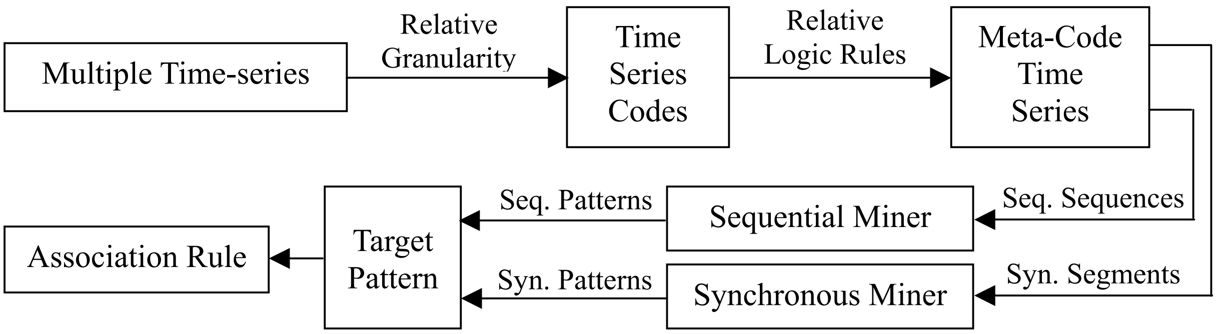
Department of Computing

PhD candidate: Yip Chi Kin
Chief Supervisor: Prof. Keith Chan
Date: 7-7-2007

Discovering Patterns in Multiple Time-series

1. Synchronous Multi-sequence Modeling



Model Characteristics

- Multi-sequence pattern (including extra-relationship)
- Real-life applications
- Possible and simple implementation

2. Synchronous Multiple-sequence

Multi-sequence: $\{ (A_1 B_3 C_3 D_2), (A_3 B_4 C_2 D_1), (A_1 B_2 C_1 D_3), (A_2 B_1 C_2 D_4), \ldots \}$

Equivalent seq.: $\{ (B_3 A_1 D_2 C_3), (C_2 D_1 B_4 A_3), (D_3 B_2 A_1 C_1), (A_2 C_2 D_4 B_1), \ldots \}$

Multi-Sequence Pattern: $\{ (A_2 B_1 C_2 D_4), (A_3 B_4 C_4 D_1), (A_4 B_3 C_2 D_1) \}$

Relationship notation: $(\updownarrow(A_2 B_1 C_2 D_4)) \mapsto (\updownarrow(A_3 B_4 C_4 D_1)) \mapsto (\updownarrow(A_4 B_3 C_2 D_1))$

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	A_4
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	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	C_1
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	D_4
Time Granularities	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	…	t+k	…	t+n-1	t+n

Mathematics notation: $M^{Syn_j^i(e)}_{{}_u Seq_v(t)}$

where
- i is nos. of codes in each series
- j is nos. of time-series
- u is starting time of series
- v is sending time of series
- t is time granularity of pattern
- e is events of series

Above Multi-Sequence is $M^{\updownarrow A^5 B^4 C^6 D^4}_{\mapsto_1 t_n}$

Above Pattern is $M^{\updownarrow(A_2B_1C_2D_4)\updownarrow(A_3B_4C_4D_1)\updownarrow(A_4B_3C_2D_1)}_{\mapsto(t_4 t_5 t_6)}$, hence t_4 is $(A_2 \updownarrow B_1 \updownarrow C_2 \updownarrow D_4)$

3. Pattern Formats

Standard Pattern

A_3	X	X	X
B_1	B_2	X	X
C_2	X	C_3	X
D_2	X	X	D_2

Weak Relative Pattern

X	X	A_2	X
B_1	X	X	X
X	X	X	C_2
D_2	X	X	X

Strong Relative Pattern

A_2	A_1	X	A_2
X	B_2	B_1	B_3
X	C_4	C_3	X
D_2	D_1	X	D_2

Tree Pattern

X	X	A_2	A_2
X	B_2	X	B_3
C_2	X	C_3	C_2
X	D_1	D_1	D_2

where X is wild-card of pattern

- Non-target Patterns

Perfect Pattern

A_3	A_1	A_2	A_2
B_1	B_2	B_1	B_3
C_2	C_4	C_3	C_2
D_2	D_1	D_1	D_2

Not happen in real situation

Missing Sequence

A_3	X	X	X
B_1	X	X	X
C_2	X	X	X
D_2	X	X	X

Missing Synchronous

A_3	X	X	X
X	B_2	X	X
X	X	C_3	X
X	X	X	D_2

Non-multiple Sequences

A_3	A_1	A_2	A_2
X	X	X	X
X	X	X	X
X	X	X	X

- Target Pattern

Pattern should be included *Synchronous* and *Sequential* pattern bonding with *Singular Event*.

Trigger Event B_1

Synchronous

A_3
B_1
C_2
D_4

and

Sequential

B_1	B_2	C_3	D_2

⇨

Ideal Pattern

A_3	X	X	X
B_1	B_2	X	X
C_2	X	C_3	X
D_4	X	X	D_4

Temporal Relations Pattern
Fixed Time Granularities

A_2	Ø	X	Ø	X	Ø	X
B_1	Ø	B_2	Ø	X	Ø	X
C_2	Ø	X	Ø	C_3	Ø	X
D_4	Ø	X	Ø	X	Ø	D_2

Ø Wild Card of Events in Time-series, e.g. Sunday or Holiday in financial market.

- Metarule-Guided of Association Rule

Meta-rule template:

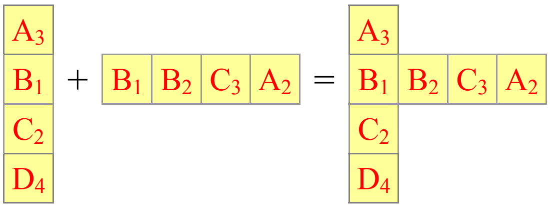
If $P_i(t+k) = Q_j(t+k)$ then

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

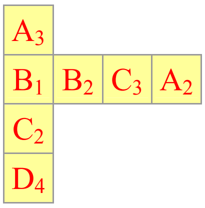
where P *is in synchronous* and Q *is in sequential*
P_i (i = A, B, C, D)
Q_j (j = A, B, C, D)
k = 0, 1, 2, 3

• Correlative Patterns (*single event junction*)

Actual Relationship Pattern of B_1



Direct relationship in synchronous pattern
$B_1 A_3$, $B_1 C_2$, $B_1 D_4$
Indirect relationship in synchronous pattern
$A_3 C_2$, $A_3 D_4$, $C_2 D_4$
Direct relationship in sequential pattern
$B_1 B_2$, $B_1 C_3$, $B_1 A_2$
Indirect relationship in sequential pattern
$B_2 C_3$, $B_2 A_2$, $C_3 A_2$



Direct relationship of pattern
$B_1 A_3$, $B_1 C_2$, $B_1 D_4$, $B_1 B_2$, $B_1 C_3$, $B_1 A_2$
Non-direct relationship of pattern
$D_4 B_2$, $C_2 B_2$, $A_3 B_2$, $A_3 C_3$, $A_3 A_2$, …

Relationship bonding probability

If $A_3 B_1 = 0.5$, $B_1 C_3 = 0.9$, then $A_3 C_3 = 0.5 \times 0.9 = 0.45$
If $D_4 C_2 = 0.6$, $C_2 B_1 = 0.8$, $B_1 A_2 = 0.9$, then $D_4 A_2 = 0.6 \times 0.8 \times 0.9 = 0.432$

• Presentation of Target Patterns

Visualization

A_3			A_2
B_1	B_2		
C_2		C_3	
D_4			
t+1	t+2	t+3	t+4

Numerical

A	3			2
B	1*	2		
C	2		3	
D	4			
	t+1	t+2	t+3	t+4

Mathematical Notations

$$\begin{pmatrix} 3 & 0 & 0 & 2 \\ 1* & 2 & 0 & 0 \\ 2 & 0 & 3 & 0 \\ 4 & 0 & 0 & 0 \end{pmatrix} = M^{\updownarrow(A_3 B_{1*} C_2 D_4)\updownarrow B_2 \updownarrow C_3 \updownarrow A_2}_{\mapsto(t_1 t_2 t_3 t_4)}$$

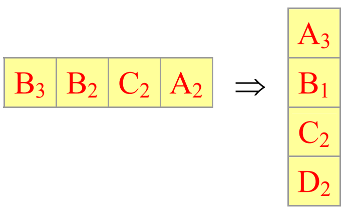
where * is the correlative event

• Correlative Junction

Correlative junction event (C_2) & (C_3):
The sequence $B_3 \Rsh B_2 \Rsh C_2 \Rsh A_2$ can be expressed in $B_3(t) \wedge B_2(t+1) \wedge C_2(t+2) \wedge A_2(t+3)$ format.
And segment $A_3 \wedge B_1 \wedge C_3 \wedge D_2$ is expressed in $A_3(t+2) \wedge B_1(t+2) \wedge C_3(t+2) \wedge D_2(t+2)$ format.

Sequential Bonding

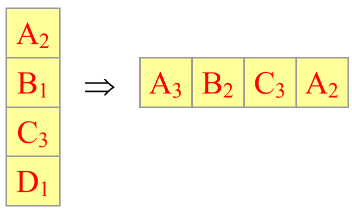
In case trigger event is C_2 the rule as below:
If $B_3 \Rsh B_2 \Rsh C_2 \Rsh A_2$ Then $A_3 \wedge B_1 \wedge C_2 \wedge D_2$



In case trigger sequence is $B_2 \Rsh C_2$ the rule is
If $B_3 \Rsh B_2 \Rsh C_2 \Rsh A_2$ Then $A_3 \wedge B_1 \wedge C_2 \wedge D_2$

Synchronous Bonding

In case trigger event is C_3 the rule as below:
If $A_2 \wedge B_1 \wedge C_3 \wedge D_1$ Then $A_3 \Rsh B_2 \Rsh C_3 \Rsh A_2$



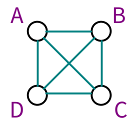
In case trigger events are $A_2 \wedge C_3$ the rule is
If $A_2 \wedge B_1 \wedge C_3 \wedge D_1$ Then $A_3 \Rsh B_2 \Rsh C_3 \Rsh A_2$

4. Synchronous Pattern Correlative intersection

- Correlative intersect cases

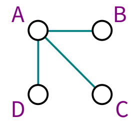
Mutual Relationships among 4 Synchronous Events

A↕B↕C↕D



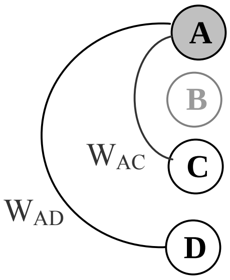
AB-AC-AD-BC-BD-CD

e.g. AB →0, AC →1, AD →2,
BC →3, BD →4, CD →5.



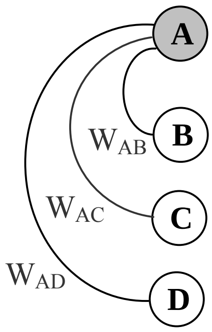
A↕B↕C↕D
→ AB-AC-AD
→ 0-1-2

- Local maximal segment



Optimal
Synchronous
Segment
$^{C}\S A_i \updownarrow C_j \updownarrow D_k$

- Global maximal segment



Optimal
Synchronous
Segment
$^{C}\S A_w \updownarrow B_x \updownarrow C_y \updownarrow D_z$

● Correlative Junction Event

n is number of time-series
w is numbers of wild card

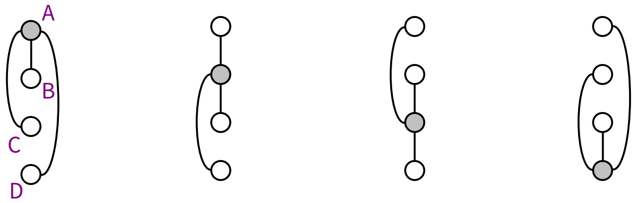
$$\sum \text{Syn\_seg\_\#w} = n \times \binom{p}{r} = n \times \frac{p!}{r!(p-r)!}$$

where $p = n - 1$, $r = n - w - 1$

$$\sum \text{Syn\_seg} = n(2^{n-1} - 1)$$

Possible Correlative Event Relationships among 4 Synchronous Events

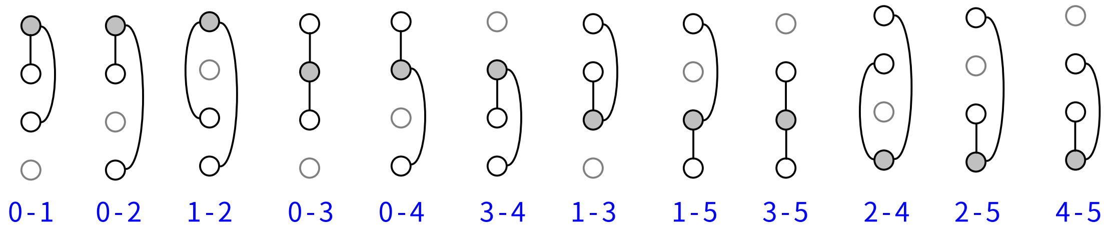
$$\sum \text{Syn\_seg\_\#0} = 4 \times \binom{3}{3} = 4 \times \frac{3 \times 2 \times 1}{(3 \times 2 \times 1)(1)} = 4 \times 1 = 4$$



0-1-2 0-3-4 1-3-5 2-4-5

Possible Correlative Event Relationships among 3 Synchronous Events

$$\sum \text{Syn\_seg\_\#1} = 4 \times \binom{3}{2} = 4 \times \frac{3 \times 2 \times 1}{(2 \times 1)(1)} = 4 \times 3 = 12$$



Maximum Synchronous sequence

$Code^i_j$ means $Code^i_1, Code^i_2, Code^i_3, Code^i_4 = A^4, B^4, C^4, D^4$ where $j = 4$

$$\prod Code^i_j \times \sum \text{Syn\_seg\_\#0} = (4 \times 4 \times 4 \times 4) \times 4 = 1024$$

$$max \left\lceil \prod Code^i_{j-1} \right\rceil \times \sum \text{Syn\_seg\_\#1} = (4 \times 4 \times 4) \times 12 = 768$$

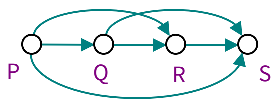
$$max \left\lceil \prod Code^i_{j-2} \right\rceil \times \sum \text{Syn\_seg\_\#2} = (4 \times 4) \times 12 = 192$$

5. Sequential Pattern

- Correlative intersect cases

Mutual Relationships among 4 Sequential Events

$P \mapsto Q \mapsto R \mapsto S$



PQ-PR-PS-QR-QS-RS

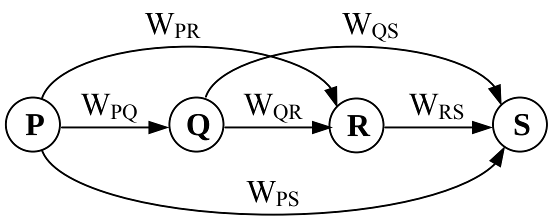
Reference: PQ →0, PR →1, PS →2,
QR →3, QS →4, RS →5.

Codes of Position (p)

$pos1 = p+1 = P \mapsto Q = Q \mapsto R = R \mapsto S.$

$pos2 = p+2 = P \mapsto R = Q \mapsto S.$

$pos3 = p+3 = P \mapsto S.$



Total possible cases of sequential sequences:

$$\sum \text{S\_seq} = \sum_{w=0}^{n-2} (n-w) \times \binom{n-1}{w}$$

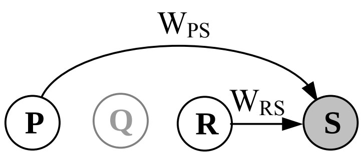
where n is number of time-series
w is numbers of wild card

e.g. n = 4,

$$\sum S\_seq = \sum_{w=0}^{n-2}(n-w)\times\binom{n-1}{w} = 4\times\binom{4}{0}+3\times\binom{4}{1}+2\times\binom{4}{2} = 4\times1 + 3\times3 + 2\times3 = 19$$

$$\sum S\_seq\_\#w = (n-w)\times\binom{n-1}{w} = (n-w)\times\frac{(n-1)!}{w!(n-w-1)!}$$

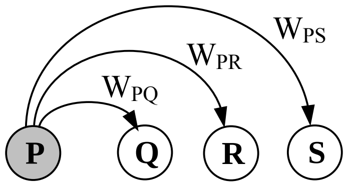
- Local maximal sequencing



Optimal Sequential Sequence

P_t ↱ R_{t+2} ↱ ^{C}§S_{t+3}

- Global maximal sequencing



Optimal Sequential Sequence

^{C}§P_t ↱ Q_{t+1} ↱ R_{t+2} ↱ S_{t+3}

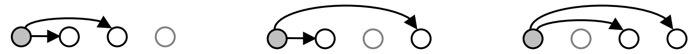
Possible Correlative Event Relationships among 4 Sequential Events

$$\sum S\_seq\_\#0 = (4-0)\times\frac{(4-1)!}{0!(4-0-1)!} = 4\times\frac{3!}{3!} = 4\times1 = 4$$

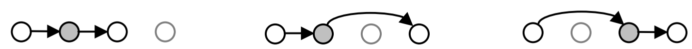
0-1-2 0-3-4 1-3-5 2-4-5

Possible Correlative Event Relationships among 3 Sequential Events

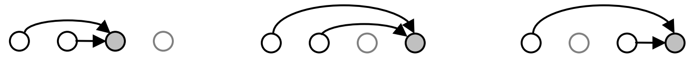
$$\sum S\_seq\_\#1 = (4-1)\times\frac{(4-1)!}{1!(4-1-1)!} = 3\times\frac{3!}{1!2!} = 3\times3 = 9$$



0-1 0-2 1-2



0-3 0-4 1-5



1-3 2-4 2-5

Maximum numbers of sequential sequences:

$Code_j^i$ means $Code_1^i, Code_2^i, Code_3^i, Code_4^i = A^4, B^4, C^4, D^4$ where $j = 4$

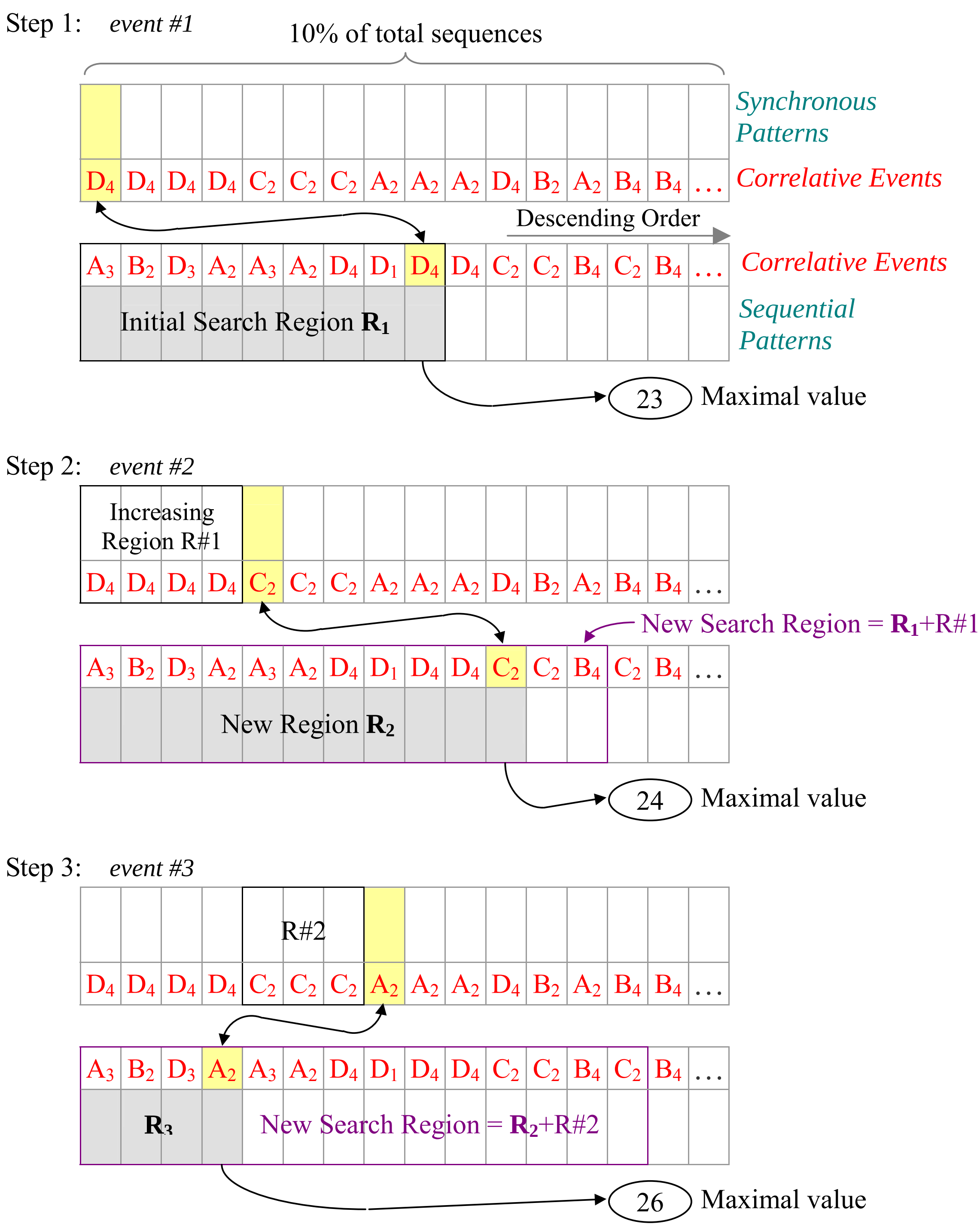
$(\sum Code_j^i)^n \times \sum S\_seq\_\#0 = (4+4+4+4)^4\times 4 = 65536\times 4 = 262144$

$(\sum Code_j^i)^{n-1} \times \sum S\_seq\_\#1 = (4+4+4+4)^3\times 9 = 4096\times 9 = 36864$

$(\sum Code_j^i)^{n-2} \times \sum S\_seq\_\#2 = (4+4+4+4)^2\times 6 = 256\times 6 = 1536$

6. Combine Synchronous and Sequential Pattern

- Shorten Search Region Algorithm



- Shorten Search Region Algorithm

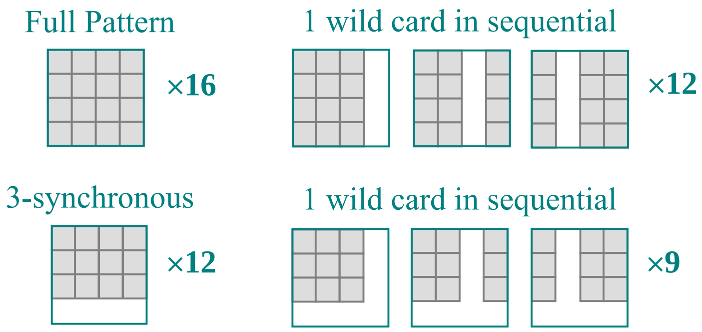
```
Search_Region( ^{search}#region, ^{match}#syn )
{ for next_syn ∈ syn E_1^4 C§(^{memory}#syn) do
    if (^{max}syn E_1^4 C§(^{max}#syn) == seq E_1^4 C§(#next_syn))
    { ^{match}#syn ← #next_syn;
      ^{match}#seq ← ^{search}#region + #Region(weight(#next_syn) – weight(^{match}#syn));
    }
}

Corr_Event_Match( ^{max}weight, ^{match}#syn, ^{match}#seq )
{ for next_seq ∈ ^{match}#seq do
    if (^{max}syn E_1^4 C§(^{match}#syn) == seq E_1^4 C§(#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;
    }
}

Shorten_Search_Region( syn_wt, syn E_1^4 C§, E_4^4 ↕, seq_wt, seq E_1^4 C§, E_4^4 ↦ )
{ ^{max}syn E_1^4 C§(^{max}#syn) ← syn E_1^4 C§#1;
  ^{max}seq E_1^4 C§(^{max}#seq) ← seq E_1^4 C§#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
  { Search_Region( ^{search}#region, ^{match}#syn );
    Corr_Event_Match( ^{max}weight, ^{match}#syn, ^{match}#seq );
    ^{match}#syn++ ;
  }
  M_{1 Seq_4(t)}^{Syn_4^4(e)} ← { E_4^4 ↕ §(^{max}#syn)} ∩ { E_4^4 ↦ §(^{max}#seq)}
}
```

7. Information Gain of Pattern

- 4×4 Pattern contents wild cards (Information Gain dependence on numbers of events)



8. Model Implementation

- Model Testing

Linear congruential randomization (uniform distribution) is:

$$\{ x^i_n \mid x^i \in \{1 \le x^i \le 6\} \text{ and } x_1, x_2, \ldots, x_n \mid n = 1000\}$$

e.g. Throw 3 times of 4 dices (dice A, B, C and D, with 6-sided block of each).
The multi-sequence is $\{ (A_1\ B_6\ C_3\ D_2), (A_3\ B_4\ C_5\ D_1), (A_6\ B_2\ C_4\ D_3) \}$

Algorithm LCR_Dataset
Output: Four text files contains $\{ x^i_n \mid x^i \in \{1 \le x^i \le 4\} \text{ and } x_1, x_2, \ldots, x_n \mid n = 1000\}$
void
{ **for** FileName ∈ *nos_of_file* **do**
  { **for** $i \in$ *file_length* / 4 **do**
      $\forall x_i \leftarrow \{ x_1, x_2, x_3, x_4 \mid x_i = (1,2,3,4) \}$;
    Generate 1000 LCR random number $\forall x \in \{0 \le x \le 32767\}$;
    $position[x_n] \leftarrow x$; $sort[x_n] \leftarrow x$; // Created duplicate code list for sorting
    Bubble sorting of random number $sort[x_n]$;
    for $j \in$ *file_length* **do**
      for $k \in$ *file_length* **do**
        if $sort[j] = position[k]$
          { $sequence[j] \leftarrow x_k$; // Insert code into *sorted list* follow random sequence
            $x_k \leftarrow 0$; // Avoid duplicate in *sorted list* by clearing memory
            $k \leftarrow$ *file_length*; // Break the looping
          }
    fileOutput "FileName" $\leftarrow sequence[x_n]$;
  }
}

- Loss Extra-relationship Pattern

If the output multi-sequence pattern contents more than 70% of intra-relationship in sequential sequence,
then Let weight = 0 (intra-relationship > 70%), and find a new optimal sequential sequence
OR Given constraints to find an additional extra-relationship sequential sequence.

- **Real-life application:** (Stock Market Behaviour example)

Primary Meta-code → Simple and Identical attribute, higher contribution in two events.

A = Today HSI Futures value – Yesterday HSI Futures value = F(*t*) – F(*t-1*)

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

B = Today HSI value – Yesterday HSI value = H(*t*) – H(*t-1*)

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

D = HSI Futures value – Today HSI value = H(*t*) – F(*t*)

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

C = (Turnover / Capitalisation) 100% P = C(*t*) – C(*t-1*)

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

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

$$\begin{pmatrix} 3* & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} = M^{\updownarrow(F_{3*}I_2T_2)\updownarrow I_2 \updownarrow F_1}_{\mapsto(t_1t_3t_4)}$$

where * is the correlative event

The pattern (Year 2003-2006) 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".

Short-term Patterns

Year 2003 is $M^{\updownarrow(F_{3*}I_1T_2)\updownarrow T_2 \updownarrow T_2}_{\mapsto(t_1t_2t_3)}$ Year 2004 is $M^{\updownarrow I_1 \updownarrow I_1 \updownarrow(I_{1*}F_1H_1)\updownarrow I_1}_{\mapsto(t_1t_2t_3t_4)}$

Year 2005 is $M^{\updownarrow I_1 \updownarrow I_1 \updownarrow(I_{1*}F_1H_1)\updownarrow I_1}_{\mapsto(t_1t_2t_3t_4)}$ Year 2006 is $M^{\updownarrow I_2 \updownarrow(I_{2*}F_2H_2)\updownarrow I_2}_{\mapsto(t_1t_3t_4)}$

- **Secondary Meta-code example:** ***Financial Market Behaviour*** (Similar attributes)

E = HKD-EUR Rate Up J = HKD-JPY Rate Up

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

t = Today HSI value – Yesterday HSI value f = HSI Futures value – Today HSI value

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

$I_1 = U_t \wedge U_f$ $I_2 = U_t \wedge D_f$ $I_3 = D_t \wedge U_f$ $I_4 = D_t \wedge D_f$

C = (Turnover / Capitalisation) 100% P = C(*t*) – C(*t-1*) A = 0.3%

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

m = 1-month Yield n = 10-year Yield M = m(*t*) – m(*t-1*) N = n(*t*) – n(*t-1*)

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

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

All the indicators are selected from year 2003 to 2006.

Long-term Pattern of extra-relationship is $M^{\updownarrow I_1 \updownarrow T_2 \updownarrow(F_3I_3T_{4*})}_{\mapsto(t_1t_2t_3)}$

• Bibliography

[CWC94] 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

[FH95] 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.

[B03] Composition Alignment
Gary Benson. (2003)
Algorithms in Bioinformatics.
Third International Workshop, WABI 2003, pp. 447-461

[AS95] Mining Sequential Patterns
Rakesh Agrawal and Ramakrishnan Srikant.
In Proc. of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, March 1995.

• Notation and Abbreviations

Syn_j^i	Synchronous *Syn* events of different codes *i* in each series and nos. of series *j*
$_u Seq_v$	Sequential *Seq* events from time *u* to *v*
M_{Seq}^{Syn}	Multi-sequence *M* with synchronous and sequential sequence
E_m^n	Event *E*, which contents nos. of different codes *m* in *n* series
$\cap$ or $\wedge$	And
$\cup$ or $\vee$	Or
$\times$ or □	Wild Card
↱ or $\mapsto$	Event (t+1) followed time (t)
$\neg$	Not
$\updownarrow$	Mutual Relationship at same time (Synchronous)
$\Rightarrow$	Associate to
§	Relationship
^{E}§	Extra-Relationship
^{C}§	Correlative event
#	Number
$\in$	belongs to
|	such that
$\forall$	for all
$\exists$	there exists
⌂	Operator

```
Local_Maximal_Segment( weights, ↕ E_j^2 )
{ for i ∈ number_of_file do
      _FileName ← { x_i y_i | x_i ∈ {_A,_B,_C,_D},   y_i ∈ {A,B,C,D}};
      while (!fileInput.eof( ))
         { fileInput⟩⟩ Weight[i][n] ← { w_n | w_n ∈ real number };
           fileInput⟩⟩ Event#1[i][n] ← { x_n y_n | x_n ∈{A,B,C,D},  y_n ∈{1,2,3,4}};
           fileInput⟩⟩ Event#2[i][n] ← { x_n y_n | x_n ∈{A,B,C,D},  y_n ∈{1,2,3,4}};
           Length[n] ← n++;
         }
      Create lookup table for local synchronous relationship of three events
      Assigned variables Correlative_Event, Link#1 and Link#2 from lookup table
      for j ∈ Length[Link#1] do
           for k ∈ Length[Link#2] do
                if ( Correlative_Event[Link#1[j]] = Correlative_Event[Link#2[k]] )
                   {  Store the weight into memory }
      Bubble sort of positive weights only
      fileOutput "FileName" ←↕ E_j^3;
  }
```

```
Global_Maximal_Segment( weights, ↕ E_t^2 )
{ for i ∈ number_of_file do
      _FileName ← { x_i y_i | x_i ∈ {_A,_B,_C,_D},  y_i ∈ {A,B,C,D}};
      while (!fileInput.eof( ))
         {  Input Event#1( weights, ↕ { x_n y_n | x_n ∈{A,B,C,D},  y_n ∈{1,2,3,4}});
            Input Event#2( weights, ↕ { x_n y_n | x_n ∈{A,B,C,D},  y_n ∈{1,2,3,4}});
         }
      Create lookup table for global synchronous relationship of four events
      Assigned variables cor_event, link#1, link#2 and link#3 from lookup table
      for i ∈ numbers_of_link#1 do
         for j ∈ numbers_of_link#2 do
            for k ∈ numbers_of_link#3 do
               if ∃ (cor_event_link#1[i] ∧ cor_event_link#2[j] ∧ cor_event_link#3[k] )
                  { weights ← weight_link#1[i] + weight _link#2[j] + weight _link#3[k];
                    ↕ { x_n y_n | x_n ∈{A,B,C,D},  y_n ∈{1,2,3,4}});
                  }
      Bubble sort of positive weights only
      fileOutput "FileName" ← ( weights, ↕ E_t^4 )
  }
```