Department of Computing

Guided Studies: COMP6814 Data Mining
PhD candidate: Yip Chi Kin
Chief Supervisor: Prof. Keith C.C.Chan
Presentation Date: 18-5-2006 (3:30pm Room PQ703)
Mining Probabilistic Sequential Patterns

• Studied Papers

[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

[YWY03] Mining Asynchronous Periodic Patterns in Time Series Data
Yang, J., Wang, W., and Yu, P. (2003)
IEEE Transactions on Knowledge and Data Engineering. 15(3):813-628

[YWY01] Meta-patterns Revealing Hidden Periodic Patterns
Yang, J., Wang, W., and Yu, P. (2001)
IEEE International Conference on Data Mining (ICDM). pp. 550-557

[YWY03a] Infominer: Mining Surprising Periodic Patterns
Yang, J., Wang, W., and Yu, P. (2003)
Proc. of the 7th ACM Int'l Conf. on Knowledge Discover and Data Mining (KDD). pp.395-400

[YWY03b] Discovery of Statistically Important Pattern Repeats in a Long Sequential Data
Yang, J., Wang, W., and Yu, P. (2003)
Proc. of the 3rd SIAM International Conference on Data Mining (SDM).

• Report Contents:

Chapter 1. Introduction
Chapter 2. Terminology
Chapter 3. Periodic Patterns
Chapter 4. Sequences Structure (*Synchronous Multi-Sequence*)
Chapter 5. Algorithms for finding Surprising Pattern
Chapter 6. Algorithms for Prediction
Chapter 7. Further Enhancements
Chapter 8. Comparative Studies of models
Chapter 9. Conclusions

Mining Probabilistic Sequential Patterns

1. Introduction

A huge amount of data is collected every day in the form of event-time sequences. Common examples are the recording of different values of stock shares during a day, every access to a computer by external network, bank transactions, or events related to malfunctions in an industrial plant. These sequences represent valuable sources of information, not only what is explicitly registered, but also for deriving implicit information and for predicting the future behavior of the process that we are monitoring. The latter activity requires an analysis of the frequency of certain events, discovery of their regularity, or discovery of sets of events that are related by particular temporal relationships. Such frequency, regularity, and relationships are very expressed in terms of multiple granularities, and thus analysis and discovery the temporal sequences must be able to deal with these granularities.

Formally, a string is a sequence of symbols. Sequence is one of the basic data types to carry information. There are several methods to analyse the sequences, such as probabilistic modelling, exact matching, approximate matching. A generalisation of the string-matching problem is the *Approximate String Matching Problem*, which involves finding substrings of a text string similar to given pattern string. This variation of the problem is important when errors are being taken into consideration, and, for example, finds application in the field of molecular biology sequencing. This approach is that involving don't-care, or wild-card, symbols which match any single symbol, including another don't-care. Note that here the problem is partly complicated by the Period Pattern [YWY03] and Meta-period Pattern [YWY01], whose are described in Chapter 3.

There are five algorithms in this report. The first two papers [YWY03] [YWY01] are the preliminary approach of the basic requirements for sequential mining. The other three papers [YWY03a] [YWY03b] [CWC94] are used for mining probabilistic sequence. Two models of

algorithm named **YWY model** [YWY03a] [YWY03b] and **CWC model** [CWC94]. Here is the related features described briefly.

- Period Pattern based on [YWY03] algorithm

  The pattern of subsequence is defined by *min_rep* and *max_dis*, those algorithms is based on exact matching to find the surprising pattern.

- Meta-pattern based on [YWY01] algorithm

  According to previous model, some noises in dataset disturb to find the pattern. Therefore, this model using meta-pattern to recover the noises by don't care events eliminations.

- ***YWY model*** based on [YWY03a] [YWY03b] algorithm

  There are two algorithms InfoMiner and Stamp of probabilistic mining. Both are using fixed period to capture statistically significant patterns. Using [YWY03b] model, all the significant patterns will be placed in very close together in the sequence, [YWY03a] model may be not.

- ***CWC model*** based on [CWC94] algorithm

  This is a probabilistic model as well, but it predicts the pattern by weight of evidence optimization. The model could be applied in synchronous multi-sequence pattern mining, example details is shown in chapter 7.

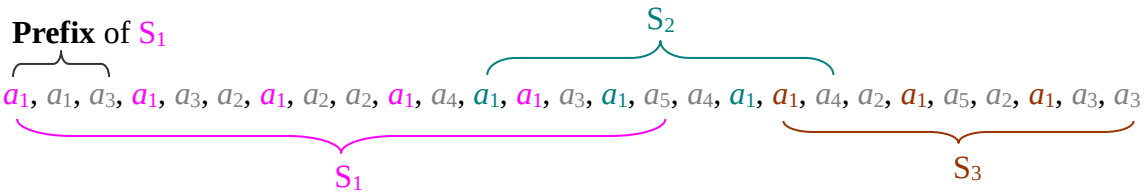
In this report, the comparative studies are mainly concern of ***YWY model*** and ***CWC model***, because of they are in the same nature of modeling (probabilistic).

The prime concern when presenting the algorithms/models described in the forthcoming chapters has been selected and rearranged. It is to be hoped that the simple, concise figures employed for this purpose adequately conveys the basic mechanism of each algorithm. Hence, all the **proofs** and **definitions** would not be repeated them in the report here, details please referred to papers [CWC94] [YWY01] [YWY03] [YWY03a] [YWY03b]. At last, it is preferable priority to read the **terminology** in Chapter 2, because all the basic notations of papers are described here.

2. Terminology

- **Singular** pattern $(*, a_1, *, *)$ where $*$ is don't-care positions
- **Complex** pattern $(a_3, *, a_2, *)$ or $(a_2, *, a_1, a_3)$ or (a_1, a_2, a_4, a_2) …
- **Subpattern** $(a_6, a_2, *, *,)$ and $(*, a_2, *, *)$ … are subpattern of $(a_6, a_2, *, a_4)$
- **Superpattern** $(a_6, a_2, *, a_4)$ is superpattern of $(a_6, a_2, *, *)$ and $(*, a_2, *, *)$ …

- **Prefix** of S_1



if Minimum repetitions (*min_rep*) is 3 then **valid segments** of $(a_1, *, *)$ are S_1, S_2 and S_3.

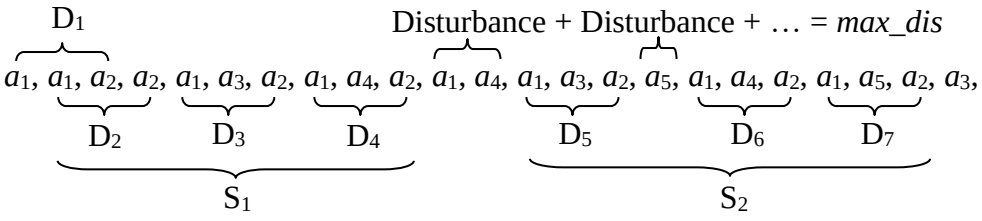
Longest Subsequence (*min_rep* = 3, excluding S_3) is

$(a_1, a_1, a_3, a_1, a_3, a_2, a_1, a_2, a_2, \sout{a_1, a_4}, a_1, a_1, a_3, a_1, a_5, a_4, a_1, a_1, a_4)$

- **Longest Subsequence** of $(a_1, *, *)$ and *min_rep* = 3 is

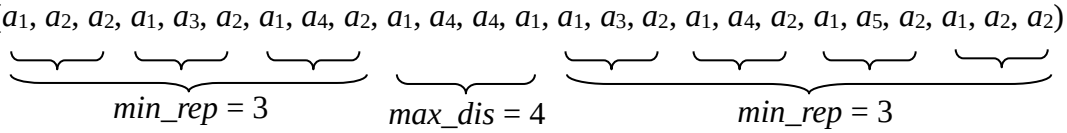
$(a_1, a_1, a_3, a_1, a_3, a_2, a_1, a_2, a_2, a_1, a_4, a_1, a_1, a_3, a_1, \sout{a_5, a_4, a_1}, a_1, a_4, a_2, a_1, a_5, a_2, a_1, a_3, a_3)$

- Here are seven matches segment of $(a_1, *, a_2)$



if *min_rep* = 3 then **valid segments** of $(a_1, *, a_2)$ is S_1, but not S_2.

if *min_rep* = 3 and *max_dis* = 4 then **valid subsequence** as follows:



According to above sequence, the **valid pattern** $(a_1, *, a_2)$ is 2-pattern of period 3.

- Pattern (a_1, a_2, a_3) consists of **Candidate pattern** $(a_1, *, *)$, … , $(a_1, a_2, *)$, … , (a_1, a_2, a_3)

  Consider sequence $(a_1, a_2, a_1, a_2, a_1, a_2, a_1, a_2)$

  (a_1, a_2, a_1, a_2) and (a_1, a_2) are valid patterns if $min\_rep \leq 3$, hence (a_1, a_2, a_1, a_2) is redundant.

  Canonical pattern is (a_1, a_2)

  Derived pattern is (a_1, a_2, a_1, a_2)

- **Information gain**

  Probability of Occurrence and Information of sequence $(a_1, a_2, a_3, a_1, a_2, a_3, a_2, a_3, a_3, a_3)$

Event	Probability	Information gain
a_1	0.2	$I(a_1) = -\log_{\|3\|}(\frac{2}{10}) = 1.465$
a_2	0.3	$I(a_2) = -\log_{\|3\|}(\frac{3}{10}) = 1.096$
a_3	0.5	$I(a_3) = -\log_{\|3\|}(\frac{5}{10}) = 0.631$

  where a is string of symbols $a_1, a_2, \ldots a_m$ of length m

- **Projected subsequence** of $(a_2, *)$ is $(□, □, □, □, a_2, a_3, a_2, a_3, □, □)$
- Maximum information of (a_2, a_3),

  max_info $= I(a_2) + I(a_3) = 1.096+0.631 = 1.727$
- Information gain of $(*, a_1, *, *)$, suppose *min_rep* of $(*, a_1, *, *)$ is 5

  info_gain $= 1.465 \times 5 = 7.325$
- **Bound information gain pruning**, suppose $(*, a_1, *, *)$ is e

  If *Repetitions* < *min_gain*÷*max_info* then remove all events e from the refined candidates list.
- **K most surprising pattern** (i.e. highest information gain)

- **Surprising pattern**

  Mining result of YWY model, it is instead of finding frequent patterns. The surprising pattern is a periodic pattern, which is a list of events that may occur recurrently in the sequence with fixed period length.

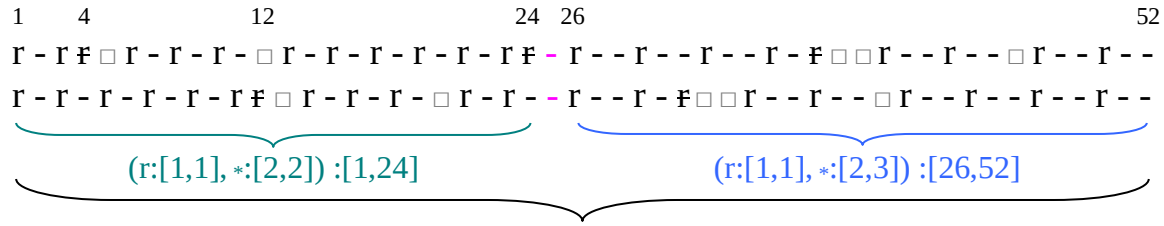
- **Synthetic sequences**

  The testing data for YWY model algorithms, it is a mixed four sequences, each of which consists of 1024 distinct events and 20M occurrences of events.

- **Frequent pattern** subsequence occurred in asynchronous sequence frequent

- Meta pattern model:

  1 … 4 … 52 positions of sequence (52 weeks)
  r means refill order of flu medicine in the corresponding week
  - represents that no flu medicine replenishment in that week
  ŧ noise/distortion of patterns
  □ position eliminated in the sequence



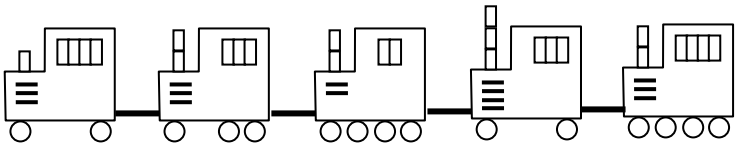
Meta-pattern ((r:[1,1], *:[2,2]):[1,24], *:[25,25], (r:[1,1], *:[2,3]):[26,52])

  * Don't care positions
  Meta-pattern P_1 (r, -) is (r:[1,1], *:[2,2])
  Meta-pattern P_2 (r, -, -) is (r:[1,1], *:[2,3])
  52 is the **span** of meta-pattern (annual)

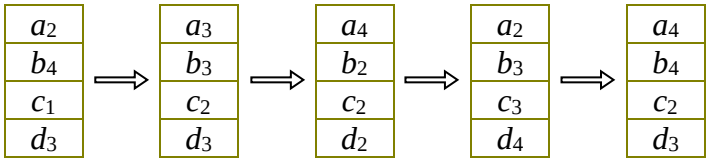
- Notation of Multiple Synchronous Sequence

  Let a_2 be 2 wheels, … , b_2 be 2 windows, … , c_1 be 1 funnel, … , d_4 be 4 stripes.

  In graphics form:



  In symbols form:



  In Text form: { (a_2 b_4 c_1 d_3), (a_3 b_3 c_2 d_3), (a_4 b_2 c_2 d_2), (a_2 b_3 c_3 d_4), (a_4 b_4 c_2 d_3) }

3. Periodic Patterns

The tandem repeats and noises are the main problems should be solved in periodic patterns mining.

- Repeats and Repetitions

Much research has been undertaken to study and research for repetitions in sequences, since many of them have a biological role. There exist many definitions of repeat or a repetition. The first notion of a **repetition** is simple a factor that is contiguously repeated more than twice. Considering complete repetitions, the notion of a maximal repetition represents them all it in a compact way. A repetition is maximal if it cannot be extended in the text to the left or to the right without breaking it. A second notion of **repeat** is used when considering non-contiguous repetitions. These definitions do not take into account the relative positions of the repeats. The last one, the approximate concept of repetition is usually called **tandem repeat**. The concepts are much fuzzy since the notions used for exact repetitions can be extended in various ways, depending on the approximate relation we want between the repeated parts.

- Noises

In [YWY03] [YWY01], the idea of *Minimum Repetitions* (*min_re*p) and *Maximum Distance* (*max_dis*) are the concepts closed to tandem repeat for finding pattern in managing the huge number of occurrences. This is a good way, that the algorithm uses *max_dis* for eliminating noises in the valid subsequence. The pair of *min_rep* and *max_dis* cannot be used in probabilistic sequence. The meta-pattern system can tolerate a greater degree of noises/distortion.

- Longest Subsequence Identification (LSI) algorithm

The purpose of this algorithm [YWY03] is finding asynchronous periodic patterns. The strategies of mining subsequences with most overall repetitions for all possible patterns are through three phases. Firstly, distance-based pruning of candidate patterns. Second step is single pattern verification. Lastly is complex pattern verification. The valid pattern result was shown in chapter 2. The techniques of position scanning are employed, such as *valid_seq*, *ongoing_seq*, and *new_seq*. Details [YWY03] do not elaborated here, because report only concerns probabilistic modeling.

- Meta-Patterns model

Earlier [YWY03] LSI algorithm tends to concentrate on exact matching with extension to handling disturbance symbols (but manage the noise is very limited). Sometimes, some patterns are regular and frequent in sequence, but contents of noises as well. The meta-pattern format is illustrated in chapter 2. Before mining meta-pattern, the candidate meta-pattern should be preprocessing as a specific component consists of don't care event, which could be match on the corresponding position. This algorithm available a priori property, but it does not render sufficient pruning in process. The algorithm applied with different lengths as well.

4. Sequence Structure

- Probabilistic modeling

The length of a longest common subsequence of two sequences can be thought of as a measure of how "close" the sequences are to each other, and in this context it is natural to ask "How close is close?" For example, are the sequences much closer than two randomly generated sequences would be? There are many possible models for random sequences. One may suppose that all letters appear independently on both sequences, and have equal probability, or that they appear independently but perhaps with different probabilities for different letters. Alternatively there may be a fixed set of letters for each sequence, and the observed sequences may be obtained by permuting these letters randomly.

- Information gain dependent on probability

This is a simple way to assign a cost (information gain) that is the logarithm of the probability of this operation occurring in the process that made the sequences differ. Hence the sum of the cost corresponds to the logarithm of the product of the probabilities of the operations, which is a good model if they are independent.

- Probabilistic Synchronous Multi-Sequence

Sequence comparison is about determining similarities and correspondences between two or more sequences. It is related to approximate searching (don't-care approach) and has many applications in computational biology, speech recognition, computer science, coding theory, chromatography, and so on. These applications look for similarities between sequences of symbols.

The following is an example of potential customer investment behaviour in banking:

C_1 = Saving < \$10000 C_2 = Saving > \$20000 C_3 = \$10000 ≤ Saving ≤ \$20000
F_1 = Up > 5%/week F_2 = Down > 5%/week F_3 = 5%Down/wk ≥ Price ≤ 5%Up/wk
S_1 = Up > 1%/day S_2 = Down > 1%/day S_3 = 1%Down/day ≥ Index ≤ 1%Up/day
A_1 = Buy product A_2 = Sell product A_3 = Hold products

Customer asset	C_1	C_3	C_1	C_2	C_1	C_1	C_1	C_2	C_1	C_3	•••
Fund performances	F_3	F_3	F_2	F_2	F_1	F_3	F_2	F_1	F_3	F_1	•••
Stock index time series	S_3	S_2	S_1	S_2	S_3	S_1	S_1	S_2	S_2	S_1	•••
Takes B/H/S Actions	A_2	A_1	A_3	A_2	A_3	A_2	A_1	A_2	A_3	A_3	•••

Time Granularities

The above multi-sequence could be expressed as the following sequence:

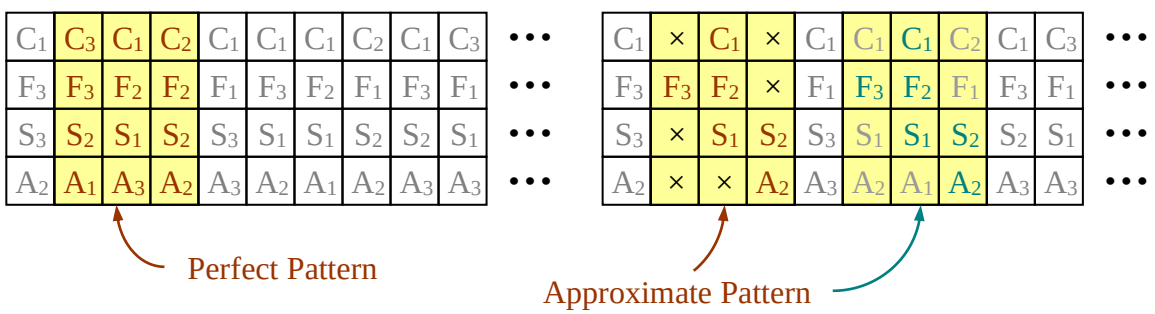
***CWC model*:** { (C_1 F_3 S_3 A_2), (C_3 F_3 S_2 A_1), (C_3 F_2 S_1 A_2), (C_2 F_2 S_3 A_1),… }

***YWY model*:** (C_1, F_3, S_3, A_2, C_3, F_3, S_2, A_1, C_3, F_2, S_1, A_2, C_2, F_2, S_3, A_1, …)

The mining sequence process of YWY model refers to *Chapter 3*. Hence, a supervised pattern (C_3, F_3, S_2, A_1) could be obtained. The mining process of sequential pattern using CWC model was described later in *Chapter 7*.

Unfortunately, in practise, it is almost impossible to find a perfect multi-sequence pattern. Some "Don't care" conditions will be applied in the mining result, in order to find an optimal approximate synchronous multi-sequence easily.

Don't care event = ×



5. Algorithms for Patterns Matching

- **YWY model** (*Calculation of information gain for finding Surprising Pattern*)

This model consists two algorithms, **InfoMiner** algorithm and **Stamp** algorithm. The InforMiner algorithm uses the information gain to measure the important/significance of the occurrence of a pattern. The model is proposed to characterise the class of so-called *Surprising* patterns (instead of frequent patterns). The limitation of InfoMiner is that it does not take into account the location of the occurrence of the patterns in the sequence. In some applications, for instance, tandem repeat in bioinformatics, a series of consecutive repeats are considered more significant than the scattered ones. Therefore, another Stamp algorithm is used, which should be some penalty associated with the gap between pattern repeats. The following are presented those algorithms.

- **InfoMiner algorithm** (*Calculation of information gain without penalty of gaps*)

This model is different from the support model. In this example, minimum information gain threshold (*min_gain* = 4.5) is given. The information gain: $I(a_1) = 1.06$, $I(a_2) = 0.90$, $I(a_3) = 0.67$, $I(a_4) = 1.16$, $I(a_5) = 1.06$, $I(a_6) = 1.45$. The procedure as shown below:

Sequence (a_1, a_3, a_4, a_5, a_1, a_4, a_3, a_3, a_2, a_6, a_3, a_2, a_1, a_4, a_3, a_3, a_1, a_3, a_3, a_5, a_1, a_3, a_4, a_5, a_2, a_3, a_3, a_5, a_1, a_3, a_4, a_5, a_2, a_6, a_3, a_5, a_2, a_6, a_2, a_2)

(, , ,)

Event	Repetition
a_1	5
a_2	3

$I(a_1) = 1.06$

(a_1) > 4.5

(a_2) < 4.5

Event	Repetition
a_3	4
a_4	1
a_6	2

$I(a_6) = 1.16$

($*$, a_3)

($*$, a_4)

($*$, a_6)

Event	Repetition
a_3	4
a_4	2

$I(a_4) = 1.16$

($*$, $*$, a_3)

($*$, $*$, a_4)

Event	Repetition
a_2	2
a_3	1
a_5	4

$I(a_5) = 1.06$

($*$, $*$, $*$, a_2)

($*$, $*$, $*$, a_3)

($*$, $*$, $*$, a_5)

max_info = $I(a_1) + I(a_6) + I(a_4) + I(a_5) = 4.73$

min_rep = $\lceil \frac{4.5}{4.73} \rceil = 1$

Pattern	Information Gain
(a_1, $*$, $*$, $*$)	$I(a_1) \times 5 = 5.30$

Projected Subsequence $(a_1, a_3, a_4, a_5, a_1, a_4, a_3, a_3, a_1, a_4, a_3, a_3, a_1, a_3, a_3, a_5, a_1, a_3, a_4, a_5, a_1, a_3, a_4, a_5)$

$(a_1, \_, \_, \_)$

Event	Repetition
a_3	3
a_4	1

$I(a_4) = 1.16$

Event	Repetition
a_3	2
a_4	2

$I(a_4) = 1.16$

Event	Repetition
a_3	1
a_5	3

$I(a_5) = 1.06$

$max\_info = I(a_1) + I(a_4) + I(a_4) + I(a_5) = 4.44$ $min\_rep = \lceil \frac{4.5}{4.44} \rceil = 2$

Event	Repetition
a_3	3

(a_1, a_3)

Event	Repetition
a_3	2
a_4	2

$(a_1, *, a_3)$

$(a_1, *, a_4)$

Event	Repetition
a_5	3

$(a_1, *, *, a_5)$

Pattern	Information Gain
$(a_1, a_3, *, *)$	$[I(a_1) + I(a_3)] \times 3 = 5.19$
$(a_1, *, *, a_5)$	$[I(a_1) + I(a_5)] \times 3 = 6.36$

Projected subsequence: $(a_1, a_3, a_4, a_5, a_1, a_3, a_3, a_5, a_1, a_3, a_4, a_5, a_1, a_3, a_4, a_5)$

$(a_1, a_3, \_, \_)$

Event	Repetition
a_4	2

(a_1, a_3, a_4)

Event	Repetition
a_5	3

$(a_1, a_3, *, a_5)$

$max\_info = I(a_1) + I(a_3) + I(a_4) + I(a_5) = 4.34$ $min\_rep = \lceil \frac{4.5}{4.34} \rceil = 2$

Pattern	Information Gain
$(a_1, a_3, a_4, *)$	5.78
$(a_1, a_3, *, a_5)$	8.37

Projected subsequence: $(a_1, a_3, a_4, a_5, a_1, a_3, a_4, a_5, a_1, a_3, a_4, a_5)$

$(a_1, a_3, a_4, \_)$

Event	Repetition
a_5	2

(a_1, a_3, a_4, a_5)

$max\_info = I(a_1) + I(a_3) + I(a_4) + I(a_5) = 4.34$

$min\_rep = \lceil \frac{4.5}{4.34} \rceil = 2$

Pattern	Information Gain
(a_1, a_3, a_4, a_5)	7.90

The final result of Surprising Pattern is (a_1, a_3, a_4, a_5)

In the previous figures elaborated the algorithm step by step. The synthetic sequence is used for testing. The overall response time largely depends on the *min_gain* threshold. The threshold *min_gain* is always equal to minimal information gain carried by any pattern in the result pattern. Clearly, with the algorithm proceeds, *min_gain* increases, thus, *min_rep* also increase, and less candidates remain. At the end, the set of result pattern contains the K most surprising patterns. The main pruning power of this algorithm is provided by the bounded information gain pruning technique.

- Stamp **algorithm** (*A series of consecutive repeats*)

This algorithm concerns penalty, which associated distance of events. Those are Generalized Information Gain (GIG), Optimal Information Surplus (OIS) pruning and Maximum Information Gain (MIG) counting. Here describes partly only, see details in paper [YWY03b].

Generalized Information Gain Given: I(a_1) = 1.1, I(a_2) = 1.2, I(a_3) = 1.3

	$a_2, a_3, a_1,$	$a_4, a_1, a_1,$	$a_2, a_3, a_4,$	$a_2, a_3, a_1,$	$a_6, a_2, a_1,$	$a_2, a_3, a_1,$	$a_2, a_3, a_7,$	GIG
(a_2, *, *)	1.2	-1.2	1.2	1.2	-1.2	1.2	1.2	2.4
(*, a_3, *)	1.3	-1.3	1.3	1.3	1.3	1.3	1.3	5.2
(a_2, a_3, *)	2.5	-1.2 -1.3	2.5	2.5	-1.2	2.5	2.5	6.3

GIG of (a_2, *, *) = – 1.2 + 1.2 + 1.2 – 1.2 + 1.2 + 1.2 = 2.4

GIG of (*, a_3, *) = – 1.3 + 1.3 + 1.3 + 1.3 + 1.3 + 1.3 = 5.2

GIG of (a_2, a_3, *) = (5 – 1) × 2.5 – 1.2 – 1.3 – 1.2 = 6.3

Optimal Information Surplus Given: I(a_2) = 1.1, Period = 3

position	2	5	8	11	14	17	20	23	26
event	a_1 a_2 a_7	a_4 a_9 a_2	a_4 a_2 a_2	a_4 a_9 a_7	a_4 a_2 a_2	a_6 a_9 a_2	a_4 a_2 a_1	a_4 a_9 a_7	a_6 a_6 a_2
	a_2 ⟷ a_2 Distance = 4		a_2 a_2 ⟷	Distance = 5	a_2 a_2 ⟷	a_2	a_2 ⟷	Distance = 7	a_2
loss		-1.1			-1.1				-2.2
gain	1.1	1.1	1.1 1.1		1.1 1.1	1.1	1.1		1.1
OIS	0	0	1.1 2.2		2.2 3.3	4.4	5.5		4.4

If 3 < distance ≤ 6 then loss = -1.1.

If 6 < distance ≤ 9 then loss = -1.1×2 = -2.2

Maximum Information Gain Given: I(a_2) = 1.1, I(a_4) = 1.2, I(a_6) = 1.3, Period = 3

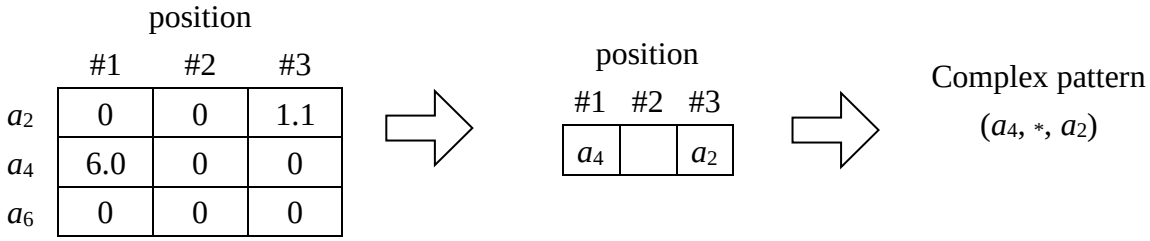
event		$a_1\ a_2\ a_7$	$a_4\ a_9\ a_2$	$a_4\ a_2\ a_2$	$a_4\ a_9\ a_7$	$a_4\ a_2\ a_2$	$a_6\ a_9\ a_2$	$a_4\ a_2\ a_1$	$a_4\ a_9\ a_7$	$a_6\ a_6\ a_2$
OIS	a_1	0						0		
	a_2	0	0	1.1 2.2		2.2 3.3	4.4	5.5		4.4
	a_4		0	1.2	2.4	3.6		4.8	6.0	
	a_6						0			0 1.3
	a_7	0			0				0	
	a_9		0		0		0		0	

MIG(a_2)$^{\text{position \#2}}$ = 1.1×3 = 3.3, MIG(a_2)$^{\text{position \#3}}$ = 1.1×4 = 4.4, MIG(a_2)$^{\text{position \#1}}$ = 0

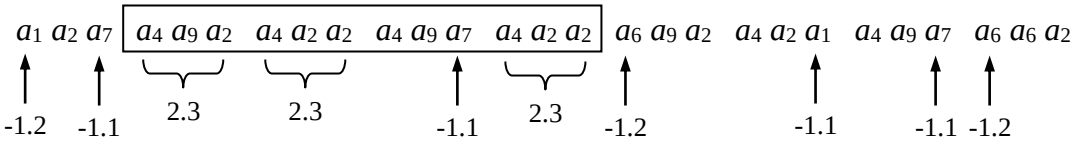
MIG(a_4)$^{\text{position \#1}}$ = 1.2×5 = 6.0, MIG(a_6)$^{\text{position \#1}}$ = 1.3×1 = 1.3, MIG(a_6)$^{\text{position \#2}}$ = 0

Minimum Information Gain = 3.5

MIG counting



Verification of pattern (a_4, *, a_2)



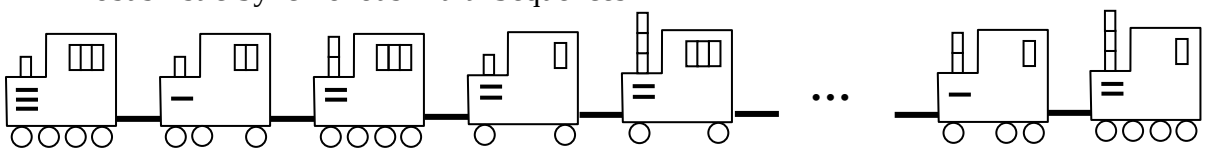
Maximum generalized information gain = 2.3 – 1.1 + 2.3 = 3.5

This algorithm employed penalty associated distance between patterns repeats. So, the significant pattern occurrences will be given a positive GIG while a non-occurrence will generate a negative GIG. The result can specify where the position in sequence.

6. Algorithms for Prediction

- **CWC model** (*Calculation of information gain for prediction*)

Probabilistic Synchronous Multi-Sequences



Identification of relevant values of *Windows and Wheels*

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 numbers in sequence is 19

3-wheel → 3-window (2nd locomotive → 3rd locomotive) Total numbers in sequence is 11

4-wheel → 1-window (3rd locomotive → 4th locomotive) Total numbers in sequence is 12

2-wheel → 3-window (4th locomotive → 5th locomotive) Total numbers in sequence is 7

⋮ ⋮ ⋮

Contingency table for number of windows and number of wheels

	1-window	2-window	3-window	Total
2-wheel	3	4	7	14
3-wheel	7	1	3	11
4-wheel	12	19	2	33
Total	22	24	12	58

e.g. $e_{lk} = 24(33)/58$ $v_{lk} = (1–24/58)(1–33/58)$

$z_{lk} = (19 – e_{lk}) / e_{lk}^{½}$ $d_{lk} = z_{lk} / v_{lk}^{½} = 2.88$

Adjusted Residuals for number of windows and number of wheels

	1-window	2-window	3-window
2-wheel	– 1.46	– 1.12	3.11
3-wheel	1.95	– 2.42	0.60
4-wheel	– 0.28	2.88	– 3.16

Significant selection

Standard normal deviate (5%) = 1.96

Weight of evidence (e.g. 4-wheel → 2-window)

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

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

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

Construction of Rules (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.

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) → 2-window(p+2),	Certainty = 0.91
Rule#6:	medium-funnel(p) → 1-window(p+2),	Certainty = 1.25
Rule#7:	medium-funnel(p) → 2-window(p+2),	Certainty = –1.44
Rule#8:	2-strip(p) → 1-window(p+1),	Certainty = 1.03
Rule#9:	2-strip(p) → 2-window(p+1),	Certainty = –1.01
Rule#10:	1-window(p) → 1-window(p+1),	Certainty = –1.38
Rule#11:	1-window(p) → 2-window(p+1),	Certainty = 1.20
Rule#12:	2-window(p) → 2-window(p+1),	Certainty = –1.42

Prediction of Future Objects

According to the Rule#1, Rule#7, Rule#9, and Rule#11:

W(number of windows = Two / number of windows ≠ Two | medium-funnel, 2-strip, 4-wheel 1-window)

= 0.94 – 1.44 – 1.01 + 1.20 = – 0.31

According to Rule#4:

W(number of windows = Three / number of windows ≠ Three | 4-wheel)

= – 2.02

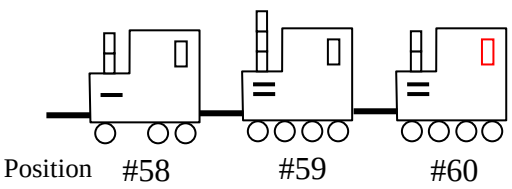
According to the Rule#6, Rule#8, and Rule#10:

W(number of windows = One / number of windows ≠ One | medium-funnel, 2-strip, 1-window)

= 1.25 + 1.03 – 1.38

= 0.90

Hence, one window is in the #60 locomotive. (Prediction result in *Red* colour)



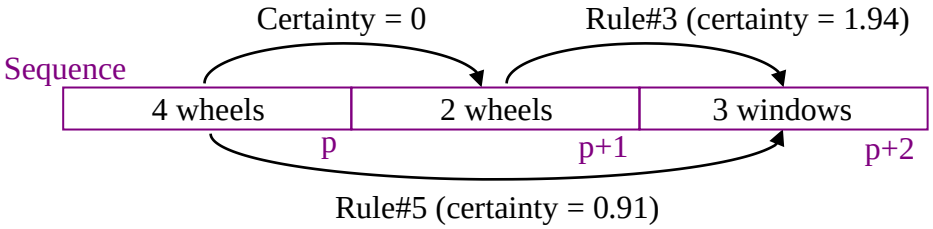
7. Further Enhancements

Inspired by concepts in data mining and dynamical systems, these papers [CWC94] [YWY03] could be introduces the new methods for identifying sequential patterns in time series that are significant for characterizing and predicting events, i.e., the important occurrences. The rule-based CWC model could be find a significant pattern, and YWY model just rearranged the sequence from single to multiple, but it should be changed to time series nature. The methodology as following:

- Sequential Pattern Mining

According to paper [CWC94] pp.1539, the rules #1 to #12, and p is position of object.

Rule#1: 4 wheels(p) → 2 windows(p+1), certainty = 0.94
Rule#2: 3 wheels(p) → 2 windows(p+1), certainty = –2.82
Rule#3: 2 wheels(p) → 3 windows(p+1), certainty = 1.94
Rule#4: 4 wheels(p) → 3 windows(p+1), certainty = –2.02
Rule#5: 4 wheels(p) → 3 windows(p+2), certainty = 0.91
⋮ ⋮ ⋮
Rule#12: 2 wheels(p) → 3 windows(p+1), certainty = –1.42

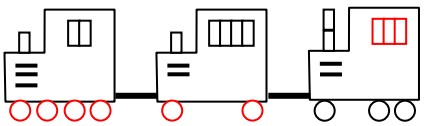


Using the same criteria and algorithm of CWC model to find the sequential pattern.

The pattern should be found by maximum of certainties = 0 + 0.91 + 1.94 = 1.85

Hence, the sequential pattern is (4 wheels → 2 wheels → 3 windows).

Pattern in diagram:

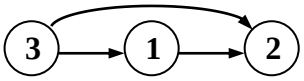


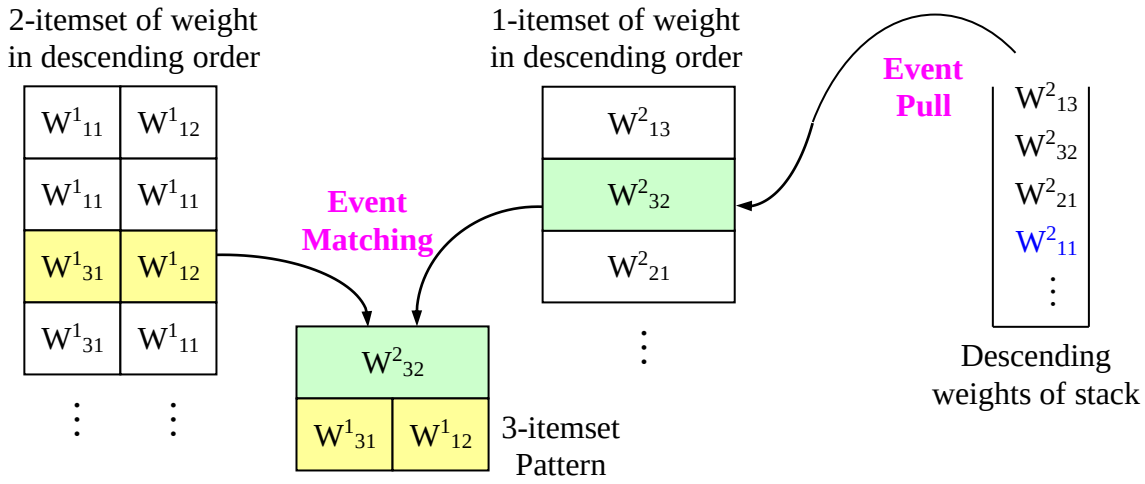
where Don't care in *Black* colour
Sequential pattern in *Red* colour

This pattern is significant, but may not be the frequent subsequence.

- Optimal Weight of Evidence Algorithm

Let n and m be the maximum numbers of rules of one-position (p+1) and two-position (p+2) respectively. Hence, maximum size of 3-itemset of candidate sequence = m × n^2 = n^2(n–1). Two sets of rule are W^1_{11} to W^1_{nn}, and W^2_{11} to W^2_{mm}. Line up 2-itemset of (p+1) in descending order, and store (p+2) rules descending in a stack. The maximum weight of certainty could be obtained, until the stack is empty **OR** in case $W^1_{31} + W^1_{12} - W^1_{11} - W^1_{12} < W^2_{32} - W^2_{11}$

e.g. Events matched 



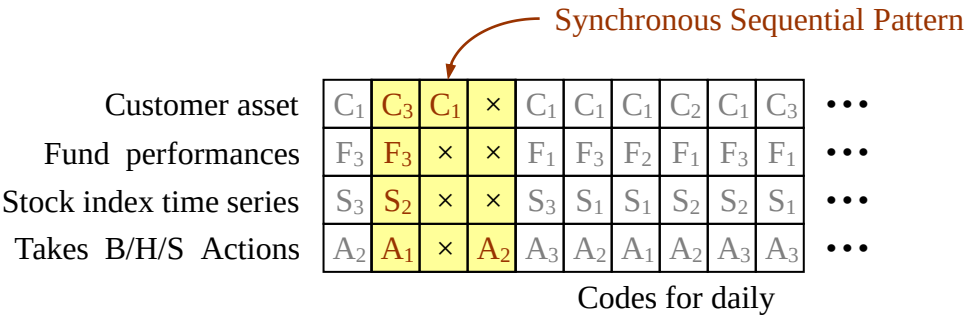
- Synchronous and Sequential Pattern

Using the previous example in chapter 4, suppose a sequential pattern ($S_2 \rightarrow C_1 \rightarrow A_2$) is obtained by sequence mining. According to following multi-sequence, the result sequential pattern means “Last day stock index down less than 1%, and the customer has less then $10000 today, he will sell the product tomorrow.”. The sequential pattern could be occurred anywhere inside the multi-sequence, so it can be merge another sequential pattern interconnect with a same event by themselves only.

Customer asset	C_1	C_3	C_1	C_2	C_1	C_1	C_1	C_2	C_1	C_3	•••
Fund performances	F_3	F_3	F_2	F_2	F_1	F_3	F_2	F_1	F_3	F_1	•••
Stock index time series	S_3	S_2	S_1	S_2	S_3	S_1	S_1	S_2	S_2	S_1	•••
Takes B/H/S Actions	A_2	A_1	A_3	A_2	A_3	A_2	A_1	A_2	A_3	A_3	•••

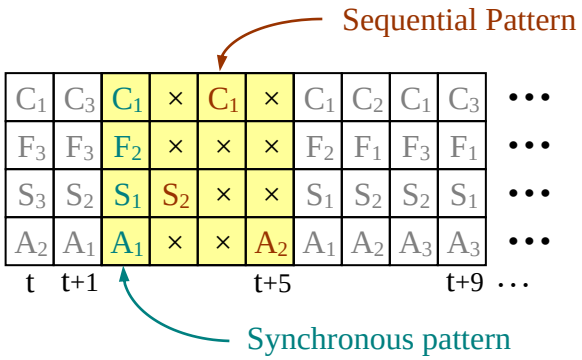
Codes for daily

Meanwhile, using the same example, a 4-tuple supervised pattern (C_3, F_3, S_2, A_1) was obtained by patterns mining of [YWY03a]. The pattern means that "While stock index (Down > 1%/day) and the fund performs (5%Down/wk ≥ Price ≤ 5%Up/wk), the customer buys product, who has ($10000≤ Saving ≤$20000).".



If two patterns can be merged together as above, mining result is that " Today, stock index is (Down > 1%/day) and the fund performs (5%Down/wk ≥ Price ≤ 5%Up/wk), the customer spends less than $10000 in next day. He will sell the product two days later.". This is an optimal synchronous multi-sequence pattern, which consists of don't care events in the pattern. The pattern is significant but not frequent pattern.

If two patterns cannot be merged by at least one event in own sequence, it is not a synchronous multi-sequence pattern. There should be separated into two patterns. One is sequential pattern. Another is synchronous pattern. They cannot be mixed up together, unless cross-merge is occurred between sequential pattern and synchronous pattern.



- Type of Sequence / data

The data is characterized by the fact that it can be written down as a long sequence of codes. Sequence algorithms occur in many areas of science and information processing. It could be said that it is the domain in which practice and theory are very close to each other. Therefore, the synchronous multi-sequence used to correspond to time series information and, no doubt, this can solving problems in many application systems.

8. Comparative Studies of models

- Edit distance

However, to begin of models, they do not apply inexact matching, although meta-pattern model has editing distance features. In practical pattern-matching applications, the exact matching is not always pertinent. It is often more important to find objects that match a given pattern in a reasonably approximate way. Algorithms are mainly based on the algorithmic method called dynamic programming. At this point, algorithms shift from the general area of exact matching and exact pattern discovery to the general area of approximate matching. But usually organized as database search, exact matching problems that arise as sub-problems in multiple sequence comparison, in large-scale sequence comparison, in database searching, and in other importance applications.

- Comparative Performance of modeling

Criteria	**CWC model**	**YWY model**
Probabilistic Sequence Type	Synchronous	Asynchronous
Sequence Profile	Multiple sequences	Single sequence
Calculation of Information Gain	Weight of evidence	Distance counting
Mining Targets	Significant Pattern	Surprising Pattern
Construction in mining process	Rule-based	Apriori Property
Format of Pattern	Multi-profile	Fixed Period
Threshold Bounded	Chi-square 5%	Information gain
Scheme of Algorithms	OBSERVER-II	InfoMiner, STAMP
Mining Limitation	Singular Profile	Predefined Length
Constraint	Short Pattern	Small *min_rep*, *max_dis*
Applications	Widespread	Specify Area (e.g. DNA)

9. Conclusions

There are several significant features of the proposed method in future. First, the method focuses on the identification of the temporal patterns that are characteristic of the events. Second, with the temporal patterns identified, the new method focuses on event prediction rather than complete time series prediction. This allows the prediction of complicated time series events such as the customer behaviour in e-banking. Third, the objective function in the optimization reflects the goal of the time series being examined, i.e., customer behaviour, and is problem specific.

Although the algorithms proposed by papers are good in simulation, the models should be applied in real situation, for example, the order of events in the sequence would be disrupted by noises, and how to cope with the compound effect of multiple noise types. Various algorithms have been proposed in this report for the problem of additionally finding optimal editing multi-sequence. Certain other approaches that may be implemented a modify method is capable of characterizing sequential patterns of complex time series, which are often non-periodic, irregular, and chaotic. This method identifies predictive sequential structures in reconstructed phase spaces. Hence, in practical pattern matching application, the exact matching is not always pertinent. It is often more important to find objects that match a given pattern in a reasonably approximate way, so-called ***Approximate Matching Mining***. For example, one very important case where simple wild cards occur is in *DNA transcription factors*. A transcription factor is a protein that binds to specific locations in DNA and regulates, either enhancing or suppressing, the transcription of the DNA into RNA.