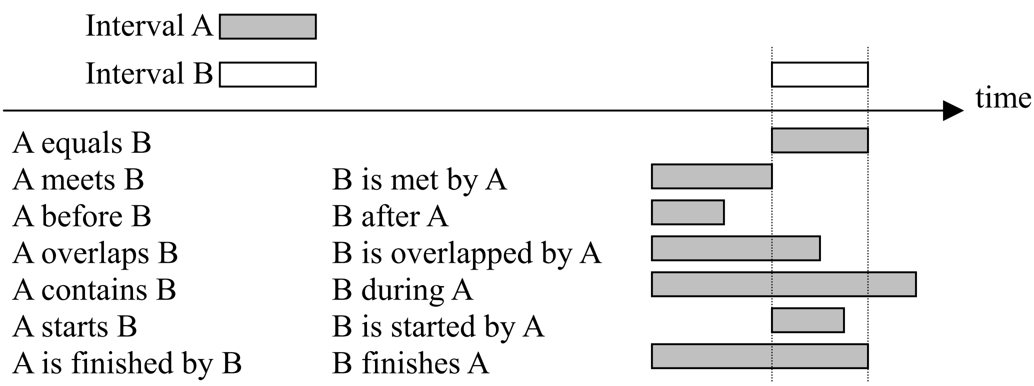


Figure 3. Allen's 13-relations interval relationships

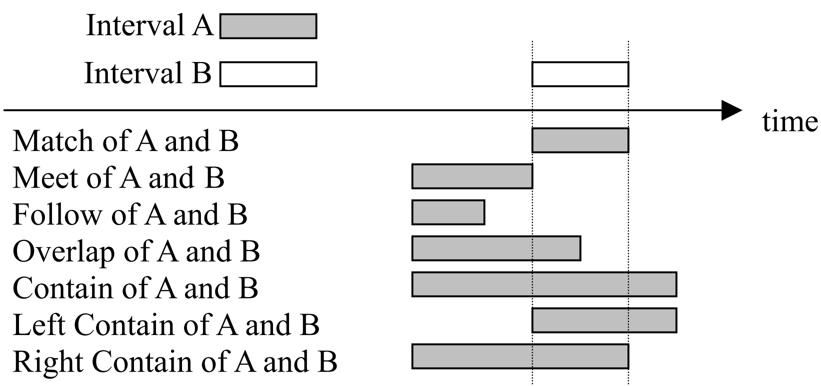


Figure 4. Papaetrou-Benson-Kollios 7-relations

Following Papaetrou-Benson-Kollios 7-relations, we present a ***code-interval relations*** system as a triplet ⌂AB(L_A,L_B,W_B) with $L_A, L_B, W_B \in \mathbb{N}$, where L_A is the number units (temporal cell) of event A (interval length of A), L_B is the number units of event B and W_B is the numbers of prefix wildcards of event B. Time interval is a set of temporal cells, and cross-relationship between two intervals denoted as symbol '⌂'. Defined the starting time point of interval A and wildcard W are at time-step t = 0. The temporal cells are user-defined and time-step could be shortened or prolonged to fulfil the relations matching mechanism. This system (cf. figure 5) could be covered all the cases of Allen's 13-relations knowledge.

We demonstrate the meta-code transformation in multiple time-series dataset. Here is the advantages of reasoning on the basis of code-intervals: (1) alignment, clustering, predictions, associations and patterns mining could be applied throughout the modeling; (2) dataset contexts are in a widespread domains applications, especially in multiple streaming and spatio-temporal datasets: (3) more simpler to represent knowledge about relations and implement computations in mining process.

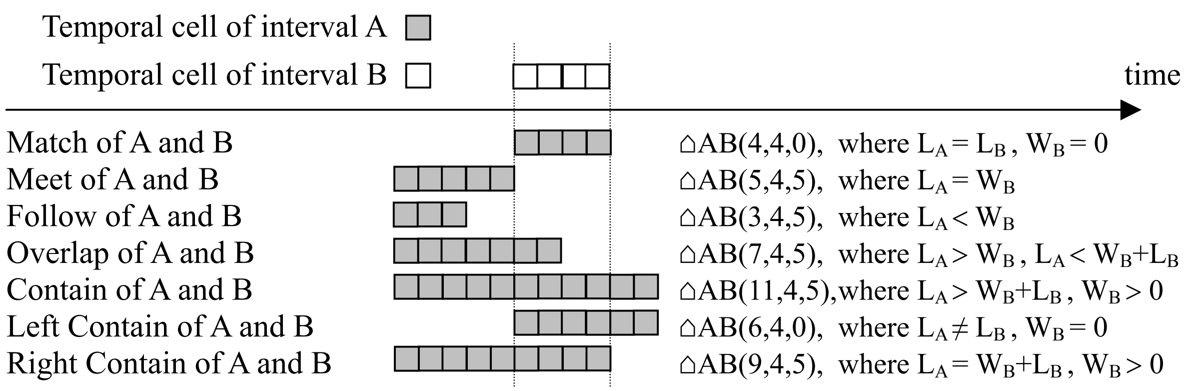


Figure 5. Code-interval Relations ⌂AB(L_A,L_B,W_B)

2.3 Causal relationships of Temporal Pattern

The Causal relationships could be described as latent structure of cause-effect relations (cf. [32], p.27). The relation of cause and effect is shown that temporal order, and even spatial order. For a small subset of natural phenomena, the cause-effect relationships are already known, either from our common knowledge, or from continuous scientific research. Thus, we know that lightning is usually followed by the sound of thunder, and force is a product of mass and acceleration. Some patterns are learned by humans surprisingly fast, such as lightning causes thunder. In many other cases, the relationship between our observations and system outputs is extremely hard to comprehend. The complexity of systems is growing rapidly, adding to the uncertainty and confusion. For example: Banks have data on millions of people and still they cannot identify in advance every unreliable customer. We note that information theory or probability theory is currently the official mathematical language of most disciplines that use causal modelling, including economics, epidemiology, sociology and psychology.

Causal relationship pattern is expressed in the form of cause-effect components in synchronicity and sequential (cf. figure 6). A set of ordered events (subsequence) form sequences is called episode, denoted as { $time_{t+k}(E_a)$, $time_{t+k+1}(E_b)$, $time_{t+k+2}(E_c)$ }, where event E_i are chronologically from time $t+k$ to $t+k+2$. Likewise, another set of events is occurred concurrently called timeframe, which denoted as $\{time_{t+k}(E_aE_dE_e)\}$, where events E_i in synchronicity at time $t+k$. Thus, by intercrossing both sets of events through a same event $time_{t+k}(E_a)$ to form a temporal pattern. This event $time_{t+k}(E_a)$ becomes a crossover correlative event. Sequential episode and synchronous timeframe could be formed by Reichenbach's macrostatistics mechanism (cf. [32], chapter 4).

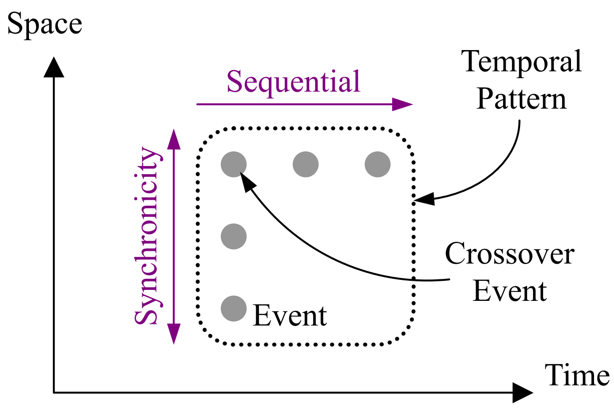


Figure 6. Causal-relationship Pattern

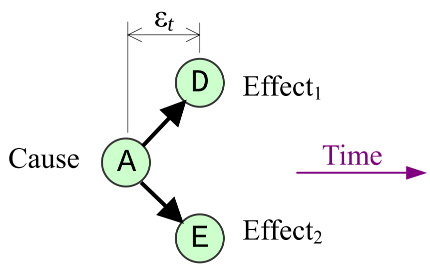


Figure 7a. Causal-fork relationship

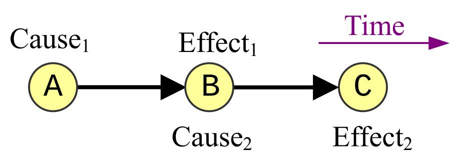


Figure 7b. Causal-chain relationship

Here is an example of a complete linkage of 3×3 mutual relations pattern (cf. figure 8a), which is a probabilistic relationship conceptual approach (all phenomena considered). A_2, A_3, B_1, C_2, … are the events of time-series, and they linked mutually by temporal relations. Obviously, this temporal pattern is almost impossible occurred in real world, and their relationships are very confusing and non-understanding relationships among themselves through over-linkages of casual events. At any rate, the significant relationships among events ought to be indicated clearly. By the way, we agree with Pearl's causality view (cf. [26], p.25): "Causal relationships are more 'stable' than probabilistic relationships. We expect such difference in stability because causal relationships are ontological, describing objective physical constraints in our world, whereas probabilistic relationships are epistemic, reflecting what we know or believe about world." Therefore, this complete mutual approach should be replaced by the causal-fork (cf. [32], p.159) and causal-chain (cf. [32], p.189) relationships connectivity underlying causal relationship.

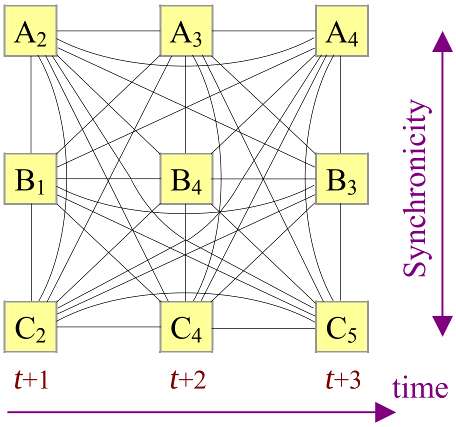


Figure 8a. Mutual relationships of pattern

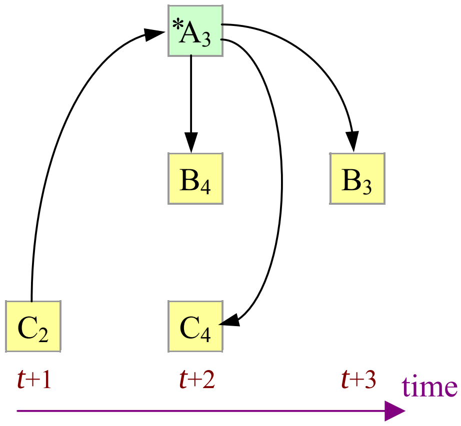


Figure 8b. Causal relationship of target pattern

According to user-defined target, we interest to discover knowledge from mining a target temporal pattern, in which should be contained synchronicity and sequential relations. Reichenbach's cause-effect relation will be used to formulate synchronous (cf. figure 7a) and sequential (cf. figure 7b) subsequences, in order to obtain an understandable temporal pattern. In figure 7a, Reichenbach defined a conjunctive fork to be a ***causal-fork*** in which D and E were conditionally independent on A. The causal-fork (timeframe) pattern mining concept is same as composition alignment [3] mentioned, so $\text{effect}_1, \ldots, \text{effect}_n$ are occurred concurrently without order. In the meantime, D and E are lagging behind very short interval ε_t (ε_t tends to zero) of A, which becomes a trigger event or symmetrical relation (cf. [32], p.29). Figure 7b contains two associated cause effects among A, B and C. The causal effect of A on B, causal effect of B on C, and B is an intermediate effect between cause A and effect C. This causal between-relation is known as a *causal-chain*. A causal-chain (cf. [32], p.193, Fig.32) may be arranged events in different time-step with more than two causal-chains underlying neighborhood relation, and sometimes it may be a fork (cf. [32], p.159, Fig.25) open toward the past with contiguous order time-steps (cf. figure 12b). A wildcard may be existed in a string, but not occurred in a chain. Thus, renamed them to be "***causal-string***" because all episodes are formulated into unique time-step in sequential order and performing like a chain, even though it is a fork. Here we introduce a *crossover correlative event* to build up causal connections of the synchronicity and sequential relations in the same space-time (cf. figure 8b). The sequential relations $C_2 \hookrightarrow {}^*A_3 \hookrightarrow B_3$ is called causal-string episode, and synchronous relations $\updownarrow({}^*A_3B_4C_4)$ is called causal-fork timeframe. Both causal links are intercrossed by crossover correlative event A_3 (denoted as $^*A_3$) to compose a multiple time-series pattern. Non-causal correlations of A_2, A_4, B_1 and C_5 are the wildcards of the pattern presenting unstable causality. A nice property of this modeling is workable, meaningful, effective, and efficient in temporal relations mining.

There are several methods to illustrate the temporal pattern (cf. figure 9), in order to acquire target information, such as casual relationships of interval events. User-defined dimension of pattern is depended on the target pattern for seeking in application domain. Our target temporal pattern should be existed different attributes of time-series inside episode, so-called ***extra-relationship*** between time-series; for instance, episode $(B_3 \hookrightarrow A_1 \hookrightarrow D_3 \hookrightarrow C_2)$. A same attribute pattern could be occurred frequent in episode easily, so-called ***intra-relationship***; for instance, episode $(A_3 \hookrightarrow A_1 \hookrightarrow A_3 \hookrightarrow A_2)$, which is in standalone time-series pattern, not a multiple time-series pattern and not to be a target pattern. Hence, extra-relationships analysis should be mainly observed inside episode, not in timeframe. In other words, framework of timeframe is in unity space constraint, which cannot exist intra-relationship by different attributes in multiple time-series.

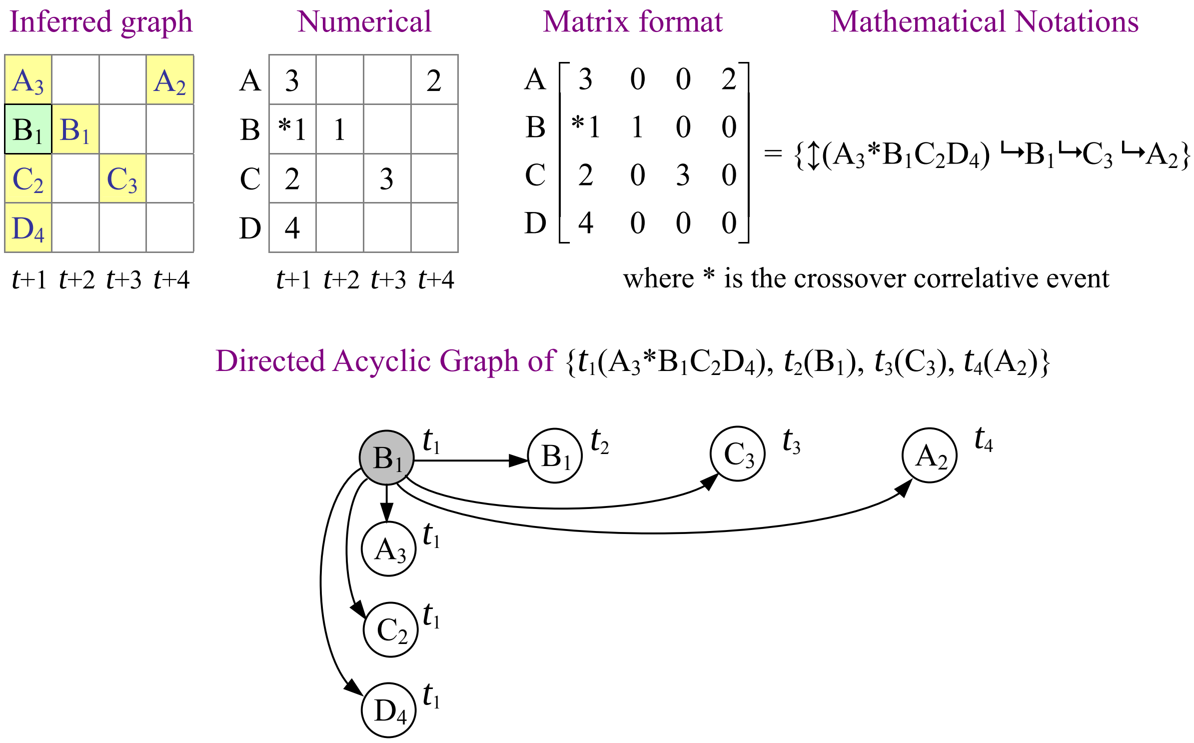


Figure 9. Examples of temporal pattern presentation

2.4 Space-Time Association Rule (STAR) and Relation Knowledge

A meta-rule-guided template proposed by Fu & Han [9] in the form of "$P_i \wedge \ldots \wedge P_{i+m} \Rightarrow Q_i \wedge \ldots \wedge Q_{i+n}$". Our multi-dimensional model could be employed this meta-rule-guided association rule format, but we revised the meta-rule template to Space-Time Association Rule (STAR), which is a new interval-relation knowledge representation to describe patterns characteristic of causal relationship. Let E be a superset of events. The STAR is an implication of the form "precursor(P) $\Rightarrow$ successor(Q)" or "cause $\Rightarrow$ effect" where $P \subseteq E$ and $Q \subseteq E$. By embedding space and time constraint into association rule $P \Rightarrow Q$, a pattern obtained by a time window (time-span) of length n in m time-series. P and Q are junctions of optimal certainties of causal pattern, such as the timeframe pattern of P is $P_i(t+k) \wedge \ldots \wedge P_{i+k}(t+k) \wedge \ldots \wedge P_{i+m}(t+k)$ and the episode pattern of Q is $Q_j(t) \wedge \ldots \wedge Q_{j+k}(t+k) \wedge \ldots \wedge Q_{j+n}(t+n)$, where $m \times n$ is dimension of pattern. Both patterns should be intercrossed by a same crossover correlative event E(t) at time t.

The optimal certainty of timeframe and episode quality measure via mutual information is $\sum I((X,\text{cause}),(Y,\text{effect}))_m$ and $\sum I((X,\text{cause}),(Y,\text{effect}))_n$ respectively (cf. section 3.1). If causal-string consists a fork opened toward the pass then $Q \Rightarrow P$, otherwise STAR is $P \Rightarrow Q$.

Expressing the template in the form: "If a crossover event E(t+k) at time t+k is occurred in timeframe P and episode Q, then the causal-fork P is associated to causal-string Q via a crossover correlative event at time t+k". And denoted as:

If $P_{i+k}(t+k) = Q_{j+k}(t+k)$ then

$$\begin{cases} P_i(t+k) \wedge \ldots \wedge P_{i+k}(t+k) \wedge \ldots \wedge P_{i+m}(t+k) \Rightarrow Q_j(t) \wedge \ldots \wedge Q_{j+k}(t+k) \wedge \ldots \wedge Q_{j+n}(t+n) \\ Q_j(t) \wedge \ldots \wedge Q_{j+k}(t+k) \wedge \ldots \wedge Q_{j+n}(t+n) \Rightarrow P_i(t+k) \wedge \ldots \wedge P_{i+k}(t+k) \wedge \ldots \wedge P_{i+m}(t+k) \end{cases}$$

where P is in synchronicity and Q is in sequential
P_i is timeframe $\{P_i \mid i \in \mathbb{N}\}$
Q_j is episode $\{Q_j \mid j \in \mathbb{N}\}$
$m \times n$ is dimension of pattern (time-series×time-span)
m is size of timeframe $\{m \mid m \in \mathbb{N}\}$
n is size of episode so-called time-span $\{n \mid n \in \mathbb{N}\}$
k is time-step of crossover correlative event $\{k \mid k \in \mathbb{N}\}$

Alternative presentation of mining knowledge obtained is causal relationship and temporal relations. Cause-effect relationships describing of the pattern about a phenomenon, causal inferences can be draw from such information. Deriving causal relationships from pattern, predicting the effects of actions and policies, evaluating explanations for observed events and scenarios. Interval-relation knowledge of temporal relations is in topological relations (before, after, overlap, during, simultaneously, between, etc.) and metric relations, i.e. those involving a distance in time or duration. Unlike the case of space, there are no directional relations, except for (before, and after). Hence, according to figure 9, temporal pattern could be obtained knowledge as followings:

- Space-time associated rule

  $*B_1(t+1) \wedge A_3(t+1) \wedge C_2(t+1) \wedge D_4(t+1) \Rightarrow *B_1(t+1) \wedge B_1(t+2) \wedge C_3(t+3) \wedge A_2(t+4)$
- Causal relationships

  $⌂*B_1A_3(1,1,0) \wedge ⌂*B_1C_2(1,1,0) \wedge ⌂*B_1D_4(1,1,0) \wedge ⌂*B_1B_1(1,1,1) \wedge ⌂*B_1C_3(1,1,2) \wedge ⌂*B_1A_2(1,1,3)$
- Causal-fork

  $⌂*B_1A_3(1,1,0) \wedge ⌂*B_1C_2(1,1,0) \wedge ⌂*B_1D_4(1,1,0)$
- Causal-string

  $⌂*B_1B_1(1,1,1) \wedge ⌂*B_1C_3(1,1,1) \wedge ⌂*C_3A_2(1,1,1)$
- Interval-relation knowledge

  $⌂*B_1A_3(2,1,0)$, $⌂*B_1C_2(2,1,0)$, $⌂*B_1D_4(2,1,0)$ are "Left Contain of interval_1 and interval_2"

  $⌂*B_1A_3(1,1,0)$, $⌂*B_1C_2(1,1,0)$, $⌂*B_1D_4(1,1,0)$ are "Match of interval_1 and interval_2"

  $⌂*B_1B_1(1,1,1)$, $⌂B_1C_3(1,1,1)$, $⌂C_3A_2(1,1,1)$ are "Meet of interval_1 and interval_2"

  $⌂*B_1C_3(1,1,2)$, $⌂*B_1A_2(1,1,3)$ are "Follow of interval_1 and interval_2"

3. Pattern Mining in Space-Time

3.1 Mutual information and Contingency table

We consider cause-effect relation of time-series based on probabilistic inference, the proposed method [6] could be uncovering hidden relationship between attributes through contingency table of mutual information. The fundamental quantities of information theory are entropy, relative entropy and mutual information, in which allows one to quantify the amount of information in a probability distribution or probability function. The entropy of a random variable X with a probability mass function $p(x)$ is defined by $H(X) = -\Sigma p(x)\log_2 p(x)$. We consider basic properties and axioms of Shannon's entropy. Let $n,m \in \mathbb{N}$ and $X = \{x_1, \ldots, x_n\}$ be a time-series (a finite set) with a probability distribution $p = (p_1, \ldots, p_n)$. Let $Y = \{y_1, \ldots, y_m\}$ be another time-series. The entropy $H(X)$ of a finite scheme (X,p) is defined by

$$H(X) = H(p) = H(p_1, \ldots, p_n) = -\sum_{j=1}^{n} p_j \log p_j .$$

The probability of (x_j,y_k) and the conditional probability of x_j given y_k are respectively denoted by $p(x_j,y_k)$ and $p(x_j|y_k) = p(x_j,y_k)/p(y_k)$. Then the conditional entropy $H(X|Y)$ of X given Y is defined by

$$H(X|Y) = -\sum_{y\in Y}\sum_{x\in X} p(y)p(x \mid y)\log p(x \mid y).$$

The quantity so-called mutual information $I(X,Y)$ between $(X,cause)$ and $(Y,effect)$ is

$$\sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}, \qquad \text{where} \quad H(X|Y) = -\sum_{x,y} p(x,y)\log p(x,y) \quad \text{is}$$

the entropy of the compound scheme $((X,cause),(Y,effect))$. And the relative entropy $H(cause|effect)$ is measure of how much '*effect*' differs from '*cause*', is given by

$$\sum_{j=1}^{n} cause_j(\log cause_j - \log effect_j) = \sum_{j=1}^{n} cause_j \log\frac{cause_j}{effect_j}.$$

We consider cause-effect relation in time domain is not symmetric, likewise relative entropy $H(cause|effect)$.

A contingency table is a cross-tabulation of data with frequency counts for two or more attributes, and using for verifying the independence of attributes. We use it to detect the cross-relationship between time-series or interval-relation. The χ^2 value is a particularly useful test for determining whether a pair of positions in a functional site is dependent or not. For instance, one wishes to test whether two symbolic random variables X and Y are dependent. One is asked to make this determination solely from a given dataset of time-series $S_{XY} = \{(x,y)\}$ of occurrences of their joint values, drawn from the unknown joint probability function $Pr(X,Y)$ on these random variables. The proceeds as follows:

(1) Corresponding to multiple time-series dataset $\{x_1, \ldots, x_n\}\times\{y_1, \ldots, y_m\}$, S_{XY} constructs contingency table C, where C_{xy} stores the number of occurrences of the pair (x,y) in S_{XY}.

(2) Computes E_{xy}, the expected number of occurrences of the pair (x,y) under the hypothesis that X and Y are independent, as $E_{xy} = E_x \times E_y \div |S_{XY}|$, where E_x is the sum of row x in the C, E_y is the sum of column y in this C, and $|S_{XY}|$ the size of the dataset S_{XY}.

(3) Next, computes the χ^2 value, $\chi^2 = \sum_{X=x,Y=y} \frac{(C_{xy} - E_{xy})^2}{E_{xy}}$.

(4) And then, computes the probability $Pr_{\chi 2}$ of this χ^2 value arising under the independence assumption, from the χ^2 probability distribution with degrees of freedom set equal to $(n-1)\times(m-1)$.

(5) Finally, if this probability $Pr_{\chi 2}$ turns out to be sufficiently close to zero, may reject the hypothesis that the random variables X and Y are independent, with confidence $1-Pr_{\chi 2}$.

3.2 Certainty of statistical significance

Chan et al. [6] found dependencies between several pairs of attribute positions in $Attr_{jp}$ and $Attr_{i(p+r)}$ (skip position r) using χ^2 test, in order to obtained certainty W_{ij} of the rule. In this paper, we follow this modeling to evaluate the certainties of causal-fork or causal-string using the same parameters, mutual information and conditional rule format. The example is taken from the paper [6] as following:

Step 1: Counting frequency of occurrences of the pair "Cause ↳Effect"

4-wheel ↘	3-wheel ↘	4-wheel ↘	2-wheel ↘	2-wheel	…	3-wheel	4-wheel
3-window	2-window	3-window	1-window	3-window	…	1-window	1-window

4-wheel ↳2-window (1st locomotive ↳2nd locomotive) Total frequency in sequence is 19
3-wheel ↳3-window (2nd locomotive ↳3rd locomotive) Total frequency in sequence is 3
4-wheel ↳1-window (3rd locomotive ↳4th locomotive) Total frequency in sequence is 12
2-wheel ↳3-window (4th locomotive ↳5th locomotive) Total frequency in sequence is 7
⋮ ⋮ ⋮

Step 2: Contingency table for number of windows and number of wheels with (χ^2-measure)

	1-window	2-window	3-window	Total
2-wheel	3, (χ^2 = 6.83)	4, (χ^2 = 5.79)	7, (χ^2 = 2.90)	14
3-wheel	7, (χ^2 = 4.17)	1, (χ^2 = 4.55)	3, (χ^2 = 2.28)	11
4-wheel	12, (χ^2 = 12.52)	19, (χ^2 = 13.66)	2, (χ^2 = 6.83)	33
Total	22	24	12	58

Step3: Weight of evidence calculations based on mutual information (e.g. 4-wheel ↳2-window)

W(number of windows = Two /number of windows ≠ Two | number of wheels = Four)

$$= \log_2 \frac{\Pr(\text{number of wheels} = \text{Four} \mid \text{number of windows} = \text{Two})}{\Pr(\text{number of wheels} = \text{Four} \mid \text{number of windows} \neq \text{Two})}$$

$$= \log_2 \frac{\frac{19}{24}}{\frac{33-19}{58-24}} = \frac{\log 1.92}{\log 2} = 0.94$$

Step 4: Construction of rules with certainty (e.g. 4-wheel ↪2-window)

Rule format: If (*condition*) then (*conclusion*) with certainty (*weight*)

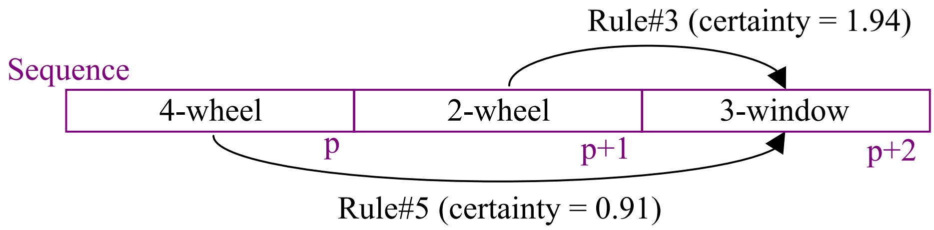
Rule#1: 4-wheel(p) ↪2-window(p+1), Certainty = 0.94

If a locomotive has four wheels when it is with certainty 0.94 that the locomotive located at one position later in the sequence has two windows.

Step 5: Rules for all relevant values (Bounded in 5% of Standard normal deviate = 1.96)

Rule#1: 4-wheel(p) ↪2-window(p+1), certainty = 0.94
Rule#2: 3-wheel(p) ↪2-window(p+1), certainty = –2.82
Rule#3: 2-wheel(p) ↪3-window(p+1), certainty = 1.94
Rule#4: 4-wheel(p) ↪3-window(p+1), certainty = –2.02
Rule#5: 4-wheel(p) ↪3-window(p+2), certainty = 0.91
⋮ ⋮ ⋮
Rule#12: 2-wheel(p) ↪3-window(p+1), certainty = –1.42

Step 6: Optimal certainty pruning (sequential pattern)



The pattern is determined by maximum of certainties = 0.91 + 1.94 = 1.85

Hence, the significance sequential pattern is {4-wheel ↪2-wheel ↪3-window}.

Applying above contingency table, mutual information and conditional rules could be determined optimal certainty of causal-fork or causal-string. The possible cases occurrences of timeframe or episode are elaborate in next section 3.4. Chan et al. [6] model, the weight of evidence is given by the weight of the rules in the same set. The top-N-certainty measure of temporal pattern is based on the range of values in same scope of quantities. But our model needs to intercross two different sets of certainty, one is in synchronicity, another one is in sequential. These two sets of certainty should be normalized into same range of values to remove the relative difference, in order to determine the optimal top-N-certainty of temporal pattern. The new range of certainties is [–1,1] after unique interval normalization.

$$\text{New certainty} = (\text{MAX} - \text{MIN})\frac{\text{Certainty} - \max_x}{\max_x - \min_x} + \text{MAX}$$

where $\min_x$, $\max_x$ are the minimum and maximum values of original certainty
MAX, MIN are the values of the target range, that is [–1,1].

3.3 Target space-time of temporal pattern

We consider 4×4 target space-time (time-series×time-span), because useful short-acting knowledge relationship of events usually are pre-occur, post-occur and current-event including a wildcard existed, so totally to be four events in an episode, likewise reasoning in timeframe. It is well known that the number of time-series is limited by computing constraints, which could be solved by meta-code transformation. Four time-series is selected for synchronicity analysis based on relationship seeking in real life dataset, normally user-defined by dataset nature. Limiting quantity could be avoided too much noise created among episodes to become more factualness. The meta-rule-template of 4×4 space-time as following:

$$\text{If } P_i(t+k) = Q_j(t+k) \text{ then}$$

$$\begin{cases} P_A(t+k) \wedge P_B(t+k) \wedge P_C(t+k) \wedge P_D(t+k) \Rightarrow Q_j(t) \wedge Q_j(t+1) \wedge Q_j(t+2) \wedge Q_j(t+3) \\ Q_j(t) \wedge Q_j(t+1) \wedge Q_j(t+2) \wedge Q_j(t+3) \Rightarrow P_A(t+k) \wedge P_B(t+k) \wedge P_C(t+k) \wedge P_D(t+k) \end{cases}$$

where P_i is timeframe $\{ P_i \mid i = A, B, C, D \}$
Q_j is episode $\{ Q_j \mid j = A, B, C, D \}$
k is time-step $\{ k \mid k = 1, 2, 3, 4 \}$

If the cross-relationship of an episode only contains intra-relationship in pattern, it becomes a loss synchronicity pattern in multiple time-series. This is not a good temporal pattern for cross-relationship mining, even though synchronicity appeared in timeframe. Therefore, obtained higher Top-N-certainty of extra-relationships among time-series is our target temporal pattern. According to 4×4 temporal pattern, there are four meaningful pattern formats with wildcards would be occurred (cf. figure 10). This 4×4 space-time format would be applied throughout of this paper.

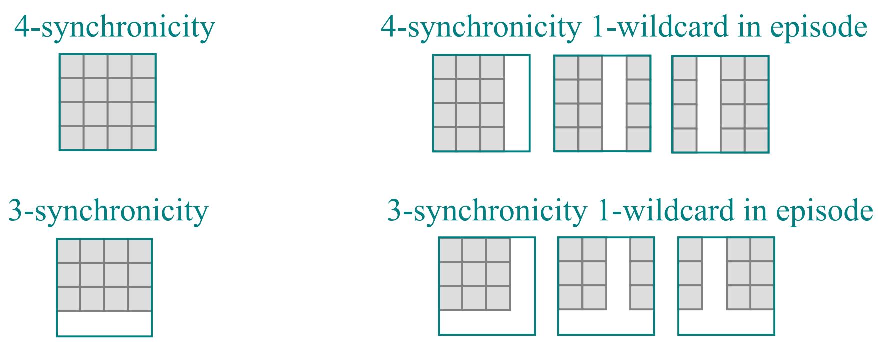


Figure 10. Space of extra-relationship with wildcards in temporal pattern

There are two essential causal relationship components to build up a temporal causal-effect pattern, one is synchronous timeframe pattern another is sequential episode pattern. Timeframes and episodes fulfil whole pattern to become a full pattern, which are called global timeframe pattern (cf. figure 11a) and global episode pattern (cf. figure 12a) respectively. These optimal patterns obtained by maximal certainties (such as $W_{AB}, W_{AC}, W_{PQ}, W_{RS}$, …) of causal-fork and causal-string. If wildcards "□" inserted in Causal-fork (timeframe) or causal-string (episode), they are named as local maximal (cf. figure 11b & 12b). Causal-fork expanded synchronicity in space, and causal-string expanded sequential in time direction. Both cause-effect relationships should be merged together by a common crossover correlative event (circles shaded in figures) to form a temporal pattern (cf. figure 10).

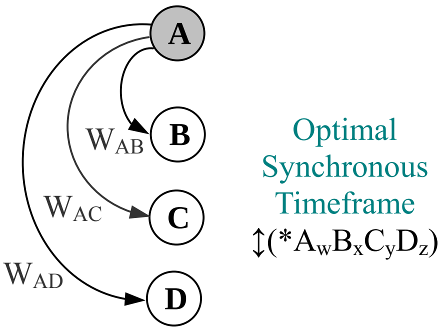


Figure 11a. Global maximal timeframe

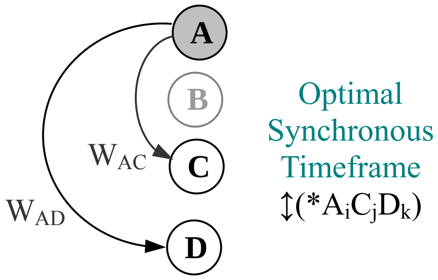


Figure 11b. Local maximal timeframe

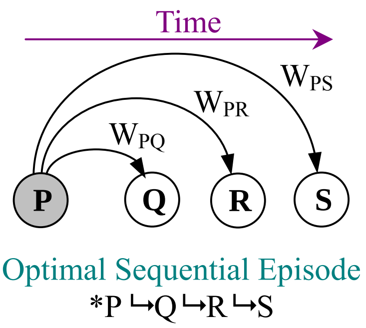


Figure 12a. Global maximal episode

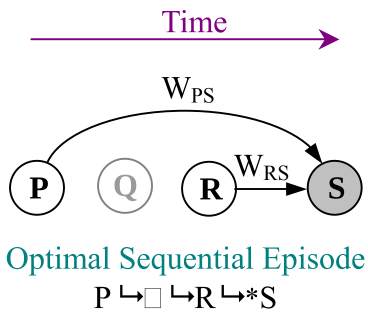


Figure 12b. Local maximal episode

3.4 Complexity analysis of components

Two causal relationship components named causal-fork and causal-string, which are grouped by meta-codes to form space-frame and time-direction contiguous subsequence respectively. Calculation of the time-complexity of this algorithm requires counting the number of times of compute operation is repeated. Applying combinatorial reasoning for complexity counting of possibilistic causal-fork and causal-string, in order to analysis performance of algorithms or point out the problems of space-complexity during computing. In this section, 4×4 dimensions of target space-time is used in following examples. The notation example as below:

e.g.
- If $A^\tau = \{A_1, A_2\}$, $B^\tau = \{B_1, B_2, B_3, B_4\}$, $C^\tau = \{C_1, C_2, C_3\}$, $D^\tau = \{D_1, D_2, D_3, D_4, D_5, D_6\}$ then $\{ A^\tau, B^\tau, C^\tau, D^\tau \mid Code^\tau = (2,4,3,6) \}$.
- Each of n time-series followed tokens denoted as $Code^\tau_n$
- If $\{ n \mid n = (1,2,3,4)\}$ then $\{ Code^\tau_1, Code^\tau_2, Code^\tau_3, Code^\tau_4 \mid Code^\tau = (2,4,3,6) \}$.
- Hence, $\sum Code^\tau_n = 2 + 4 + 3 + 6$,
- If $i = 2$ then $k_1k_2 = (12,13,14,23,24,34)$;
  where $\{ k_1 \ldots k_i \mid i \in \{1 \le i \le n\}, k_i \in \{1,2,3,4\}$ and $k_i < k_{i+1} \}$.

- Hence, $\prod Code^{\tau}(k_1k_2)$ = 2×4, 2×3, 2×6, 4×3, 4×6, 3×6;
  where $\{\, Code^{\tau}(k_1 \ldots k_i) = Code^{\tau}k_1, \ldots, Code^{\tau}k_i \mid Code^{\tau}k_i \in \{2,4,3,6\} \,\}$.
- Finally, $max\{\prod Code^{\tau}(k_1k_2)\}$ = 4×6.

3.4.1 Possibility of Causal-fork

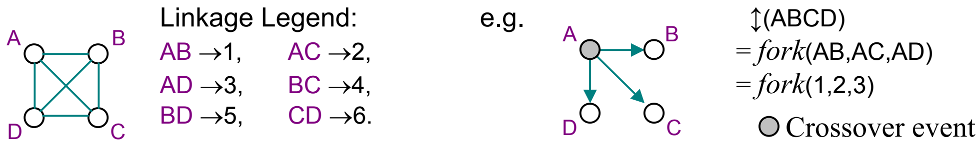


Figure 13. Correlative-linkages of causal-fork

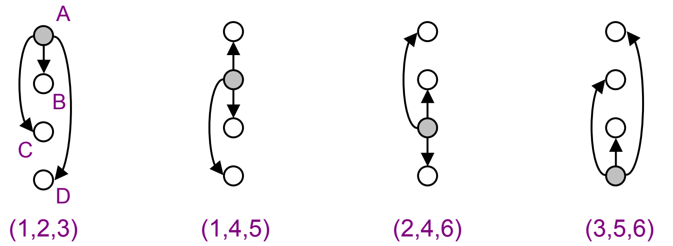


Figure 14. $fork^{\#0}$ = 4 (Possible causal-fork of 4 synchronous events)

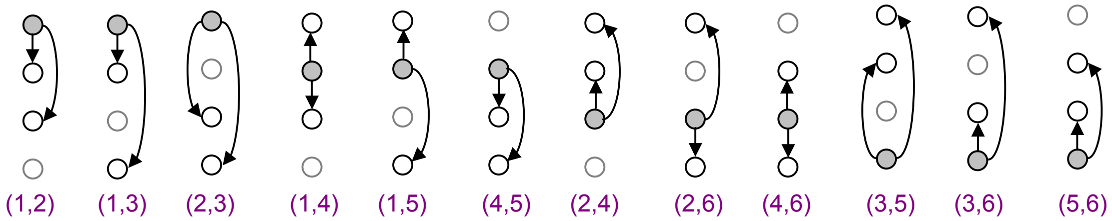


Figure 15. $fork^{\#1}$ = 12 (Possible causal-fork of 3 synchronous events)

For evaluating the optimal certainty in all possibility cases should be counted, and including wildcard options in a causal-fork. Simple linkages of fork structure layout by reference numbers as shown in figure 13. The total numbers of possibility cases occurred of causal-fork named $fork^{\#}$, which contains 1 wildcard so-called $fork^{\#1}$ (cf. Figure 15), and $fork^{\#0}$ (cf. Figure 14).

- $$fork^{\#w} = n \times \binom{p}{r} = n \times \frac{p!}{r!(p-r)!}$$
  where $p = n - 1$, $r = n - w - 1$
  n is number of time-series
  w is number of wildcard
- $$fork^{\#} = fork^{\#0} + fork^{\#1} + \ldots + fork^{\#(n\text{-}2)} = n(2^{n-1}-1)$$

Possibility of timeframe for implementing in n time-series with meta-codes $Code^{\tau}_{n}$ will generated maximum number of timeframes named $timeframe^{\#}$, which is equal to maximum number of $timeframe^{\#0} + timeframe^{\#1} + \ldots + timeframe^{\#(n\text{-}2)}$ corresponding to $fork^{\#0} + \ldots + fork^{\#(n\text{-}2)}$.

e.g. If n = 4, $Code^{\tau} = (4,4,4,4)$, then $fork^{\#} = 4+12+12 = 28$

$$timeframe^{\#0} = fork^{\#0} \times \prod Code^{\tau}(k_1k_2k_3k_4) = 4 \times (4\times4\times4\times4) = 1024$$

$$timeframe^{\#1} = fork^{\#1} \times max\{\prod Code^{\tau}(k_1k_2k_3)\} = 12 \times (4\times4\times4) = 768$$

$$timeframe^{\#2} = fork^{\#2} \times max\{\prod Code^{\tau}(k_1k_2)\} = 12 \times (4\times4) = 192$$

N.B. $fork^{\#3}$ cannot be occurred.

$$timeframe^{\#} = 1024 + 768 + 192 = 1984$$

3.4.2 Possibility of Causal-string

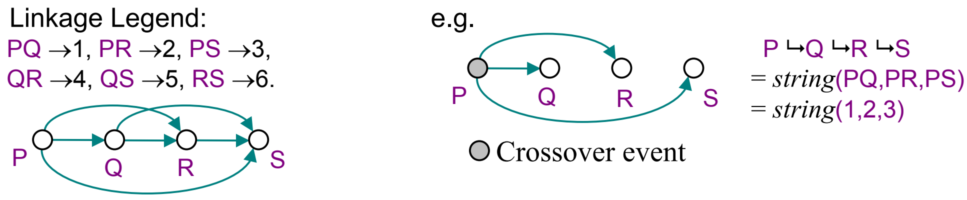


Figure 16. Correlative-linkages of causal-string

Figure 17. $string^{\#0}$ = 4 (Possible causal-string of 4 sequential events)

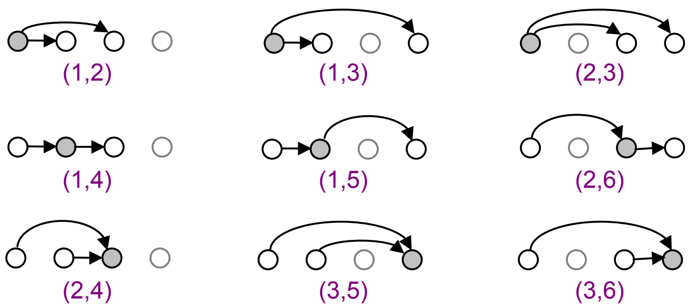


Figure 18. $string^{\#1}$ = 9 (Possible causal-string of 3 sequential events)

Possibility cases of causal-string ought to have complexity analysis, because it would be generated massive causal-strings during certainties finding. According to figure 16 notations, the total numbers of possibility cases occurred of causal-string named $string^{\#}$, which contains one wildcard so-called $string^{\#1}$ (cf. Figure 18), likewise in $string^{\#0}$ (cf. Figure 17).

- $$string^{\#w} = (n-w) \times \binom{n-1}{w} = (n-w) \times \frac{(n-1)!}{w!(n-w-1)!},$$

  where n is number of time-series
  w is number of wildcard

- $$string^{\#} = string^{\#0} + string^{\#1} + \ldots + string^{\#(n-2)} = \sum_{w=0}^{n-2} (n-w) \times \binom{n-1}{w}$$

Similarly, possibility of episode for implementing in n time-series with meta-codes $Code^{\tau}_{n}$ will generated maximum number of sequential episodes named $episode^{\#}$, which is equal to maximum number of $episode^{\#0} + episode^{\#1} + \ldots + episode^{\#(n-2)}$ corresponding to $string^{\#0} + \ldots + string^{\#(n-2)}$.

e.g. If n = 4, $Code^{\tau} = (4,4,4,4)$, then $string^{\#} = 4 + 9 + 6 = 19$

$episode^{\#0} = string^{\#0} \times (\sum Code^{\tau}_{n})^{n} = 4 \times (4+4+4+4)^{4} = 262144$

$episode^{\#1} = string^{\#1} \times (\sum Code^{\tau}_{n})^{n-1} = 9 \times (4+4+4+4)^{3} = 36864$

$episode^{\#2} = string^{\#2} \times (\sum Code^{\tau}_{n})^{n-2} = 6 \times (4+4+4+4)^{2} = 1536$

N.B. $string^{\#3}$ cannot be occurred.

$episode^{\#} = 262144 + 36864 + 1536 = 300544$

Due to $episode^{\#} \gg timeframe^{\#}$, we can draw the conclusion that computation complexity should be focus on measuring number of time-series rather than counting meta-codes in standalone time-series. Hence, secondary meta-code is better than primary meta-code during computing implementation (cf. Section 5.3).

3.5 Pruning Algorithm

We introduce a Shorten Search Region (SSR) algorithm to obtain an optimal temporal pattern of timeframe and episode intercrossing with maximum weight (certainty). All algorithms are coding in C++ programs, legend of symbols is given in Table 3, and pseudo-code in Algorithm 1. Two buffers are reserved for timeframes and episodes, and the lengths of episode region corresponding to search size. Intercrossing search started from timeframe buffer head, crossover event is matched by increasing weight via descending number of timeframe. The episode search region will be rebuilt every iteration until this search region is zero. The SSR pruning is divided into four modules, which are Weight_To_Region, Rebuild_Search_Region, Crossover_Event_Match and Shorten_Search_Region procedures, pseudo-codes in Algorithm 1.1, 1.2, 1.3 & 1.4 respectively.

Symbol	Meaning
#	Number of / Position number of
*⌂	Relationship of crossover correlative event
$^{search}\#region$	Position number of search region
$^{match}\#syn$	Position number of synchronous matched pattern
$^{memory}\#syn$	Position number of synchronous pattern in memory
$^{max}\#syn$	Position number of timeframe in maximum weight
$^{max}weight$	Maximum weight of pattern
$^{total\#}E^{4}_{4}\updownarrow$	Total number of synchronous event sequences
$E^{4}_{4}\hookrightarrow$	4 meta-codes in sequential episode of 4 series
$E^{1}_{4}\updownarrow$	1 meta-code in synchronous timeframe of 4 series
$^{max}synE^{1}_{4}$	Maximum weight of synchronous events
$SynE^{1}_{4}$*⌂#1	First number of crossover event in timeframe

Table 3. Legend of SSR algorithm

Algorithm 1.1 Convert weight to region for revising the length of search region

```
Weight_To_Region( ^search#region, #next_syn, ^match#syn, ^match#seq )
{ for next_seq ∈ ^max#seq do
    if (weight(#next_syn) – weight(^match#syn) ≤ weight(#next_seq) – weight(^search#region))
    { ^match#seq ← ^search#region + #next_seq;
      next_seq ← ^max#seq;
    }
}
```

Algorithm 1.2 Rebuild the search region for crossover event matching

```
Rebuild_Search_Region( ^search#region, ^match#syn )
{ for next_syn ∈ synE^1_4*⌂(^memory#syn) do
    if (^max synE^1_4*⌂(^max#syn) == seqE^1_4*⌂(#next_syn))
    { Weight_To_Region( ^search#region, #next_syn, ^match#syn, ^match#seq );
      ^match#syn ← #next_syn;
    }
}
```

Algorithm 1.3 Crossover event intercross with timeframe and episode

```
Crossover_Event_Match( ^max weight, ^match#syn, ^match#seq )
{ for next_seq ∈ ^match#seq do
    if (^max synE^1_4*⌂(^match#syn) == seqE^1_4*⌂(#next_seq))
    { if ((weight(^match#syn) + weight(#next_seq)) ≥ ^max weight)
      { ^max weight ← weight(^max#syn) + weight(^max#seq);
        ^max#seq ← #next_seq;
        ^max#syn ← ^match#syn;
      }
      next_seq ← ^match#seq;
    }
}
```

Algorithm 1.4 Iteration for crossover event intercrossing

```
Shorten_Search_Region( syn_wt, synE^1_4*⌂, E^4_4↕, seq_wt, seqE^1_4*⌂, E^4_4↳)
{ ^max synE^1_4*⌂(^max#syn) ← synE^1_4*⌂#1;
  ^max seqE^1_4*⌂(^max#seq) ← seqE^1_4*⌂#1;
  ^search#region ← ^total# E^4_4↳;
  ^memory#syn ← 1;
  ^match#syn ← 1;
  while ((^memory#syn ≤ ^total# E^1_4↕) ∨ (^match#syn ≥ ^total# E^4_4↕)) do
  { Rebuild_Search_Region( ^search#region, ^match#syn );
    Crossover_Event_Match( ^max weight, ^match#syn, ^match#seq );
    ^match#syn++ ;
  }
  Temporal Pattern ← { E^4_4↕⌂(^max#syn)} ∩ { E^4_4↳⌂(^max#seq)}
}
```

Algorithm 1: Pseudo-code of SSR Algorithm

4. MMS Model Validation

4.1 Synthetic Sequences

For the purpose of evaluation of the performance of MMSM, and detect the relations threshold of suggested model. Synthetically sequences are generated randomly underlying linear congruential randomization (uniform distribution). The generated dataset behavior as a dice throwing: $\{ \text{dice}^i_n \mid \text{dice}^i \in \{1 \le \text{dice}^i \le 6\} \text{ and } \text{dice}_1, \text{dice}_2, \ldots, \text{dice}_n \mid n \in \mathbb{N} \}$.

e.g. Throw 3 times of 4 dices (dice A, B, C and D, with 6-sided block of each);

1st throw is (A_1 B_6 C_3 D_2), 2nd throw is (A_3 B_4 C_5 D_1) and 3rd throw is (A_6 B_2 C_4 D_3);

The multiple-sequence is $\{ t_1(A_1B_6C_3D_2), t_2(A_3B_4C_5D_1), t_3(A_6B_2C_4D_3) \}$.

In Algorithm 2, four Linear Congruential Randomization (LCR) sequences are generated in text file format. Each sequence consists of four distinct events and 1000 occurrences of events. The synthetic data numbers are in same attribute, uniform distribution, and sequence order in randomize. The aims of this synthetic data testing are determine the acceptability of noise level and loss extra-relationship percentage for real-life implementation.

```
Algorithm: LCR_Dataset_Generator
Output:  Four text files contains { x^i_n | x^i ∈ {1 ≤ x^i ≤ 4} and x^i_1, x^i_2, …, x^i_n | n = 1000}
void
{ for n ∈ nos_of_file do
   {  for i ∈ file_length / 4 do
            ∀x^i ← { x^i_1, x^i_2, x^i_3, x^i_4 | x^i = (1,2,3,4) };
      Generate 1000 LCR random number ∀x ∈ {0 ≤ x ≤ 32767};
      position[x^i_n] ← x;    sort[x^i_n] ← x;        // Created duplicate code list for sorting
      Bubble sorting of random number sort[x^i_n];
      for j ∈ file_length do
         for k ∈ file_length do
            if sort[j] = position[k]
               {  sequence[j] ← x^i_k;   // Insert code into sorted list follow random sequence
                  x^i_k ← 0;             // Avoid duplicate in sorted list by clearing memory
                  k ← file_length;       // Break the looping
               }
      fileOutput "FileName" ← sequence[x^i_n];
   }
}
```

Algorithm 2: Pseudo-code of LCR generator algorithm

4.2 Loss Extra-relationship Analysis

According to synthetic sequence, the percentage of noise or outliners (noise level) is a parameter, which should be added into sequences, in order to test the performance of noise level in model. Outliners are distributed uniformly at random throughout in sequence. For our extra-relationship detection, noise level is set at 25% (3 extra-relationship events in 4 events) and

50% (2 extra-relationship events in 4 events). During the target patterns mining process, various correlation thresholds result used different synthetic dataset, which contains corresponding different noise level in percentage. The higher noise level, the fewer extra-relationship are occurred. Our target goal is multiple time-series pattern, which ideally will be a correlation threshold level that gives 100% of extra-relationship between time-series. But normally correlation threshold more than 50% is acceptable, in case it can be detect the non-causal relationship time-series from pattern.

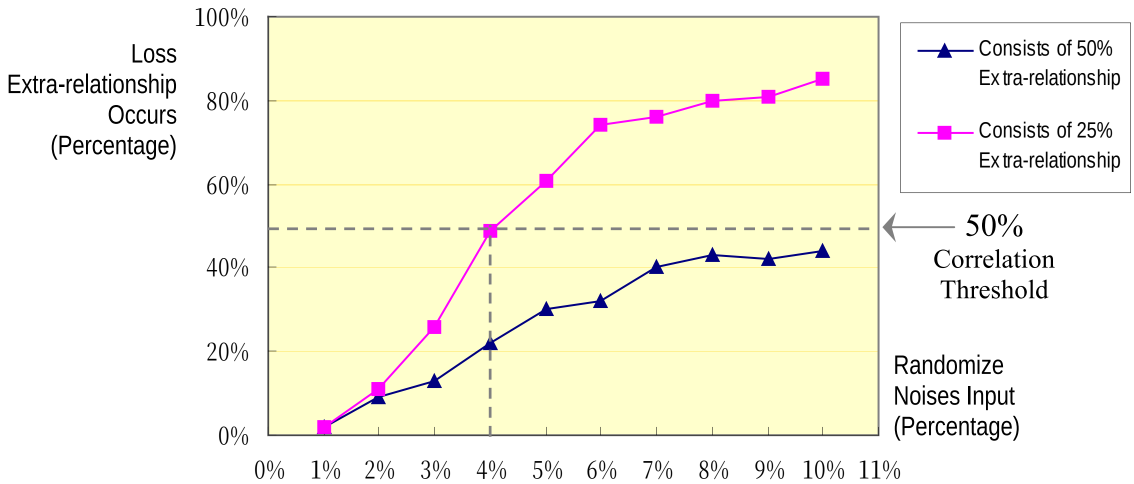


Figure 19. Loss of extra-relationship w.r.t. Noise percentage

According to the experiment result in Figure 19, 50% extra-relationship curve inside upper region and 25% extra-relationship curve inside lower region. The threshold of intra-relationship to be 50% is selected, in order to distinguish whether target pattern in multiple time-series or not. If threshold boundary at 50%, then loss extra-relationship will not happened below 4% of noise existed during mining. Therefore, it is a stable mechanism of discovering multiple time-series patterns under 4% noise level. If the outcome of pattern contains less than 50% of extra-relationship, it can be apply a rule to ignore intra-relationship conditions in episode for extra-relationship recovering, in order to obtain target pattern. The rule of relation-seeking pattern as following:

If outcome of temporal pattern contains more than 50% of intra-relationship in episode,
then let weight of this episode be 0, which is more than 50% of intra-relationship,
and timeframe intercross another optimal episode.

5. Case Study: Financial Time-series Implementation

In this paper, we have conducted some experiments on real data of financial time-series to determine temporal pattern. All the indicators from internet [38] are selected from year 2003 to year 2006, and shifting their indexes corresponding to business-day. The data are the closed market price in a trading day. If there is no trading activity for a particular indicator, we use the last trading price as its price. In this way, all the indicators are corresponding to the same synchronicity at every time-step in multiple time-series. Using MMSM, we can detect extra-relationship among financial time-series, which are the state time-series.

5.1 Primary Meta-code Modeling

Primary meta-code is identical attribute and simple RLR transformation, normally higher contribution in two meta-codes. RLR should be excluded extra-relationship of time-series, which may be analysed in mining. Here, the financial indicators are selected from [38] for mining patterns of stock market behavior. The following represents the financial markets in the Hong Kong stock market:

(1) HSI: The Hang Seng Index (HSI), a market capitalisation-weighted index (shares outstanding multiplied by stock price), consists of 33 constituent stocks. The influence of each stock on index's performance is directly proportional to its relative market value. HSI has proven to be an effective tool to track the overall performance of the Hong Kong equity market.

(2) HSI Futures: The HSI is commonly used as the base index for variety of derivatives products, which provide investors with a set of effective instruments to manage portfolio risk and to capture index arbitrage opportunities.

(3) Turnover: The trading volume of the Hong Kong stock market.

(4) Capitalisation: The aggregate market capitalisation of 33 stocks accounts for about 70% of the total market capitalisation of the Hong Kong stock market.

The following A, B, C and D are defined as "Code" of time-series, and F_1, F_2, F_3, H_1, … , T_3 are defined as "Meta-code" of time-series. The primary meta-code transformation as below:

$A \Leftarrow F(t) - F(t\text{-}1)$ = Today HSI Futures value – Yesterday HSI Futures value

$F_1 \Leftarrow 0<A$ $\quad F_2 \Leftarrow A<0$ $\quad F_3 \Leftarrow \neg(F_1 \vee F_2)$ $\quad$ 'F_3' *seldom happened in real stock market*

$B \Leftarrow H(t) - H(t\text{-}1)$ = Today HSI value – Yesterday HSI value

$H_1 \Leftarrow 0<B$ $\quad H_2 \Leftarrow B<0$ $\quad H_3 \Leftarrow \neg(H_1 \vee H_2)$ $\quad$ 'H_3' *does not happen in real stock market*

$D \Leftarrow H(t) - F(t)$ = HSI Futures value – Today HSI value

$I_1 \Leftarrow 0<D$ $\quad I_2 \Leftarrow D<0$ $\quad I_3 \Leftarrow \neg(I_1 \vee I_2)$ $\quad$ 'I_3' *does not happen in real stock market*

$C \Leftarrow$ (Turnover / Capitalisation) 100% $\qquad P = C(t) - C(t\text{-}1)$

$T_1 \Leftarrow 0<P$ $\quad T_2 \Leftarrow P<0$ $\quad T_3 \Leftarrow \neg(T_1 \vee T_2)$ $\quad$ 'T_3' *does not happen in real stock market*

The categories of meta-code are in three different codes, but as the results are in binary. Because meta-code F_3, H_3, I_3 and T_3 are almost did not occurred in real stock market. The RLRs are applied, such as F-code and H-code for up-down analysis of HSI futures and HSI respectively, I-code for monthly prediction analysis of HSI, T-code for relation analysis of investors' actions, and C-code is normalized turnover values according to capitalisation in percentage. Our target temporal pattern is in 4×4 dimensions, and mining optimal top-5-certainty target patterns for selection of extra-relationship attribute pattern. The outcomes may be presented in pattern matrix format, association rule and relations knowledge as following:

$$
\text{Certainty} = \quad
\begin{matrix} \\ F \\ H \\ I \\ T \end{matrix}
\begin{matrix} 0.79154 \\ \begin{bmatrix} *3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix} \end{matrix}
\begin{matrix} 0.569945 \\ \begin{bmatrix} 3 & 0 & 3 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & *2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}
\begin{matrix} 0.568985 \\ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & *1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}
\begin{matrix} 0.449151 \\ \begin{bmatrix} 3 & 3 & 0 & 3 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & *2 \end{bmatrix} \end{matrix}
\begin{matrix} 0.411198 \\ \begin{bmatrix} *1 & 0 & 3 & 3 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}
$$

0.79154	0.569945	0.568985	0.449151	0.411198
	75% Loss	100% Loss	50% Loss	50% Loss
	Extra-relationship	Extra-relationship	Extra-relationship	Extra-relationship
Target Pattern	*Non-target*	*Non-target*	*Target Pattern*	*Target Pattern*

- **Selected target pattern:**

	t+1	t+2	t+3	t+4
HSI Futures	*3			1
HSI				
HSI & Futures	2		2	
Turnover	2			

$$
\begin{matrix} F \\ H \\ I \\ T \end{matrix}
\begin{bmatrix} *3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix}
= \{\, t_1(*F_3I_2T_2),\ t_3(I_2),\ t_4(F_1) \,\}
$$

where * is the crossover correlative event

The target pattern states that —— Two days ago, the Turnover was down, HSI Futures was unchanged, and foresaw the HSI will be down in the future. Today the investors foresee HSI will be going down. But HSI Futures values are going up tomorrow.

- **Association rule of target pattern**: $*F3(t) \wedge I2(t+2) \wedge F1(t+3) \Rightarrow *F3(t) \wedge I2(t) \wedge T2(t)$

- **Causal relationships**: ⌂*F3I2(1,1,0) ∧ ⌂*F3T2(1,1,0) ∧ ⌂*F3I2(1,1,2) ∧ ⌂*F3F1(1,1,3)

Using the same algorithm to mine patterns with different dataset. By observing below short-term temporal patterns, which indicated the stock market behavior was changed every year. Obviously, the meaningful interval-relation knowledge is easier obtained in intra-relationship pattern, such as P_{2004}, and causal relationship of P_{2004}, P_{2005} and P_{2006} is concentrated on HSI trends foreseeing, but P_{2003} is turnover downward status. Those can be observed from intra-relationship in episodes.

- In year 2003, the temporal pattern is $P_{2003} = \{\, t_1(*F_3I_1T_2),\ t_2(T_2),\ t_3(T_2) \,\}$
- In year 2004, the temporal pattern is $P_{2004} = \{\, t_1(I_1),\ t_2(I_1),\ t_3(*I_1F_1H_1),\ t_4(I_1) \,\}$
- In year 2005, the temporal pattern is $P_{2005} = \{\, t_1(*F_2H_2I_2),\ t_2(F_3),\ t_3(I_1),\ t_4(I_2) \,\}$
- In year 2006, the temporal pattern is $P_{2006} = \{\, t_1(I_2),\ t_3(*I_2F_2H_2),\ t_4(I_2) \,\}$

5.2 Secondary Meta-code example:

If we want to know the extra-relationship between Hong Kong market and foreign market behaviour, the secondary meta-code transformation may be used. The secondary meta-code can be absorbed more time-series without increasing timeframe size. But complex

RLR and combining similar attributes would be used. There is an example as below:

(1) HKD Rate: Hong Kong Dollar exchange rate.

(2) JPY Rate: Japan Yen exchange rate.

(3) EUR Rate: Euro exchange rate.

(4) Yield: Treasury bonds issued by the government of the United States. They are expired on 1-month, 10-year, … etc., and Hong Kong interest rate affected by them.

(5) HIS, Turnover, … : As same as in section 5.1.

$E \Leftarrow$ HKD-EUR Rate Up $\quad J \Leftarrow$ HKD-JPY Rate Up

$F_1 \Leftarrow E \wedge J \quad F_2 \Leftarrow E \wedge \neg J \quad F_3 \Leftarrow \neg E \wedge J \quad F_4 \Leftarrow \neg E \wedge \neg J = \neg(E \vee J)$

$t \Leftarrow$ Today HSI value – Yesterday HSI value $\quad f \Leftarrow$ HSI Futures value – Today HSI value

U = Up $\quad D$ = Down $\quad E \Leftarrow \neg(U \vee D)$ ‘E’ *does not happen in real financial market*

$I_1 \Leftarrow U_t \wedge U_f \quad I_2 \Leftarrow U_t \wedge D_f \quad I_3 \Leftarrow D_t \wedge U_f \quad I_4 \Leftarrow D_t \wedge D_f$

$C \Leftarrow$ (Turnover / Capitalisation) 100% $\quad P \Leftarrow C(t) - C(t\text{-}1) \quad A = 0.3\%$

$T_1 \Leftarrow (C<A) \wedge (0<P) \quad T_2 \Leftarrow (A \le C) \wedge (0<P) \quad T_3 \Leftarrow (C<A) \wedge (P \le 0) \quad T_4 \Leftarrow (A \le C) \wedge (P \le 0)$

m = 1-month Yield $\quad$ n = 10-year Yield $\quad M \Leftarrow m(t) - m(t\text{-}1) \quad N \Leftarrow n(t) - n(t\text{-}1)$

U = Up $\quad D$ = Down $\quad E \Leftarrow \neg(U \vee D)$

$Y_1 \Leftarrow U_M \wedge U_N \quad Y_2 \Leftarrow U_M \wedge \neg U_N \quad Y_3 \Leftarrow D_M \wedge U_N \quad Y_4 \Leftarrow D_M \wedge \neg U_N \quad Y_5 \Leftarrow E_M$

Outcome pattern $P_{SI} = \{t_1(T_4), t_2(T_2), t_3(*T_4F_3I_3), t_4(T_4)\}$ which is a loss extra-relationship temporal pattern. Therefore, the applied the relation-seeking rule to obtain the target pattern of extra-relationship is $P_{SE} = \{ t_1(I_1), t_2(T_2), t_3(*T_4F_3I_3)\}$. Obviously, outcome temporal pattern P_{SI} is easier to obtained interval-relation knowledge, which is $\triangle{*}T_4F_3(2,1,0)$ = “Left Contain of T_4 and F_3”. Finally, computation complexity of episode is

$$episode^{\#0} = 4 \times (4+4+4+5)^4 = 334084$$
$$episode^{\#1} = 9 \times (4+4+4+5)^3 = 44217$$
$$episode^{\#} = 334084 + 44217 = 378301$$

5.3 Meta-code transformation complexity analysis

In section 5.2, four secondary meta-code time series { F, I, T, Y } could be transformed to eight primary meta-code time-series { F, F’, I, I’, T, T’, Y, Y’ } as following:

$A \Leftarrow F(t) - F(t\text{-}1)$ = Today HKD-EUR Rate – Yesterday HKD-EUR Rate

$F_1 \Leftarrow 0<A \quad F_2 \Leftarrow A \le 0$

$B \Leftarrow F'(t) - F'(t\text{-}1)$ = Today HKD-JPY Rate – Yesterday HKD-JPY Rate

$F'_1 \Leftarrow 0<B \quad F'_2 \Leftarrow B \le 0$

$C \Leftarrow H(t) - H(t\text{-}1)$ = Today HSI value – Yesterday HSI value

$I_1 \Leftarrow 0<C \quad I_2 \Leftarrow C<0 \quad I_3 \Leftarrow \neg(I_1 \vee I_2)$ ‘I_3’ *does not happen in real stock market*

$D \Leftarrow F(t) - H(t)$ = HSI Futures value – Today HSI value

$I'_1 \Leftarrow 0<D \quad I'_2 \Leftarrow D<0 \quad I'_3 \Leftarrow \neg(I'_1 \vee I'_2)$ ‘I'_3’ *does not happen in real stock market*

$E \Leftarrow$ (Turnover / Capitalisation) 100% $\quad P \Leftarrow E(t) - E(t\text{-}1)$

$T_1 \Leftarrow 0<P \quad T_2 \Leftarrow P \le 0$

$E \Leftarrow$ (Turnover / Capitalisation) 100% $Q = 0.3\%$

$T'_1 \Leftarrow E<Q$ $T'_2 \Leftarrow Q \leq E$

$M \Leftarrow Y(t) - Y(t\text{-}1)$ = Today 1-month Yield – Yesterday 1-month Yield

$Y_1 \Leftarrow 0<M$ $Y_2 \Leftarrow M<0$ $Y_3 \Leftarrow \neg(Y_1 \vee Y_2)$

$N \Leftarrow Y'(t) - Y'(t\text{-}1)$ = Today 10-year Yield – Yesterday 10-year Yield

$Y'_1 \Leftarrow 0<M$ $Y'_2 \Leftarrow M<0$ $Y'_3 \Leftarrow \neg(Y'_1 \vee Y'_2)$

Computation complexity of episode:

$$episode^{\#0} = 8 \times (2+2+2+2+2+2+3+3)^8 = 8 \times 18^8 \approx 8.816 \times 10^{10}$$

$$episode^{\#1} = 49 \times (2+2+2+2+2+2+3+3)^7 = 49 \times 18^7 \approx 3.000 \times 10^{10}$$

$$episode^{\#} = 8 \times 18^8 + 49 \times 18^7 \approx 1.1816 \times 10^{11}$$

Comparing Section 5.2 and 5.3, the *episode*$^{\#}$ is 378301 and 1.1816×10^{11} respectively. Their transformations are different, but they have same scope of attributes in mining. And the primary meta-code could be easier to detect extra-relationship between time-series rather than secondary meta-code. Primary meta-code is more factual in mining, but sequences pruning are occurred very huge number of episodes ($1.1816 \times 10^{11} \gg 378301$). It should be solved this problem to improve the model in the future.

6. Discussion

A novel algorithm for discovering statistical significant pattern in multiple time-series, but may not be a frequent pattern. The MMS model in space-time causal relationship can be used frequent mining to obtain optimal pattern as well. As those results, the multiple sequences mining mechanism of MMS model are simple and workable in real-life applications. **Primary meta-code** in binary and unity attribute is the best methodology for temporal pattern mining. Detecting **extra-relationship** technique is the solution of multiple time-series target pattern mining. Relations between meta-codes rather than intervals are used as the basic units of knowledge. **Interval-relation knowledge** mining can be instead of time **code-based** relations, which has many advantages, like as easier to implement in *alignment*, *clustering*, *associations*, *prediction*, *multi-dimensional pattern* and *relation-seeking* for decision mining. Causal-fork does not apply backward cause-effect between-relation during mining. Hence, causal relationship is a good tool for relations knowledge mining in real-life application domain. In fact, *interactive fork* approaching could be developed between-relation of multiple DNA sequence mining in near future.

• Bibliography

[1] Maintaining Knowledge about Temporal Intervals
James F. Allen.
Communications of the ACM, Vol.26, Number 11, pp. 832-843, 1983.

[2] Mining Sequential Patterns
Rakesh Agrawal and Ramakrishnan Srikant.
In Proc. of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, March 1995.

[3] Composition Alignment
Gary Benson. (2003)
Algorithms in Bioinformatics. Third International Workshop, WABI 2003, pp. 447-461

[4] Temporal Aggregation and Spurious Instantaneous Causality in Multiple Time Series Models
Jörg Breitung and Norman R. Swanson
Journal of Time Series Analysis, VOL.23, No.6, Nov. 2002, pp.651-665.

[5] Discovering Frequent Event Patterns with Multiple Granularities in Time Sequences
Claudio Bettini, Sean Wang, Sushil Jajodia and Jia-Ling Lin.
IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 2, March/April, 1998.

[6] Learning Sequential Patterns for Probabilistic Inductive Prediction
Keith C.C. Chan, Andrew K.C. Wong and David K.Y. Chiu
IEEE Transactions on System, Man, and Cybernetics, VOL.24, No.10, 1994

[7] Rules Discovery from Time Series
Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan and Padhraic Smyth.
In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), August 27-31, 1998, New York, USA., pp16-22.

[8] Temporal Reasoning Based on Semi-Intervals
Christian Freksa.
Artificial Intelligence, Vol.54, Number 1, pp.199-227, 1992.

[9] Meta-Rule-Guided Mining of Association Rules in Relational Databases
Yongjian Fu and Jiawei Han.
In Proc. 1st Int. Workshop Integration of Knowledge Discovery with Deductive and Object-Oriented Databases (KDOOD'95), pages 39-46, Singapore, Dec. 1995.

[10] Noisy Time Series Prediction using Recurrent Neural Networks and Grammatical Inference
C. Lee Giles, Steve Lawrence and Ah Chung Tsoi
Machine Learning, 44, 2001, pp161-183.

[11] Learning Dependencies in Multivariate Time Series
Frank Höppner.
In Workshop on Knowledge Discovery in (Spatio-) Temporal Data at the 15th Eureopean Conference on Artificial Intelligence (ECAI'02), pp.25-31, 2002.

[12] Time-Series Segmentation and Symbolic Representation, from Process-Monitoring to Data-Mining
Bernard Hugueney and Bernadatte Bouchon-Meunier.
In Proceedings of the International Conference, 7th Fuzzy Days on Computational Intelligence, Theory and Applications, pp.118-123, 2001.

[13] Discovering Sequential Association Rules with Constraints and Time Lags in Multiple Sequences
Sherri K. Harms, Jitender Deogun and Tsegaye Tadesse.
In Proceedings of the 2002 International Symposium on Methodologies for Intelligent Systems (ISMIS '02), pages 432—442

[14] Symbolic Representation of Long Time-Series
Georges Hebrail and Bernard Hugueney.
In Conference on Applied Statistical Models and Data Analysis (ASMDA), pp.537-542, June 2001.

[15] Adaptive Query Processing for Time-Series Data
Yun-Wu Huang and Philip S. Yu.
In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, California,USA, August,1999, pp 282-286.

[16] New Introduction to Multiple Time Series Analysis
Helmut Lütkepohl.
Springer 2005.

[17] A Symbolic Representation of Time Series, with Implications for Streaming Algorithms
Jessica Lin, Eamonn Keogh, Stefano Lonardi and Bill Chiu.
In Proceedings of the 8th ACM SIGMOD workshop on Research issues in Data Mining and Knowledge Discovery, San Diego, California, pp.2-11, 2003.

[18] Experiencing SAX: A Novel Symbolic Representation of Time Series
Jessica Lin, Eamonn Keogh, Li Wei and Stefano Lonardi.
Data Mining and Knowledge Discovery, Volume 15, Issue 2 (October 2007).

[19] Unsupervised Pattern Mining from Symbolic Temporal Data
Fabian Mörchen.
ACM SIGKDD Explorations Newsletter, Volume 9, Issue 1, pp.41-55 (June 2007).

[20] Discovery of Frequent Episodes in Event Sequences
Heikki Mannila, Hannu Toivonen and A.Inkeri Verkamo
Data Mining and Knowledge Discovery 1, 1997, pp.259-289.

[21] Efficient Mining of Understandable Patterns from Multivariate Interval Time Series
Fabian Mörchen and Alfred Ultsch.
Data Mining and Knowledge Discovery, Volume 15, Number 2, pp181-215, (October 2007).

[22] Efficient and Robust Feature Extraction and Pattern Matching of Time Series by a Lattice Structure
Polly Wan Po Man and Man Hon Wong.
In Proceedings of the tenth international conference on Information and knowledge management, Atlanta, Georgia, USA, Pages: 271 - 278, 2001.

[23] A Multiresolution Symbolic Representation of Time Series
Vasileios Megalooikonomou, Qiang Wang, Guo Li and Christos Faloutsos.
21st International Conference on Data Engineering (ICDE'05), pp. 668-679.

[24] Mining Frequent Episodes for Relating Financial Events and Stock Trends
Anny Ng and Ada Wai-chee Fu
In Proc. of the 7th Advances in Knowledge Discovery and Data Mining Pacific-Asia Conference, 2003, pp27-39.

[25] Searching for Structure in Multiple Streams of Data
Tim Oates and Paul R. Cohen.
In Proceedings of Thirteenth International Conference on Machine Learning,1996, pp 346-354.

[26] Causality: Models, Reasoning, and Inference
Judea Pearl.
Cambridge University Press, 2000.

[27] Discovering Frequent Poly-Regions in DNA Sequences
Panagiotis Papaetrou, Gary Benson and George Kollios.
In Proceedings of the IEEE ICDM 2006 Workshop on Data Mining in Bioinformatics (DMB 2006), Hong Kong, China, pp 94-98, 2006.

[28] Data Mining of Multiple Non-stationary Time Series
Richard J. Povinelli and Xin Feng.
Proc. of Artificial Neural Networks in Engineering, St. Louis, MO, 1999, 511-516.

[29] Multi-dimensional Sequential Pattern Mining
Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qining Chen and Umeshwar Dayal.
In Proc. 2001 Int. Conf. Information and Knowledge Management (CIKM' 01), Atlanta, GA, Nov. 2001, pp. 81-88.

[30] Discovering Frequent Arrangements of Temporal Intervals
Panagiotis Papaterou, George Kollios, Stan Sclaroff and Dimitrios Gunopoulos.
In Proceedings of International Conference on Data Mining (ICDM), pp 354-361, 2005.

[31] Streaming Pattern Discovery in Multiple Time-Series
Spiros Papadimitrion, Jimeng Sun and Christos Faloutsos.
Proceedings of the 31st VLDB Conference, Trondbeim, Norwny, 2005.

[32] The direction of time
Hans Reichenbach.
University of California, 1956.

[33] Inferring Pairwise Regulatory Relationships from Multiple Time Series Datasets
Yanxin Shi, Tom Mitchell and Ziv Bar-Joseph.
Bioinformatics, Vol. 23 no. 6, 2007, pages 755–763.

[34] Distributed Pattern Discovery in Multiple Streams
Jimeng Sun, Spiros Papadimitrion and Christos Faloutsos.
In Proc. of the 10th Advances in Knowledge Discovery and Data Mining Pacific-Asia Conference, Singapore, April, 2006, pp713-718.

[35] Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures
Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos and Eamonn Keogh
In Proceedings of the 9th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, Washington, D.C., 2003, pp 216-225.

[36] Online Event-driven Subsequence Matching over Financial Data Streams
Huanmei Wu, Betty Salzberg and Donghui Zhang.
In Proceedings of the ACM SIGMOD international conference on Management of data, Paris, France, 2004, pp 23-34.

[37] Choosing the Content of Textual Summaries of Large Time-Series Data Set
Jin Yu, Ehud Reiter, Jim Hunter and Chris Mellish.
Natural Language Engineering, Volume 13, Issue 1 (March 2007).

[38] All the dataset can be downloaded from following links or software:
US Yield: http://www.treasury.gov/offices/domestic-finance/debt-management/interest-rate/yield.shtml
Foreign Currency Exchange Rates: http://www.oanda.com/convert/fxhistory
HSI Turnover & Capitalisation: http://www.hkex.com.hk/data/markstat/statarchive_c.htm
Heng Sang Index: http://hk.finance.yahoo.com/q/hp?s=%5EHSI
HSI Futures: From **Data Stream** Database, File Code: HSICS00

• Notation and Abbreviations

E^{m}_{n}	Event E, which contains nos. of different meta-codes m in n time-series
$Code^{\tau}_{n}$	Meta-code *Code* where n is time-series, τ is the tokens corresponding to n
↕ or ⇳	Mutual Relationship at same time (Synchronous)
↳ or ⇨	Event (t+1) followed time (t)
× or □	Wildcard
∩ or ∧	And
∪ or ∨	Or
¬	Not
τ	Token (inter-attribute)
⇒	Associate to
⇐	Transformation by rule (eg. Code ⇐ Token, Meta-code ⇐ Code)
⌂	Relationship
^{I}⌂	Intra-Relationship
^{E}⌂	Extra-Relationship
^{C}⌂	Cross-Relationship
^{C}E	Causal event
*E	Crossover correlative event
#	Number
∈	belongs to
|	such that
∀	for all
∃	there exists
ℕ	Nature number (integer)
ℝ	Real number
§	Chapter or Section
◊	Operator
⌘	Pattern