Periodic pattern mining

Alexandre Termier

Université de Rennes 1

Equipe Inria/IRISA Lacodam

Data Mining and Visualization course – M2 SIF

These slides adapted from an invited talk at [SFC 2019]

Motivation

- Pattern mining : finding regularities in data

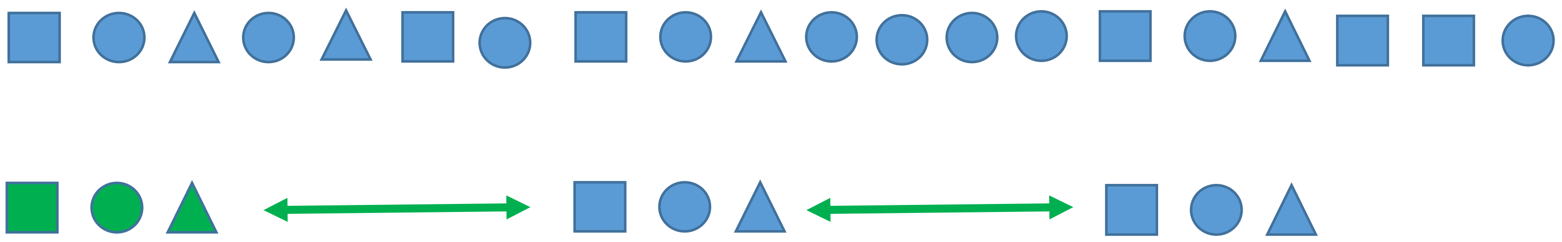
- « Habits »
  - Regularity in the actions performed
  - Temporal regularity between occurrences

- Different problem for pattern miner
  - WHAT is repeated => HOW is it repeated

Periodicity

- Pattern P is reapeated (as usual) => has occurrences
- Some temporal property between occurrences
  - Sequencing
  - Timestamps

- Periodicity (naïve version) : constant inter-occurrence delay



This talk

Several approaches on periodic/near periodic pattern mining

- Condensed representation for mining periodic pattern with gaps
- Nested periodic pattern mining with MDL
- « Signature » patterns

How periodic is your set-top box?
Analyzing the execution of a video decoder

Patricia López Cueva, Aurélie Bertaux, Alexandre Termier, Jean-François Méhaut, Miguel Santana: *Debugging embedded multimedia application traces through periodic pattern mining*. EMSOFT 2012: 13-22

Particia López Cueva, *Debugging Embedded Multimedia Application Execution Traces through Periodic Pattern Mining*, PhD, 2013.

Slides adapted from Patricia Lopez Cueva

Context

- Data : execution traces of set-top boxes
  - System level info : interrupts, context switches,…
  - Applicative info : start/end of (some) high level functions
  - Application : video decoding

- Problem :
  - Understand complex periodic behavior of video decoding software
  - Determine when the periodicity is broken

Data

Execution trace = Sequence of timestampped events

Cut into windows

Transform into sequence of itemsets (window -> itemset)

Execution Trace ($s.\mu s$)

0.1 ms

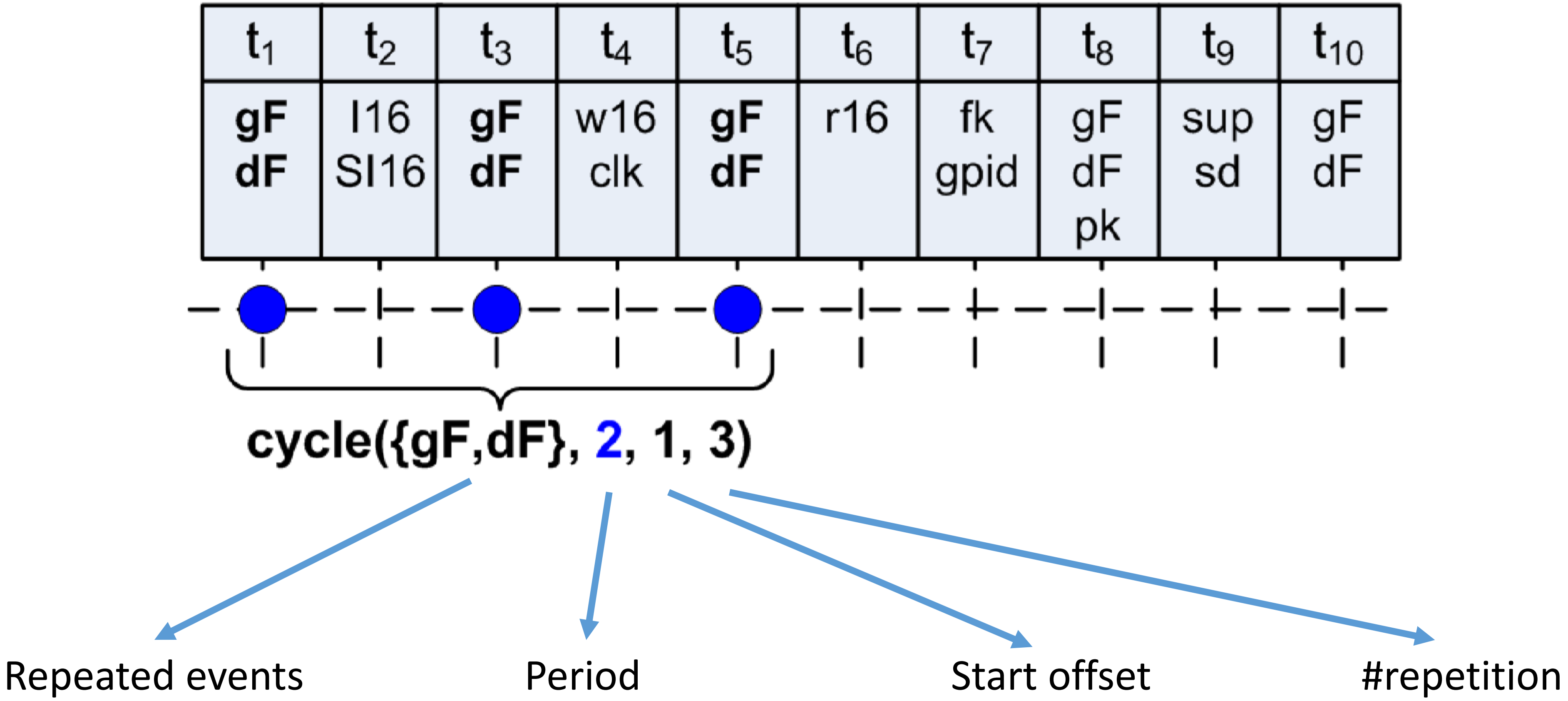
68.770630	getFrame
68.770697	displayFrame
68.770741	int16
68.770768	swint16
68.770869	displayFrame
68.770913	getFrame
68.770959	write16
68.770982	cpu_clock
68.771032	getFrame
68.771099	displayFrame
68.771150	read16
68.771235	fork
68.771324	get_pid
68.771346	getFrame
68.771372	displayFrame
68.771402	printk
68.771456	sem_up
68.771487	sem_down
68.771540	getFrame
68.771586	displayFrame

Preprocessing

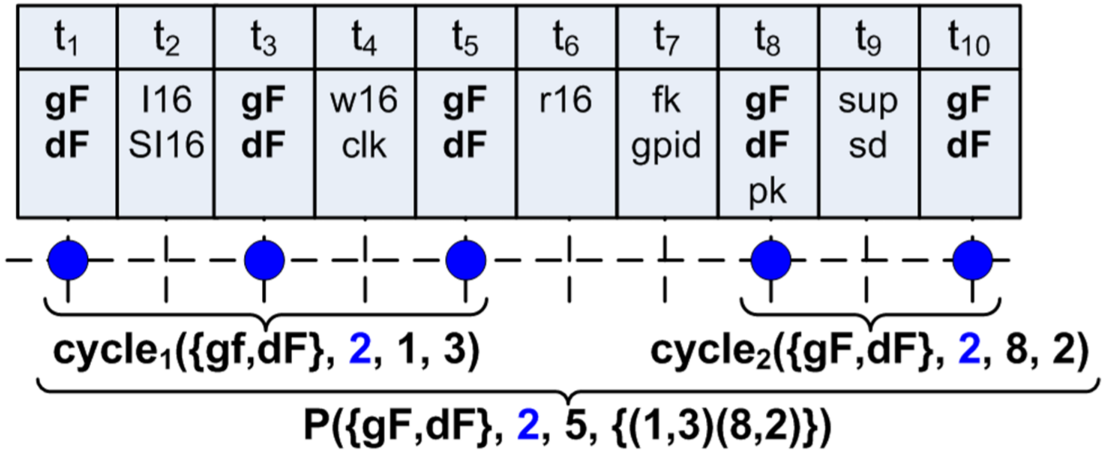
Transactional Database

t_1	getFrame, displayFrame
t_2	int16, swint16
t_3	displayFrame, getFrame
t_4	write16, cpu_clock
t_5	getFrame, displayFrame
t_6	read16
t_7	fork, get_pid
t_8	getFrame, displayFrame, printk
t_9	sem_up, sem_down
t_{10}	getFrame, displayFrame

Pattern building block : the **cycle**



Periodic pattern



Periodic Pattern [Ma & Hellerstein, 2001]

A group of cycles forms a periodic pattern if:

1. Same period for all cycles.
2. All cycles are consecutive.
3. Cycles do not overlap.

Support

Sum of all *cycles* lengths:

$$cycles = \{(o_1, l_1), \ldots, (o_k, l_k)\}$$

$$support = \sum_{i=1}^{k} l_i$$

Many redundancies

Frequent Periodic Pattern

Given a minimum support threshold (min_sup), a pattern is frequent if

$$support \geq min\_sup$$

Redundancies in periodic patterns defined

1. All subsets of the itemset part
2. Combinations / multiples of the period

Frequent Periodic Patterns
$P_1(\{gF\}, 2, 5, \{(1,3)(8,2)\})$
$P_2(\{dF\}, 2, 5, \{(1,3)(8,2)\})$
$P_3(\{gF, dF\}, 2, 5, \{(1,3)(8,2)\})$
...
$P_6(\{gF, dF\}, 3, 2, \{(5,2)\})$
...
$P_9(\{gF, dF\}, 4, 2, \{(1,2)\})$
...
$P_{12}(\{gF, dF\}, 5, 2, \{(3,2)\})$
...
$P_{15}(\{gF, dF\}, 5, 2, \{(5,2)\})$

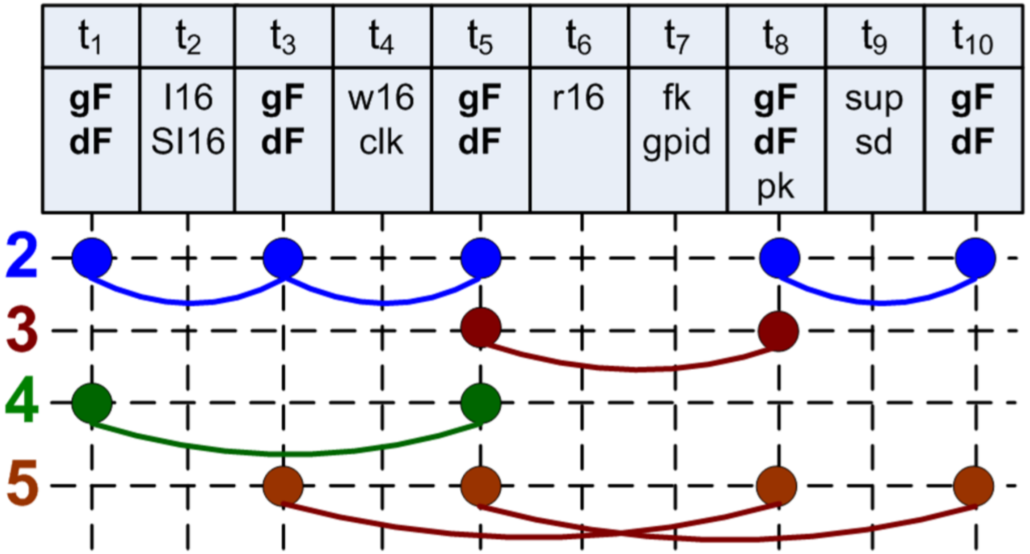
Towards a condensed representation

- Too many redundant patterns -> condensed representation
  - Closed periodic patterns ?

- Pb : cannot compute classic closure with (Itemset, Period, Transactions)

- Solution : move from diadic to triadic !
  - Based on a *ternary relation*

Triadic representation



	Periods	2										3									
Itemsets	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

	Periods	4										5									
Itemsets	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

Itemsets	Periods	2										3									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

Itemsets	Periods	4										5									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

Triples
$(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5\})$

	Periods	2										3									
Itemsets	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

	Periods	4										5									
Itemsets	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

Periodic Concepts
$T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$

Itemsets	Periods	2										3									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

Itemsets	Periods	4										5									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

Periodic Concepts
$T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$
$T_2(\{gF, dF\}, \{2, 4\}, \{t_1, t_5\})$

Itemsets	Periods	2										3									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

Itemsets	Periods	4										5									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

Periodic Concepts
$T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$
$T_2(\{gF, dF\}, \{2, 4\}, \{t_1, t_5\})$
$T_3(\{gF, dF\}, \{2, 5\}, \{t_3, t_5, t_8, t_{10}\})$

	Periods	2										3									
Itemsets	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

	Periods	4										5									
Itemsets	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

Periodic Concepts
$T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$
$T_2(\{gF, dF\}, \{2, 4\}, \{t_1, t_5\})$
$T_3(\{gF, dF\}, \{2, 5\}, \{t_3, t_5, t_8, t_{10}\})$
$T_4(\{gF, dF\}, \{2, 3, 5\}, \{t_5, t_8\})$

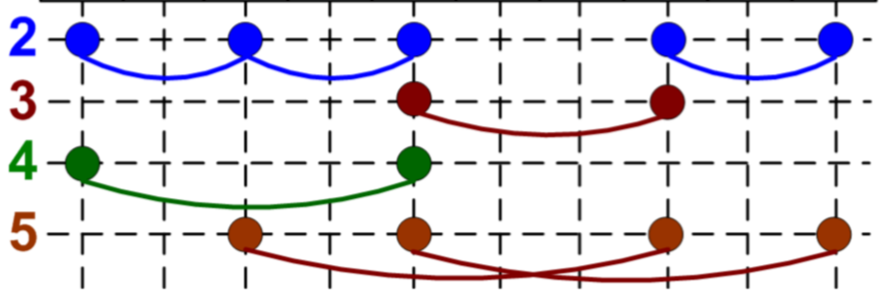
Core Periodic Concept [EMSoft 2012]

Core Periodic Concept

A periodic concept (I, P, T) is a **core periodic concept** if there does not exist any other periodic concept (I', P', T') such that $I = I'$, $P' \subset P$ and $T' \supset T$.

Periodic Concepts
$T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$
$T_2(\{gF, dF\}, \{2, 4\}, \{t_1, t_5\})$
$T_3(\{gF, dF\}, \{2, 5\}, \{t_3, t_5, t_8, t_{10}\})$
$T_4(\{gF, dF\}, \{2, 3, 5\}, \{t_5, t_8\})$

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF **dF**	l16 Sl16	**gF** **dF**	w16 clk	**gF** **dF**	r16	fk gpid	**gF** **dF** pk	sup sd	**gF** **dF**

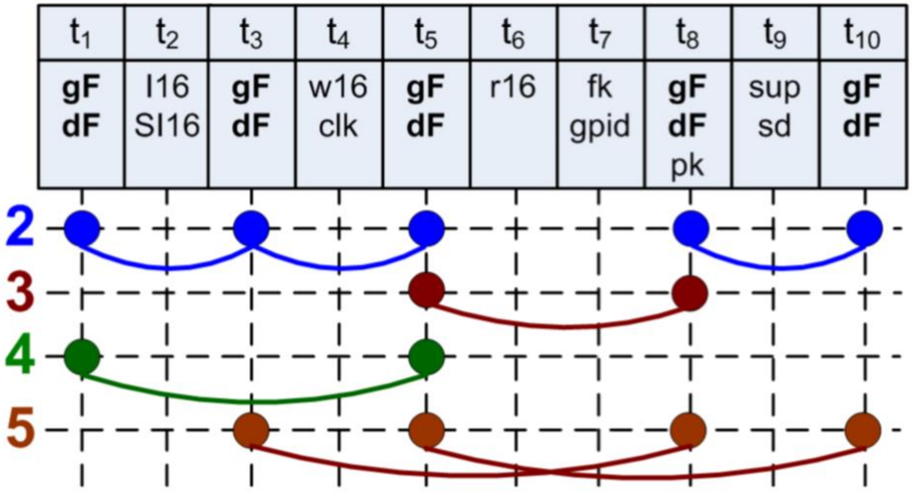


Core Periodic Concept [EMSoft 2012]

Core Periodic Concept

A periodic concept (I, P, T) is a **core periodic concept** if there does not exist any other periodic concept (I', P', T') such that $I = I'$, $P' \subset P$ and $T' \supset T$.

Core Periodic Concepts
$T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$
$T_2(\{gF, dF\}, \{2, 4\}, \{t_1, t_5\})$
$T_3(\{gF, dF\}, \{2, 5\}, \{t_3, t_5, t_8, t_{10}\})$
$T_4(\{gF, dF\}, \{2, 3, 5\}, \{t_5, t_8\})$

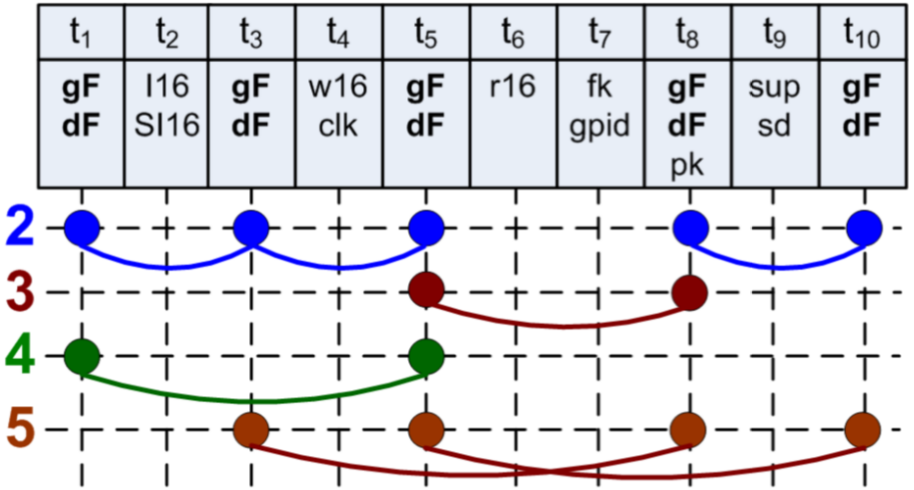


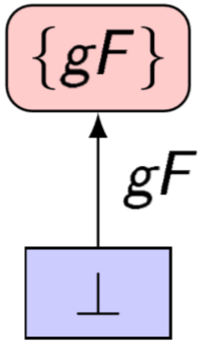
CPC = condensed representation of all periodic concepts

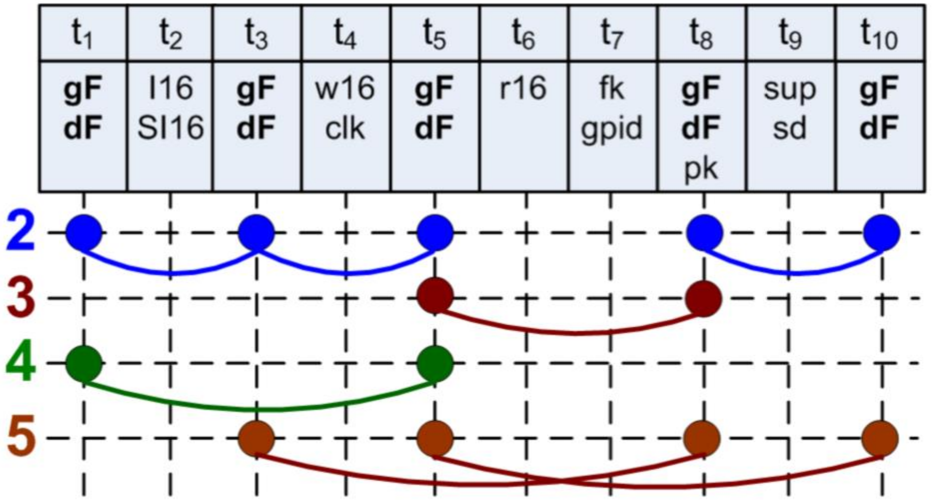
Mining Core Periodic Concepts

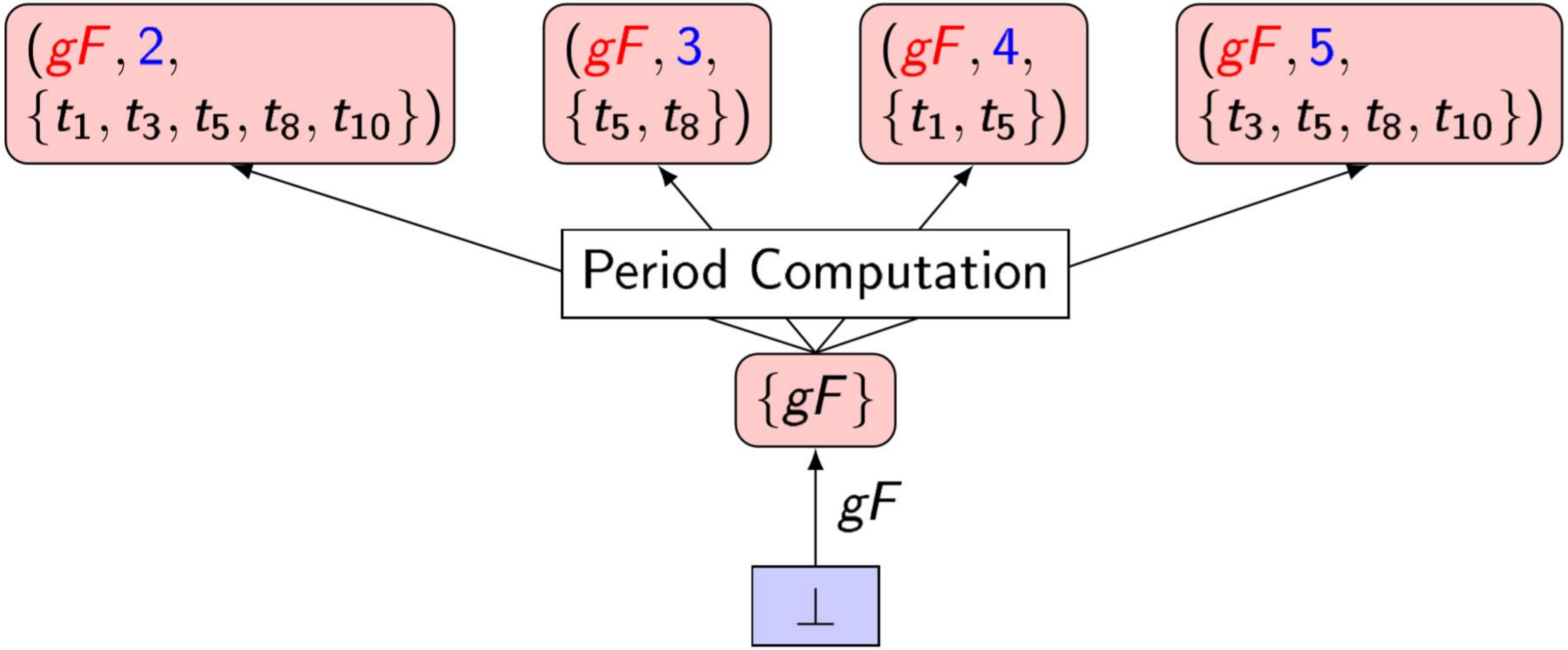
- Solution 1: [EMSoft 2012]
  - Use DataPeeler (Cerf et al., 2009) to get triadic patterns
  - Postprocess to filter CPC

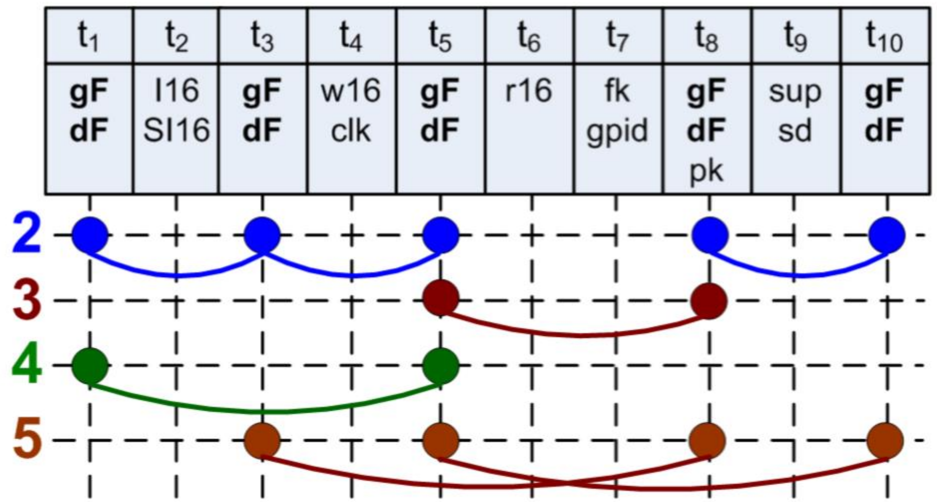
- Solution 2: [López Cueva PhD, 2013]
  - Direct mining of CPC
  - Based on LCM/CbO enumeration strategy
  - Proven poly-delay time, poly space


t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
gF dF
l16 Sl16
gF dF
w16 clk
gF dF
r16
fk gpid
gF dF pk
sup sd
gF dF
2
3
4
5

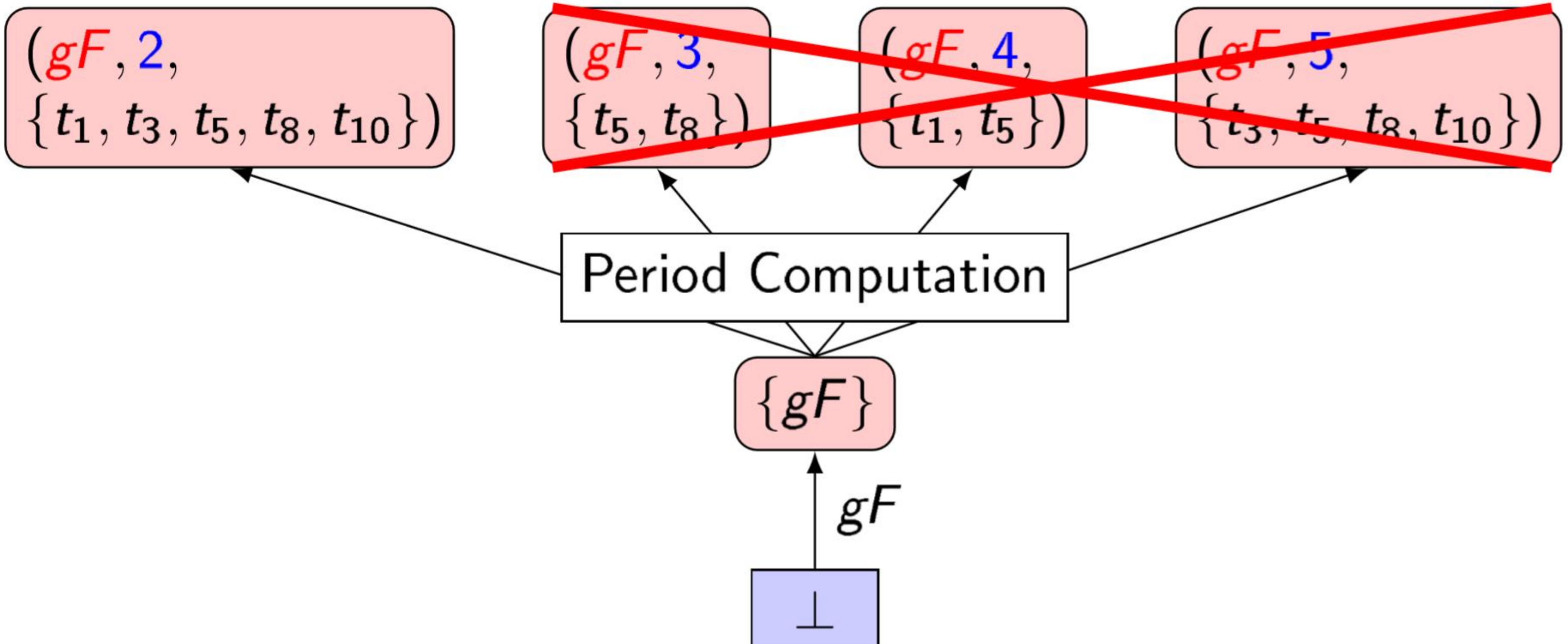

{gF}
gF
⊥


t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
gF dF
l16 Sl16
gF dF
w16 clk
gF dF
r16
fk gpid
gF dF pk
sup sd
gF dF
2
3
4
5

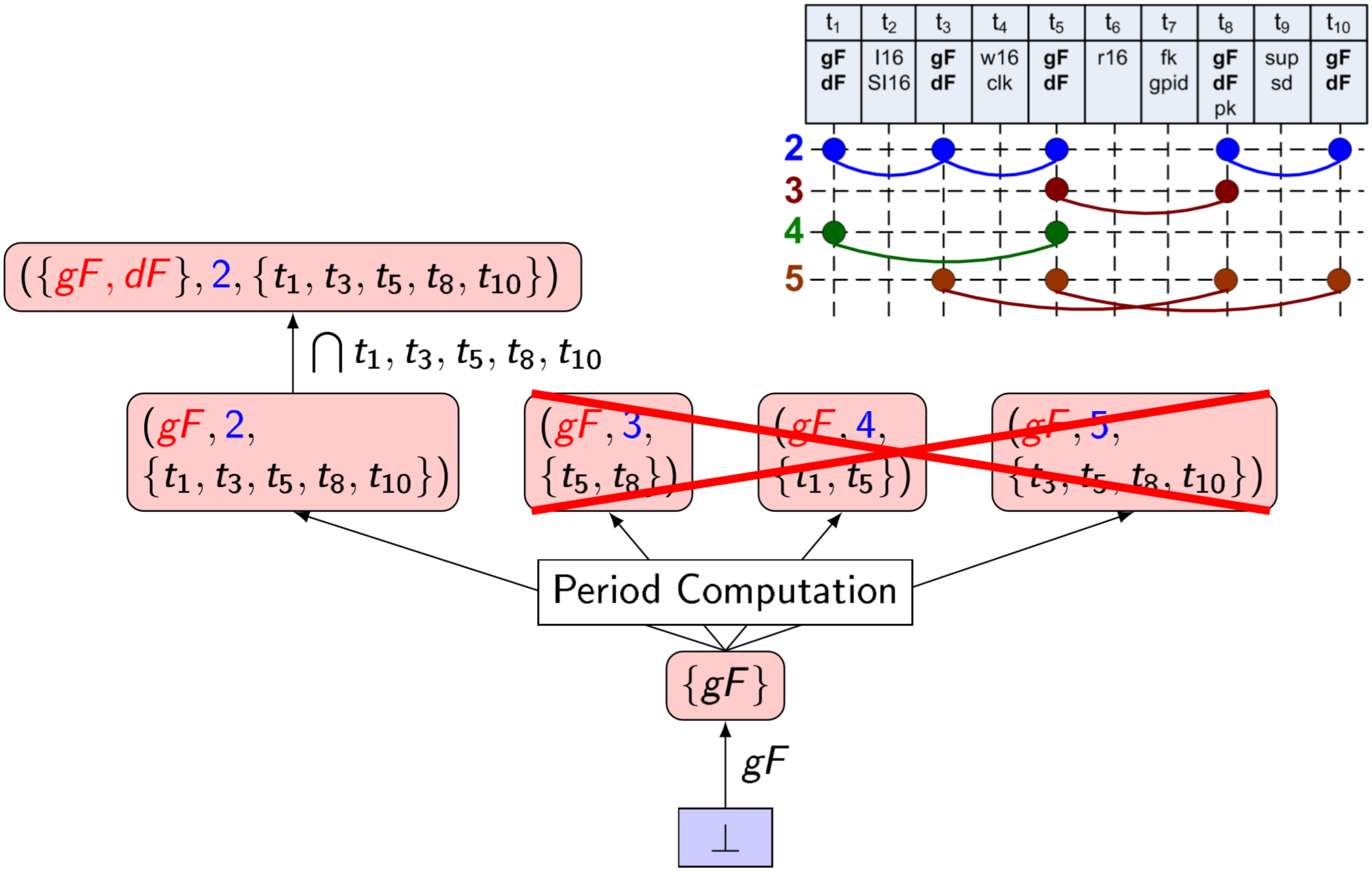

$(gF, 2, \{t_1, t_3, t_5, t_8, t_{10}\})$
$(gF, 3, \{t_5, t_8\})$
$(gF, 4, \{t_1, t_5\})$
$(gF, 5, \{t_3, t_5, t_8, t_{10}\})$
Period Computation
$\{gF\}$
gF
$\perp$

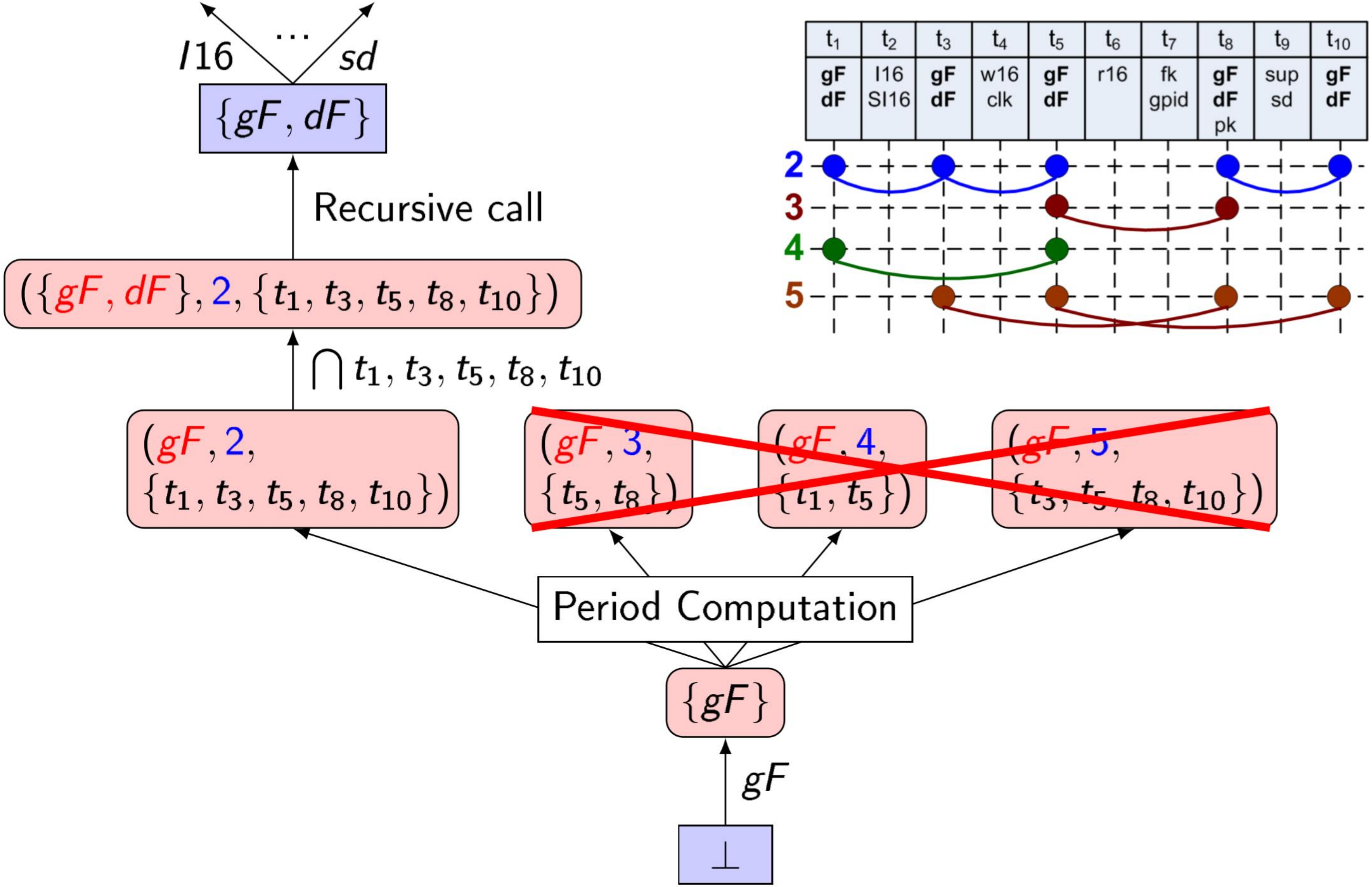


t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
gF dF
l16 Sl16
gF dF
w16 clk
gF dF
r16
fk gpid
gF dF pk
sup sd
gF dF
2
3
4
5



(gF, 2, {t1, t3, t5, t8, t10})
(gF, 3, {t5, t8})
(gF, 4, {t1, t5})
(gF, 5, {t3, t5, t8, t10})
Period Computation
{gF}
gF
⊥


t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
gF dF
l16 Sl16
gF dF
w16 clk
gF dF
r16
fk gpid
gF dF pk
sup sd
gF dF
2
3
4
5
({gF, dF}, 2, {t1, t3, t5, t8, t10})
⋂ t1, t3, t5, t8, t10
(gF, 2, {t1, t3, t5, t8, t10})
(gF, 3, {t5, t8})
(gF, 4, {t1, t5})
(gF, 5, {t3, t5, t8, t10})
Period Computation
{gF}
gF
⊥



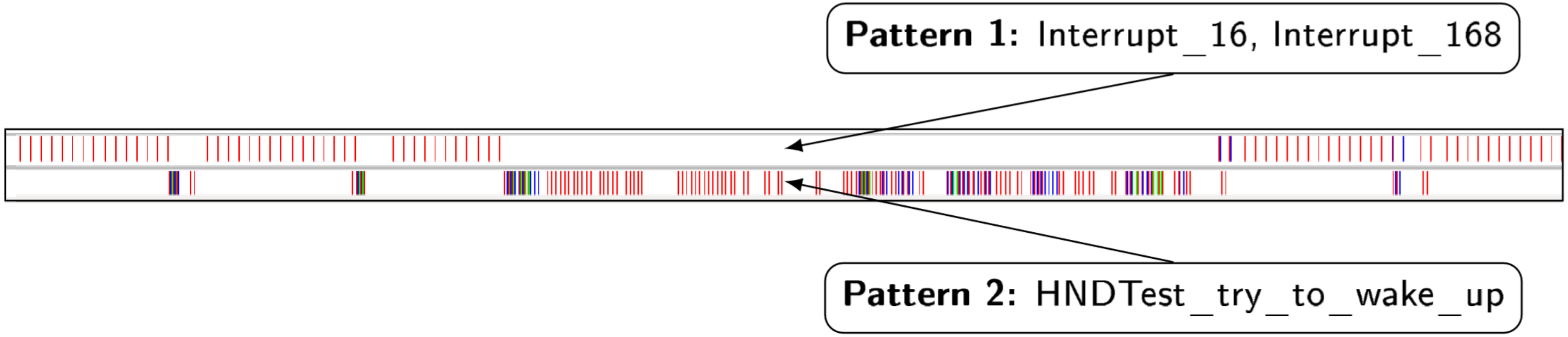
l16 … sd
{gF, dF}
Recursive call
({gF, dF}, 2, {t1, t3, t5, t8, t10})
⋂ t1, t3, t5, t8, t10
(gF, 2, {t1, t3, t5, t8, t10})
(gF, 3, {t5, t8})
(gF, 4, {t1, t5})
(gF, 5, {t3, t5, t8, t10})
Period Computation
{gF}
gF
⊥
t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
gF dF
l16 SI16
gF dF
w16 clk
gF dF
r16
fk gpid
gF dF pk
sup sd
gF dF
2
3
4
5

Trace of STi7200 stb
500k events
-> 13k transactions, ~8 items/transac
Mining 10% -> 195s
758 CPC (20k per pats)

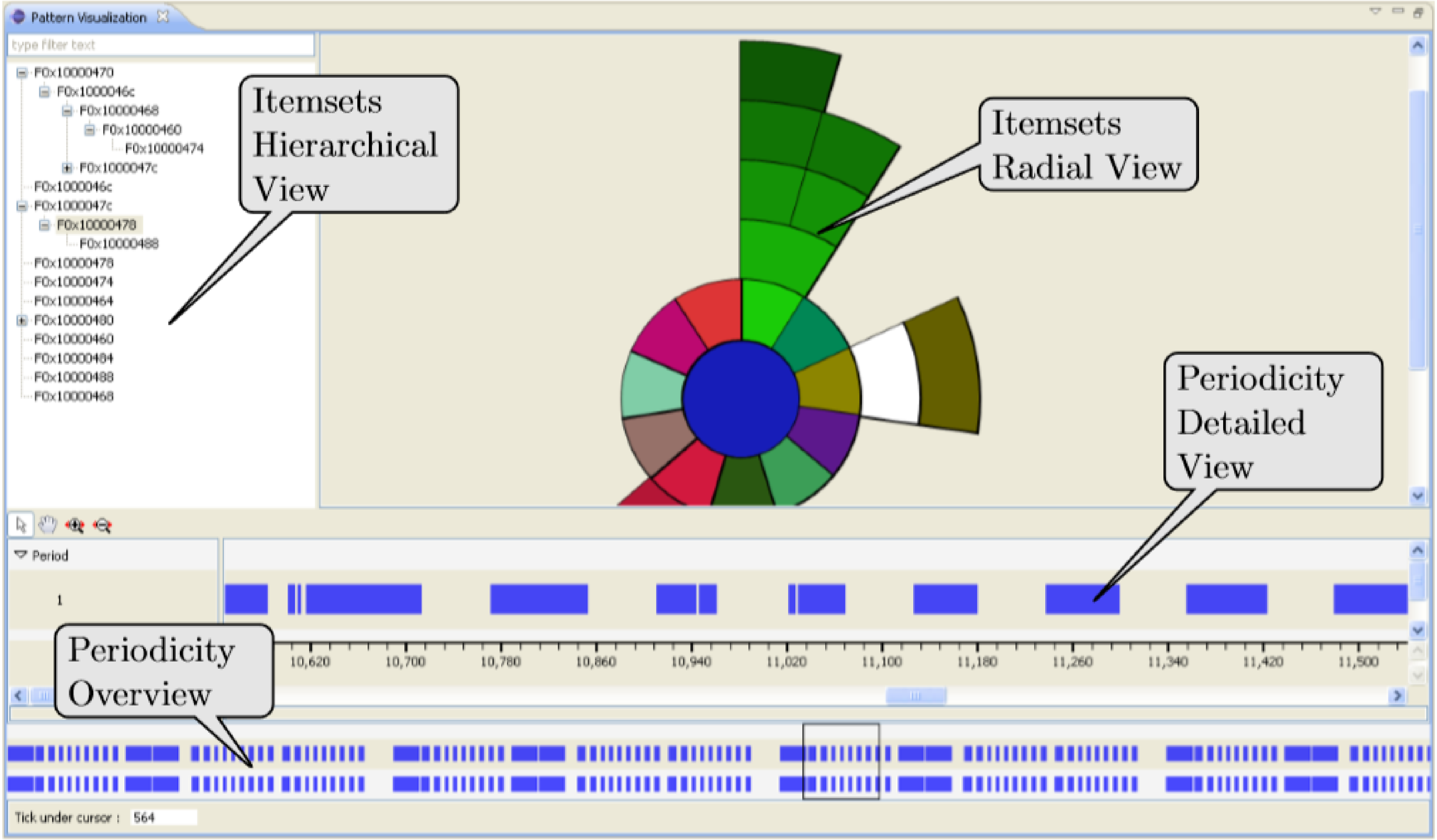
Application on real execution trace

Discovered conflict between the application and the system (USB port)

- `Interrupt_16`: processor clock interrupt.
- `Interrupt_168`: USB interrupt.
- `HNDTest_try_to_wake_up`: system call (`try_to_wake_up`).



A visualization of CPC



How periodic are we?

Analyzing the life of Sacha

Esther Galbrun, Peggy Cellier, Nikolaj Tatti, Alexandre Termier, Bruno Crémilleux:
Mining Periodic Patterns with a MDL Criterion. ECML/PKDD (2) 2018: 535-551

Slides adapted from Peggy Cellier

Motivations

- Previous work: we wanted few patterns
- With CPC, we still have 1k patterns…

- H. Arimura, at the defense of Patricia Lopez Cueva:
  - « *Periodic patterns should compress well the data* »

- => Periodic patterns + MDL *à la Krimp*?
  - Should give fewer patterns
  - With good representativity of the data

http://quantifiedawesome.com/records

Motivation, #2: data of Sacha Chua

Download as spreadsheet

Start	End	Category	Duration	Data
02 Sep 18:44		Discretionary » Play » Other		
02 Sep 18:44	18:44	Sleep	0:00	
02 Sep 12:45	18:44	A- » Childcare	5:59	
02 Sep 12:12	12:45	Personal » Routines	0:32	
02 Sep 12:12	12:12	A- » Childcare	0:00	
02 Sep 12:11	12:12	Personal » Routines	0:00	
02 Sep 11:45	12:11	A- » Childcare	0:26	
02 Sep 10:12	11:45	A- » Childcare	1:33	
02 Sep 09:12	10:12	A- » Childcare	0:59	
02 Sep 08:04	09:12	Personal » Routines	1:07	
01 Sep 23:02	02 Sep 08:03	Sleep	9:00	

Simple periodic pattern in activity trace

$S = \langle$ (**16-04-2018 7:30**, **wake up**), $\leftarrow \#1$
(16-04-2018 7:40, prepare coffee),
...
(16-04-2018 8:10, take metro),
...
(16-04-2018 11:00, attend meeting),
...
(16-04-2018 11:00, eat dinner),
...
(**17-04-2018 7:32**, **wake up**), $\leftarrow \#2$
(17-04-2018 7:38, prepare coffee),
...
(**20-04-2018 7:28**, **wake up**), $\leftarrow \#5$
(20-04-2018 7:41, prepare coffee),
...
(15-06-2018 7:28, wake up),
...$\rangle$

16-04-2018 7:30, wake up

repeat every 24 hours for 5 days

Cycle, again

On April 16, at 7:30 AM, wake up,
repeat every 24 hours for 5 days

A cycle is specified by:

event α: the repeating event,

length r: the number of repetitions of the event,

period p: the inter-occurrence distance,

starting point τ: the timestamp of the first occurrence, and

shift corrections E: a list of time offsets.

Hence, a cycle is a 5-tuple $C = (\alpha, r, p, \tau, E)$.

Noise tolerance

Tolerate variation in inter-occurrence distances,
shift corrections $E = \langle e_1, \dots, e_{r-1} \rangle$.

Reconstruct occurrences timestamps of repetitions recursively:

$$
\begin{aligned}
t_1 &= \tau, \\
t_2 &= t_1 + p + e_1, \\
&\dots \\
t_r &= t_{r-1} + p + e_{r-1}.
\end{aligned}
$$

Problem statement v1

- Input
  - An event sequence
- Output
  - A representative collection of cycles

Introducing *cycle cover*

Denote as $\mathit{cover}(C)$ the corresponding set of reconstructed timestamp–event pairs:

$$\mathit{cover}(C) = \{(t_1, \alpha), (t_2, \alpha), \ldots, (t_r, \alpha)\} \ ,$$

and for a collection $\mathcal{C}$ of cycles

$$\mathit{cover}(\mathcal{C}) = \bigcup_{C \in \mathcal{C}} \mathit{cover}(C) \ .$$

For a sequence S and cycle collection $\mathcal{C}$ we call **residual** the timestamp–event pairs of S not covered by any cycle in $\mathcal{C}$:

$$\mathit{residual}(\mathcal{C}, S) = S \setminus \mathit{cover}(\mathcal{C}) \ .$$

Problem statement v2

We associate

- a cost $L(o)$ to each individual occurrence
- a cost $L(C)$ to each cycle

Then, we can reformulate our problem as follows:

Problem

Given an event sequence S, find the collection of cycles $\mathcal{C}$ minimising the cost

$$L(\mathcal{C}, S) = \sum_{C \in \mathcal{C}} L(C) + \sum_{o \in residual(\mathcal{C}, S)} L(o) \; .$$

What cost?

- Many possible choices for *cost*

- Our cost: based on the MDL principle (Grünwald, 2007)
  - Comes from Information Theory

  - Based on compression
  « *more representative structures allow better compression of data* »

  - Good results in **model selection**…

  - …especially for pattern mining!
    - cf works of Vreeken, van Leeuwen, Siebes, Tatti…

Alignement of MDL and our problem

- Classic MDL formula

L(Data, Model) = L(Model) + L(Model | Data)

*where L(…) = description length in bits - called **encoding***

- Our problem

$$L(\mathcal{C}, S) = \sum_{C \in \mathcal{C}} L(C) + \sum_{o \in residual(\mathcal{C}, S)} L(o)$$

- Paper explains our encoding L(…) in detail

More complex patterns

$$
\begin{aligned}
S = \langle\ & (\text{16-04-2018 } 7{:}30,\ \text{wake up}), \leftarrow \#1 \textbf{ - 1st week}\\
& (\text{16-04-2018 } 7{:}40,\ \text{prepare coffee}),\\
& \ldots\\
& (\text{16-04-2018 } 8{:}10,\ \text{take metro}),\\
& \ldots\\
& (\text{16-04-2018 } 11{:}00,\ \text{attend meeting}),\\
& \ldots\\
& (\text{16-04-2018 } 11{:}00,\ \text{eat dinner}),\\
& \ldots\\
& (\text{17-04-2018 } 7{:}32,\ \text{wake up}), \leftarrow \#2\\
& (\text{17-04-2018 } 7{:}38,\ \text{prepare coffee}),\\
& \ldots\\
& (\text{20-04-2018 } 7{:}28,\ \text{wake up}), \leftarrow \#5\\
& (\text{20-04-2018 } 7{:}41,\ \text{prepare coffee}),\\
& \ldots\\
& (\text{15-06-2018 } 7{:}28,\ \text{wake up}), \leftarrow \#5 \textbf{ - 9th week}\\
& \ldots\rangle
\end{aligned}
$$

16-04-2018 7:30, wake up

10 min later, prepare coffee

repeat every 24 hours for 5 days

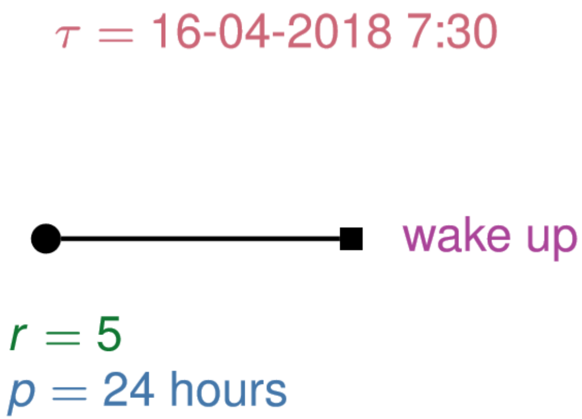
repeat every 7 days for 3 months

Nested cycles

Tree structure

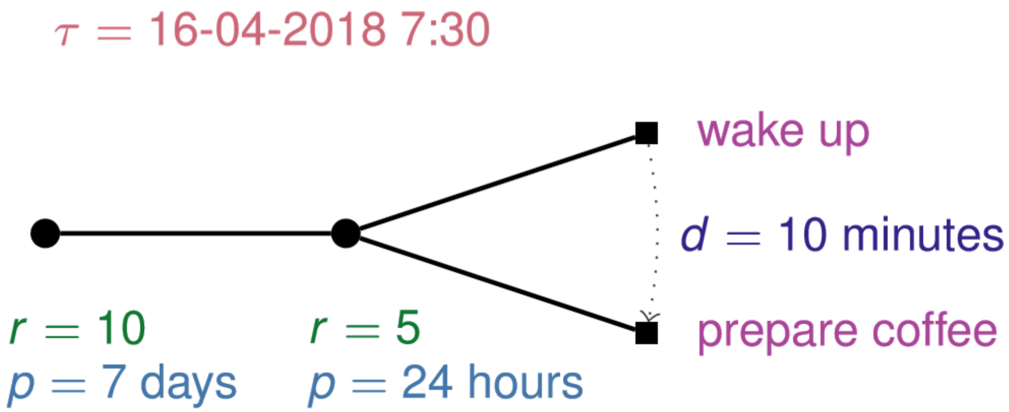
Tree representation

On April 16, at 7:30 AM, wake up,
repeat every 24 hours for 5 days



Tree representation

On April 16, at 7:30 AM, wake up,
10 minutes later, prepare coffee,
repeat every 24 hours for 5 days,
repeat this every 7 days for 3 months



Update to problem statement

- Problem statement updated: cycles -> patterns

Problem

Given an event sequence S, find the collection of patterns $\mathcal{P}$ minimising the cost

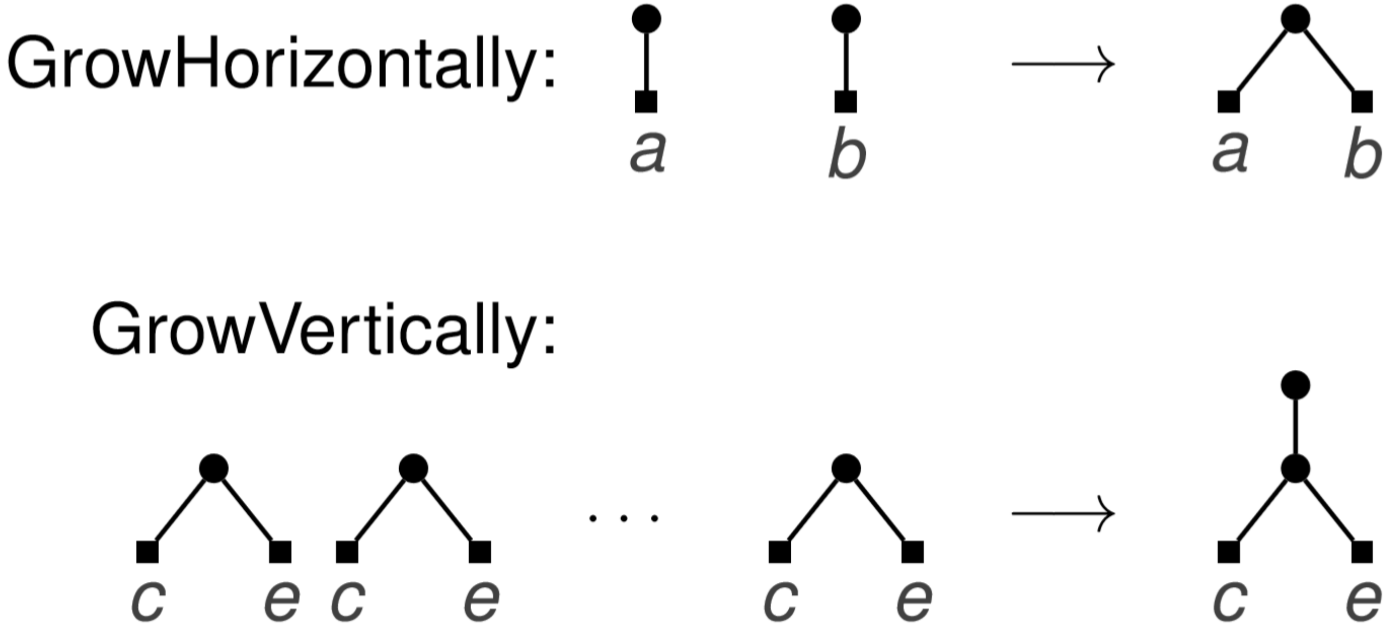
$$L(\mathcal{P}, S) = \sum_{P \in \mathcal{P}} L(P) + \sum_{o \in residual(\mathcal{P}, S)} L(o) \ .$$

- Encoding L(…) defined for the tree patterns

Algorithm 1/2

- Start from cycles (easy to extract)

- Combine them:



Algorithm 2/2

We propose an algorithm with three stages:

Extracting cycles: extract cycles for each event in turn, using a dynamic programming routine and a heuristic extracting triples and chaining them

Building tree patterns from cycles: perform combination rounds to generate increasingly complex patterns

Selecting the final pattern collection: solve weighted set cover problem with greedy algorithm

Some qualitative results

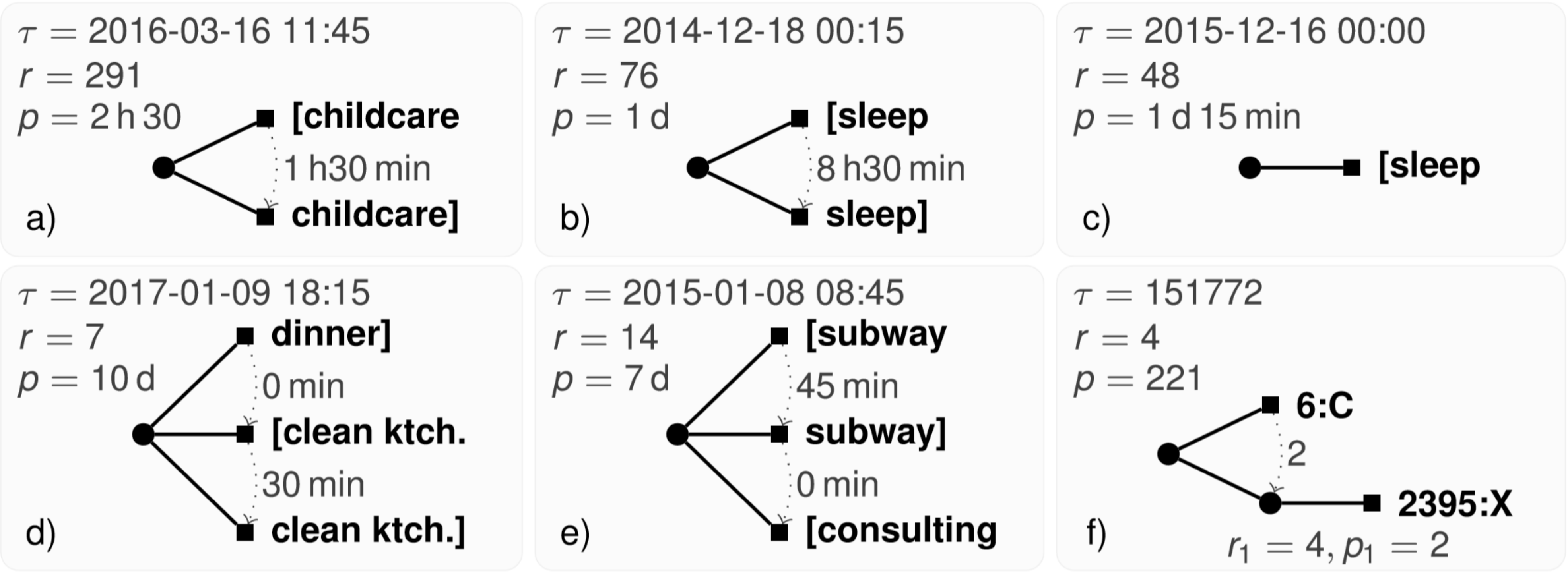


Figure: Example patterns from sacha (a–e) and 3zap (f).

https://arxiv.org/abs/1807.01706: long version of paper -> detailed theory and 30+pages of experiments

How periodic is your shopping?
Analyzing your market basket tickets, v2.0

Clément Gautrais, René Quiniou, Peggy Cellier, Thomas Guyet, Alexandre Termier:
Purchase Signatures of Retail Customers. PAKDD (1) 2017: 110-121

Clément Gautrais, Peggy Cellier, René Quiniou, Alexandre Termier:
Topic Signatures in Political Campaign Speeches. EMNLP 2017: 2342-2347

Signatures: detecting and characterizing recurrent behavior in sequential data, Clément Gautrais PhD, 2018

Slides adapted from Clément Gautrais

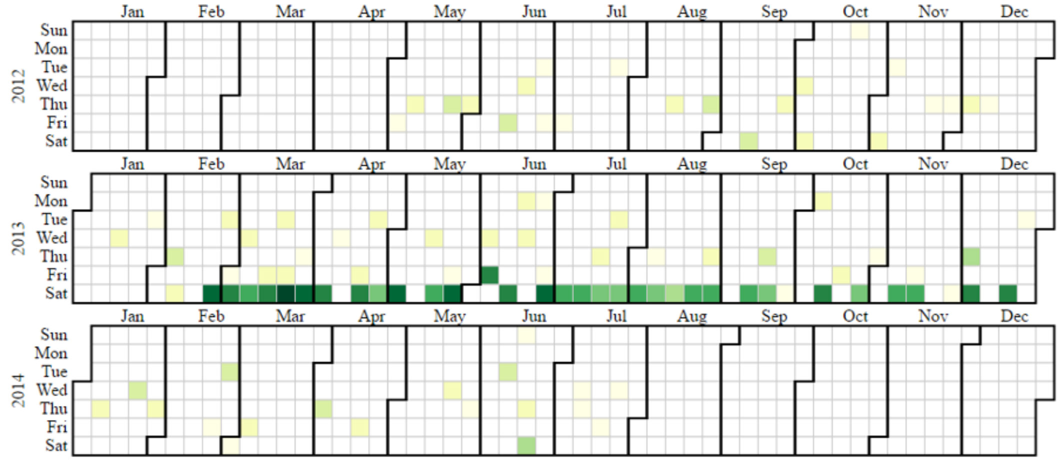
Motivations

- Detection customer habits in market basket data
- => what are the favorite products of customers?
- => how often do they replenish these products?

- Challenges
  - Few results (ideally, ONE pattern with the set of products)
  - Robustness to noise

Example from real data

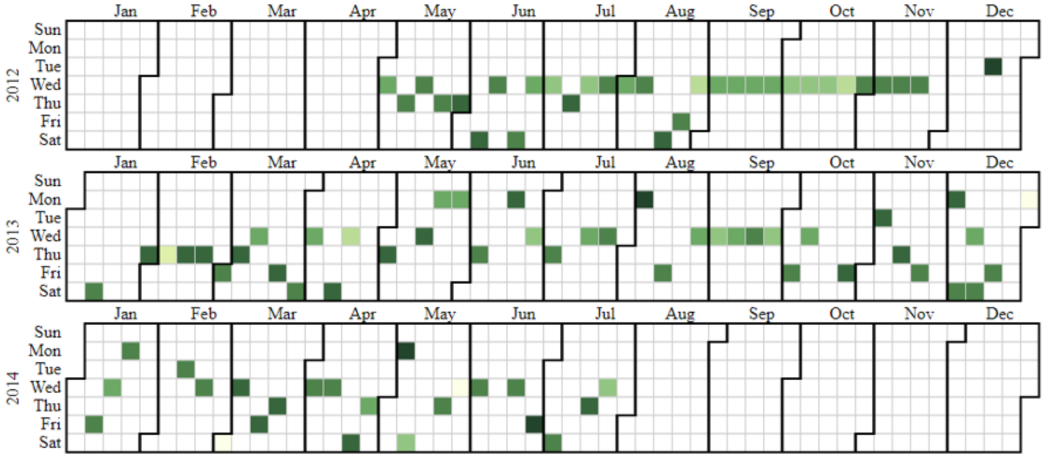
- 2 real customers



Ideal rhythm (replenishment period)

Rare profiles

Might have non ideal purchases



Some regularities

Most profiles

Challenging!

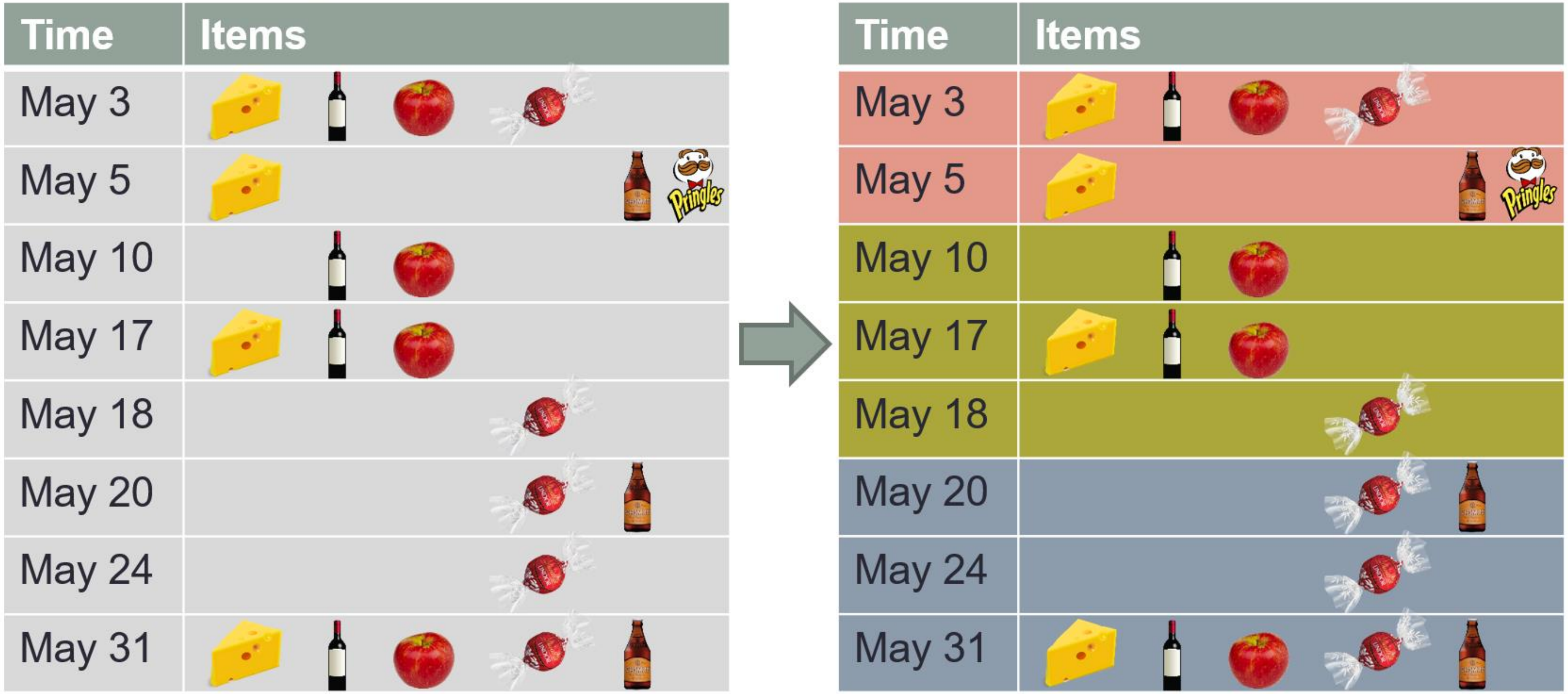
Signature model intuition

- Find favourite products of a customer
  - Bought several times with some regularity
    - Not necessarily in the same transaction

- Find recurrent symbols and their occurrences in a symbolic sequence, with no predefined period
  - A set of products and its occurrences as results
  - Period adapts to the sequence rhythm

Sequence segmentation

- k-segmentation [TT06]: split a sequence of n transactions into k segments

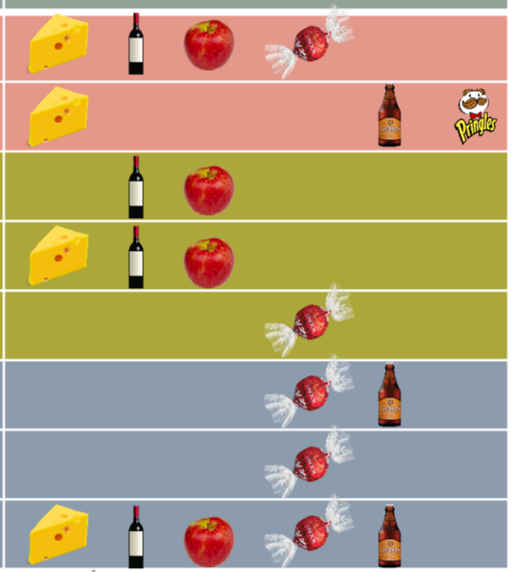


A 3-segmentation of a customer purchase sequence

Segment representative

- Segment representative: $\mu(S_i) = \bigcup_{t \in S_i} t$

	Time	Items
S1	May 3	
	May 5	
S2	May 10	
	May 17	
	May 18	
S3	May 20	
	May 24	
	May 31	



Segment index	Segment representatives $\mu(S_i)$
1	
2	
3	

Adequation

- Adequation: $A(\alpha, S) = \left|\cap_{S_i \in S} \mu(S_i)\right|$

Segment index	Segment representatives $\mu(S_i)$
1	
2	
3	

- $A(\alpha, S) = \left|\cap_{S_i \in S} \mu(S_i)\right| =$ 

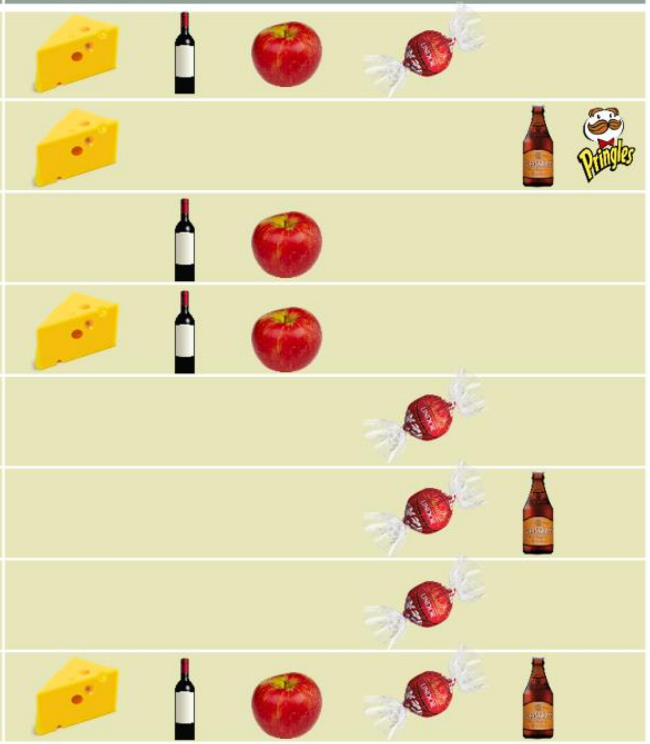
 $= 4$

Segment index	Segment representatives $\mu(S_i)$
1	
2	
3	

Signature problem statement

- $S_{opt}(\alpha, k) = \underset{S \in S_{n,k}}{\operatorname{argmax}} A(\alpha, S)$

Time	Items
May 3	
May 5	
May 10	
May 17	
May 18	
May 20	
May 24	
May 31	



\+ $k = 3$

- Solve $S_{opt}(\alpha, k)$

Example

	Time	Items
S1	May 3	
	May 5	
S2	May 10	
	May 17	
	May 18	
S3	May 20	
	May 24	
	May 31	

Segment index	Segment representatives $\mu(S_i)$
1	
2	
3	

- $A(\alpha, S) = 4$

Example

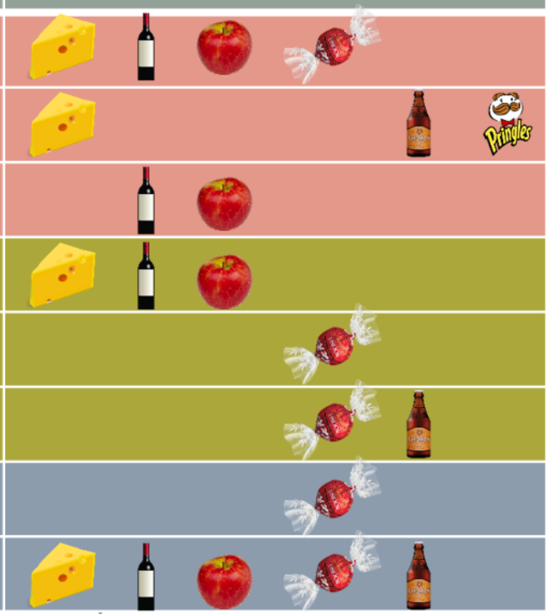
	Time	Items
S1	May 3	
	May 5	
	May 10	
	May 17	
S2	May 18	
	May 20	
S3	May 24	
	May 31	

Segment index	Segment representatives $\mu(S_i)$
1	
2	
3	

- $A(\alpha, S) = 2$

Example

Segment	Time	Items
S1	May 3	
	May 5	
	May 10	
S2	May 17	
	May 18	
	May 20	
S3	May 24	
	May 31	



Segment index	Segment representatives $\mu(S_i)$
1	
2	
3	

- $A(\alpha, S) = 5 = \underset{S \in \mathcal{S}_{8,3}}{\operatorname{argmax}} A(\alpha, S)$

Signature mining

- Mining algorithms: exact approaches
  - Dynamic programming $O(n^2k)$
  - Pattern growth $O(2^{|I|})$

- Mining algorithms: other approaches
  - Greedy algorithms $O(nk)$

Dynamic programming

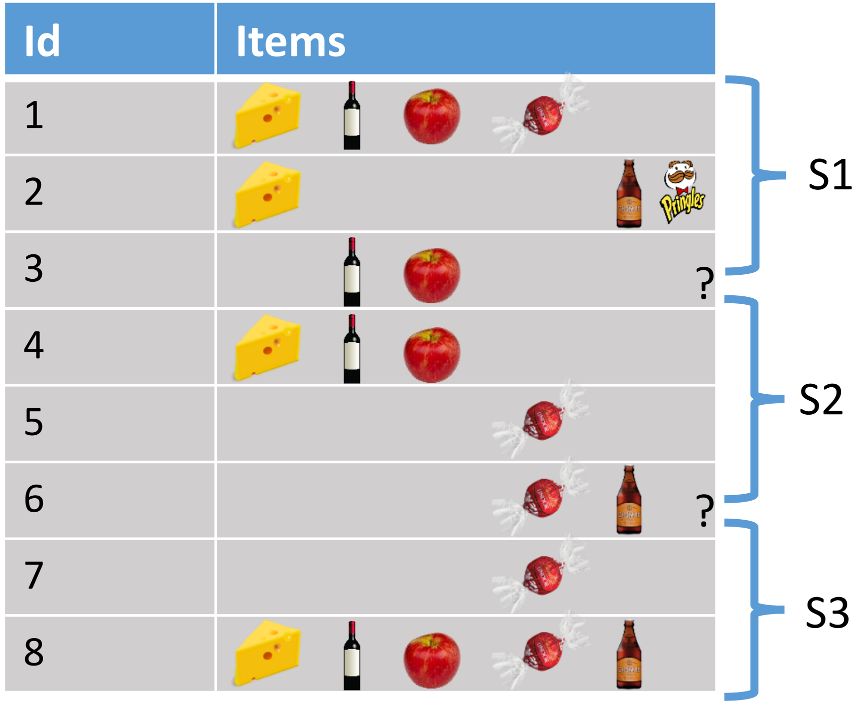
- Dynamic programming [Bel13]
  - Optimization method based on sub problem decompositions

- Find $\underset{S \in \mathcal{S}_{n,k}}{\operatorname{argmax}} A(\alpha, S)$
  - First solve $\underset{S \in \mathcal{S}_{n_1,k-1}}{\operatorname{argmax}} A(\alpha, S) \quad \forall n_1 < n$

Number of segments (k)

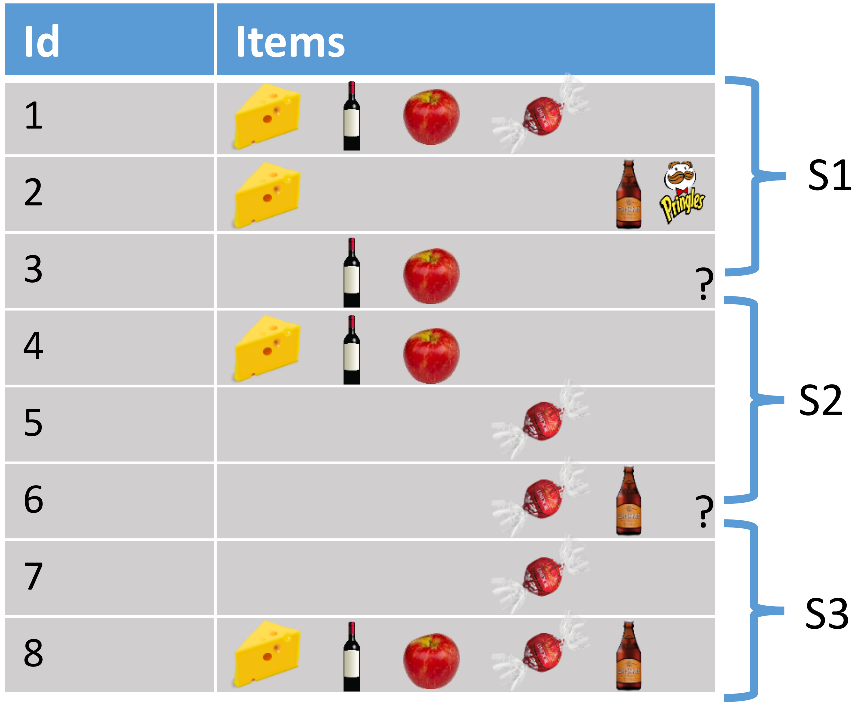
		$\mathcal{S}_{3,3}$	$\mathcal{S}_{4,3}$	$\mathcal{S}_{5,3}$	$\mathcal{S}_{6,3}$	$\mathcal{S}_{7,3}$	$\boldsymbol{\mathcal{S}_{8,3}}$
	$\mathcal{S}_{2,2}$	$\mathcal{S}_{3,2}$	$\mathcal{S}_{4,2}$	$\mathcal{S}_{5,2}$	$\mathcal{S}_{6,2}$	$\mathcal{S}_{7,2}$	
$\mathcal{S}_{1,1}$	$\mathcal{S}_{2,1}$	$\mathcal{S}_{3,1}$	$\mathcal{S}_{4,1}$	$\mathcal{S}_{5,1}$	$\mathcal{S}_{6,1}$		

Sequence length



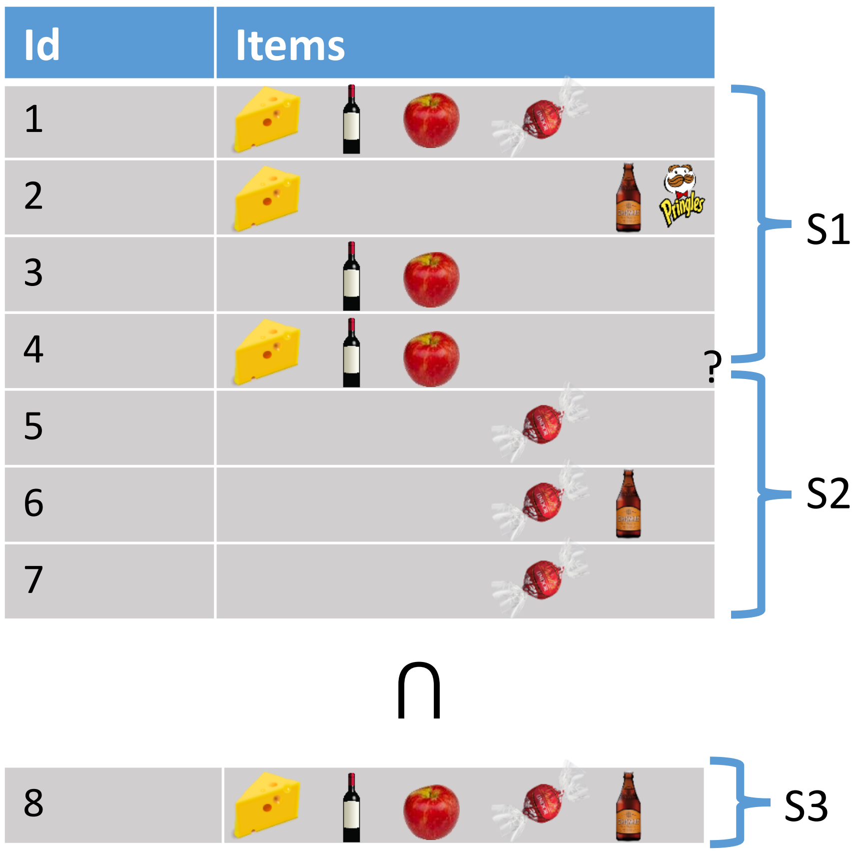
Dynamic programming (level 1.1)

$$\underset{S \in \mathcal{S}_{8,3}}{\operatorname{argmax}} A(\alpha, S)$$



$\Rightarrow$

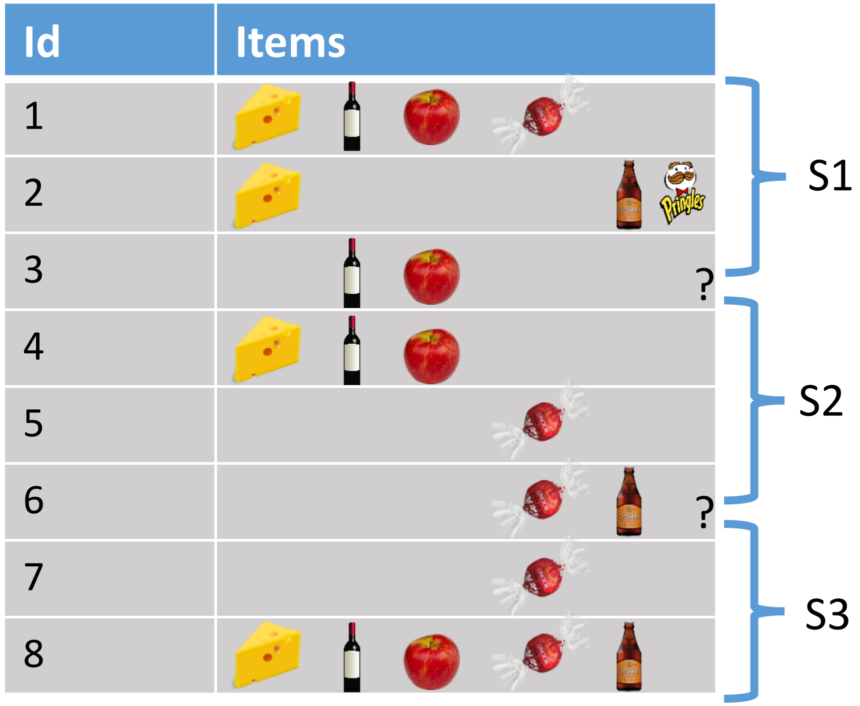
$$\underset{S \in \mathcal{S}_{7,2}}{\operatorname{argmax}} A(\alpha, S)$$



		$\mathcal{S}_{3,3}$	$\mathcal{S}_{4,3}$	$\mathcal{S}_{5,3}$	$\mathcal{S}_{6,3}$	$\mathcal{S}_{7,3}$	$\boldsymbol{\mathcal{S}_{8,3}}$
	$\mathcal{S}_{2,2}$	$\mathcal{S}_{3,2}$	$\mathcal{S}_{4,2}$	$\mathcal{S}_{5,2}$	$\mathcal{S}_{6,2}$	$\mathcal{S}_{7,2}$	
$\mathcal{S}_{1,1}$	$\mathcal{S}_{2,1}$	$\mathcal{S}_{3,1}$	$\mathcal{S}_{4,1}$	$\mathcal{S}_{5,1}$	$\mathcal{S}_{6,1}$		

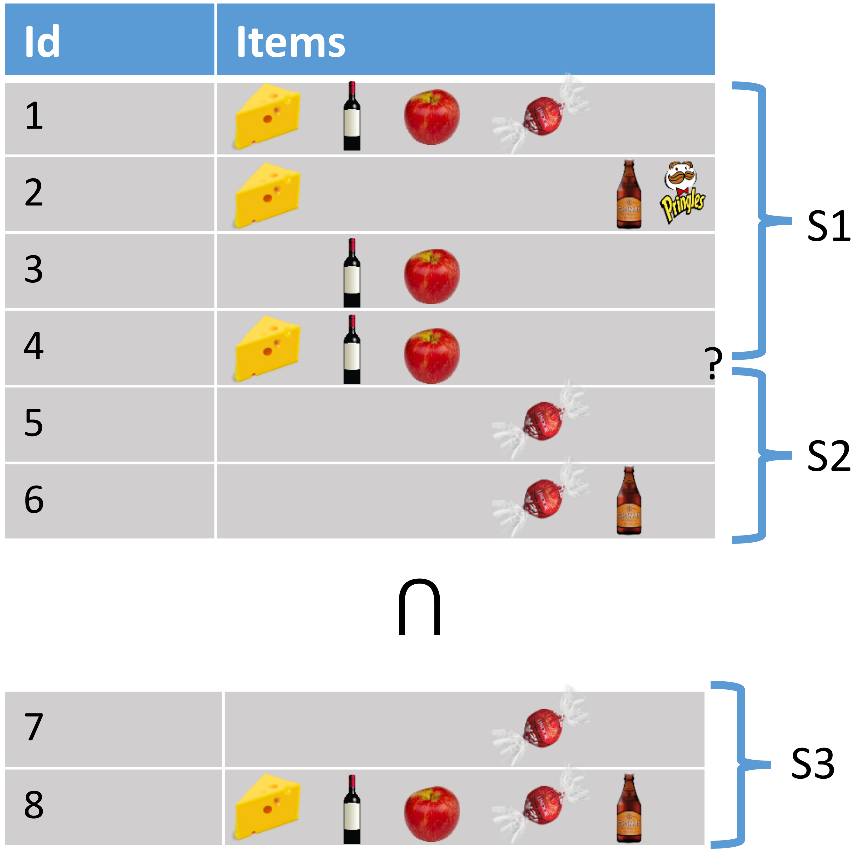
Dynamic programming (level 1.2)

$$\underset{S \in \mathcal{S}_{8,3}}{\operatorname{argmax}} A(\alpha, S)$$



$\Rightarrow$

$$\underset{S \in \mathcal{S}_{6,2}}{\operatorname{argmax}} A(\alpha, S)$$



		$\mathcal{S}_{3,3}$	$\mathcal{S}_{4,3}$	$\mathcal{S}_{5,3}$	$\mathcal{S}_{6,3}$	$\mathcal{S}_{7,3}$	$\boldsymbol{\mathcal{S}_{8,3}}$
	$\mathcal{S}_{2,2}$	$\mathcal{S}_{3,2}$	$\mathcal{S}_{4,2}$	$\mathcal{S}_{5,2}$	$\mathcal{S}_{6,2}$	$\mathcal{S}_{7,2}$	
$\mathcal{S}_{1,1}$	$\mathcal{S}_{2,1}$	$\mathcal{S}_{3,1}$	$\mathcal{S}_{4,1}$	$\mathcal{S}_{5,1}$	$\mathcal{S}_{6,1}$		

Dynamic programming (level 1.3)

$$\underset{S \in \mathcal{S}_{8,3}}{\operatorname{argmax}} A(\alpha, S)$$

Id	Items
1	
2	
3	
4	
5	
6	
7	
8	

S1 (rows 1–3) ?
S2 (rows 4–6) ?
S3 (rows 7–8)

		$\mathcal{S}_{3,3}$	$\mathcal{S}_{4,3}$	$\mathcal{S}_{5,3}$	$\mathcal{S}_{6,3}$	$\mathcal{S}_{7,3}$	$\boldsymbol{\mathcal{S}_{8,3}}$
	$\mathcal{S}_{2,2}$	$\mathcal{S}_{3,2}$	$\mathcal{S}_{4,2}$	$\mathcal{S}_{5,2}$	$\mathcal{S}_{6,2}$	$\mathcal{S}_{7,2}$	
$\mathcal{S}_{1,1}$	$\mathcal{S}_{2,1}$	$\mathcal{S}_{3,1}$	$\mathcal{S}_{4,1}$	$\mathcal{S}_{5,1}$	$\mathcal{S}_{6,1}$		

$\Rightarrow$

$$\underset{S \in \mathcal{S}_{5,2}}{\operatorname{argmax}} A(\alpha, S)$$

Id	Items
1	
2	
3	
4	
5	

S1 (rows 1–3) ?
S2 (rows 4–5)

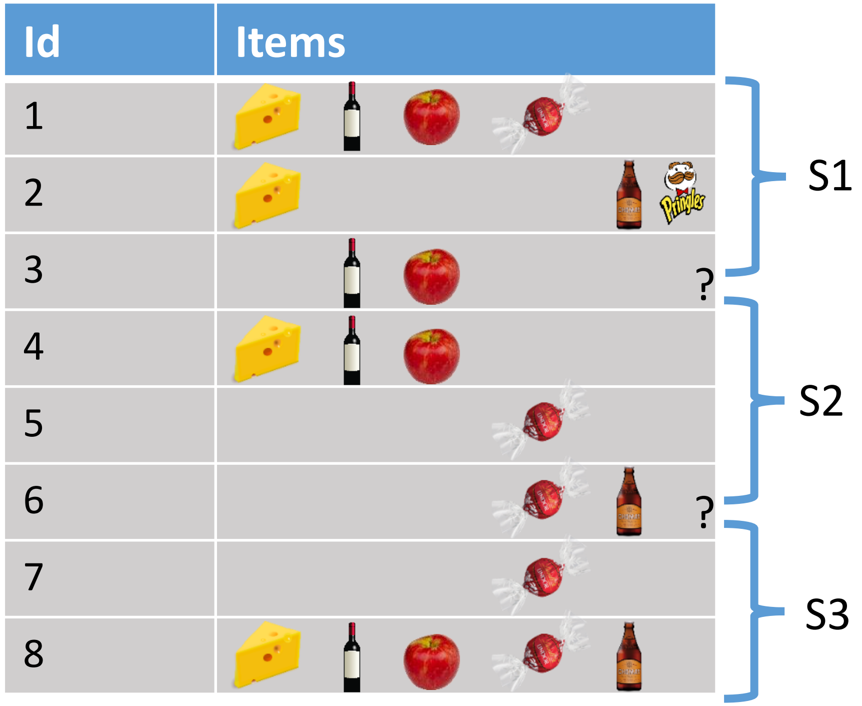
$\cap$

6	
7	
8	

S3

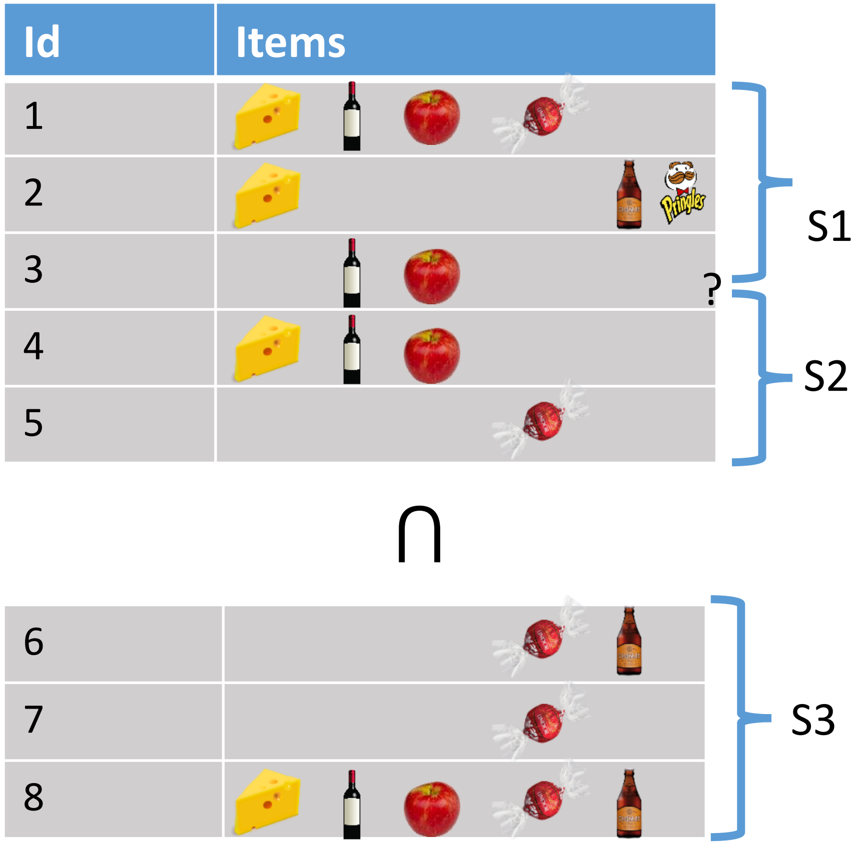
Dynamic programming (level 1.3, in depth)

$$\underset{S \in \mathcal{S}_{8,3}}{\operatorname{argmax}} A(\alpha, S)$$



$\Rightarrow$

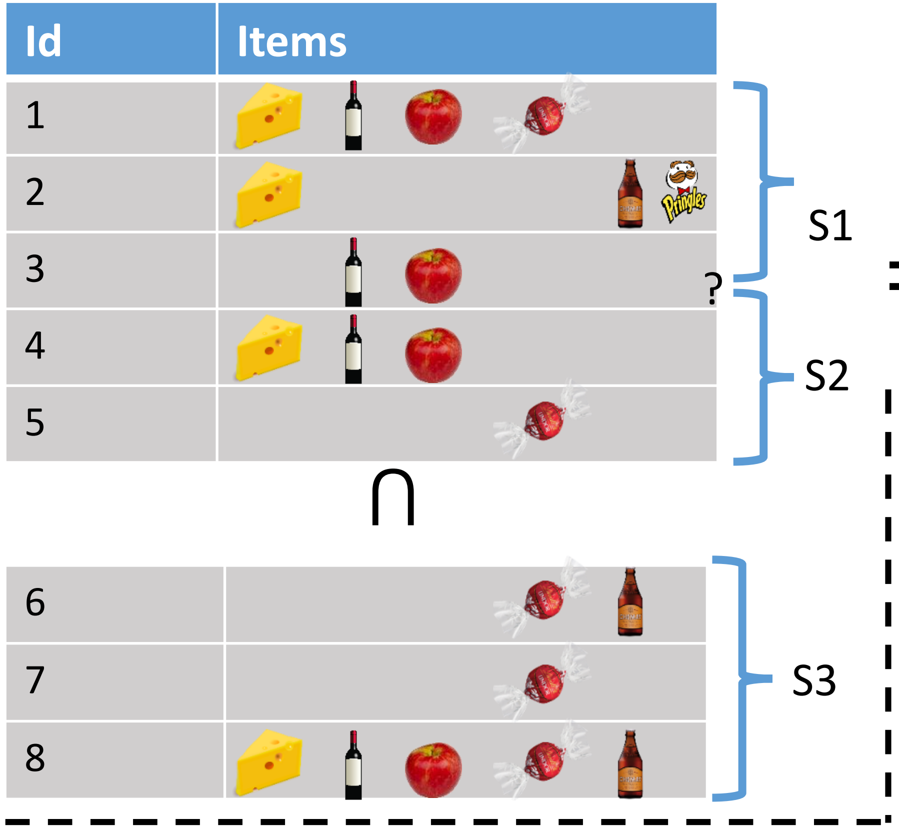
$$\underset{S \in \mathcal{S}_{5,2}}{\operatorname{argmax}} A(\alpha, S)$$



		$\mathcal{S}_{3,3}$	$\mathcal{S}_{4,3}$	$\mathcal{S}_{5,3}$	$\mathcal{S}_{6,3}$	$\mathcal{S}_{7,3}$	$\mathcal{S}_{8,3}$
	$\mathcal{S}_{2,2}$	$\mathcal{S}_{3,2}$	$\mathcal{S}_{4,2}$	$\boldsymbol{\mathcal{S}_{5,2}}$	$\mathcal{S}_{6,2}$	$\mathcal{S}_{7,2}$	
$\mathcal{S}_{1,1}$	$\mathcal{S}_{2,1}$	$\mathcal{S}_{3,1}$	$\mathcal{S}_{4,1}$	$\mathcal{S}_{5,1}$	$\mathcal{S}_{6,1}$		

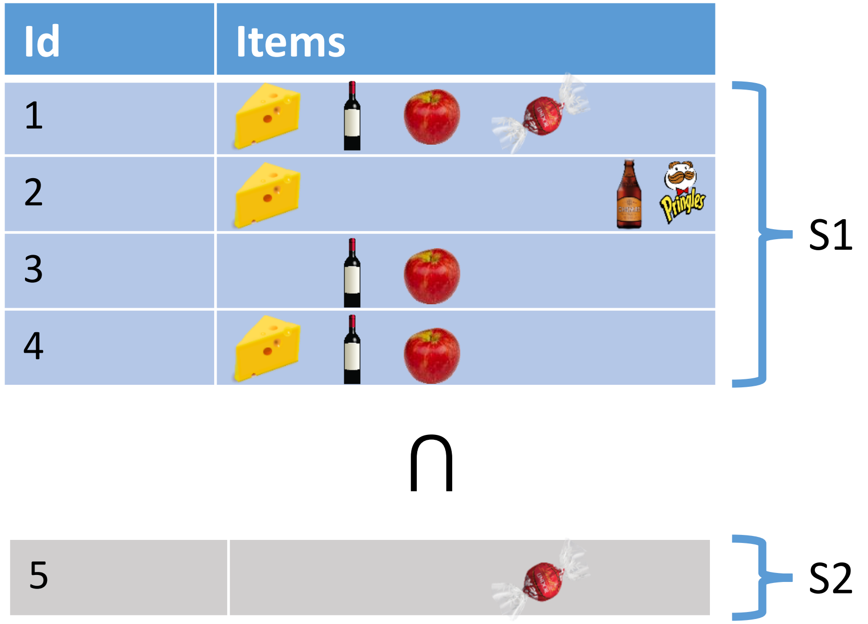
Dynamic programming (level 2.1)

$$\underset{S \in \mathcal{S}_{5,2}}{\operatorname{argmax}} A(\alpha, S)$$



$\Rightarrow$

$$\underset{S \in \mathcal{S}_{4,1}}{\operatorname{argmax}} A(\alpha, S)$$

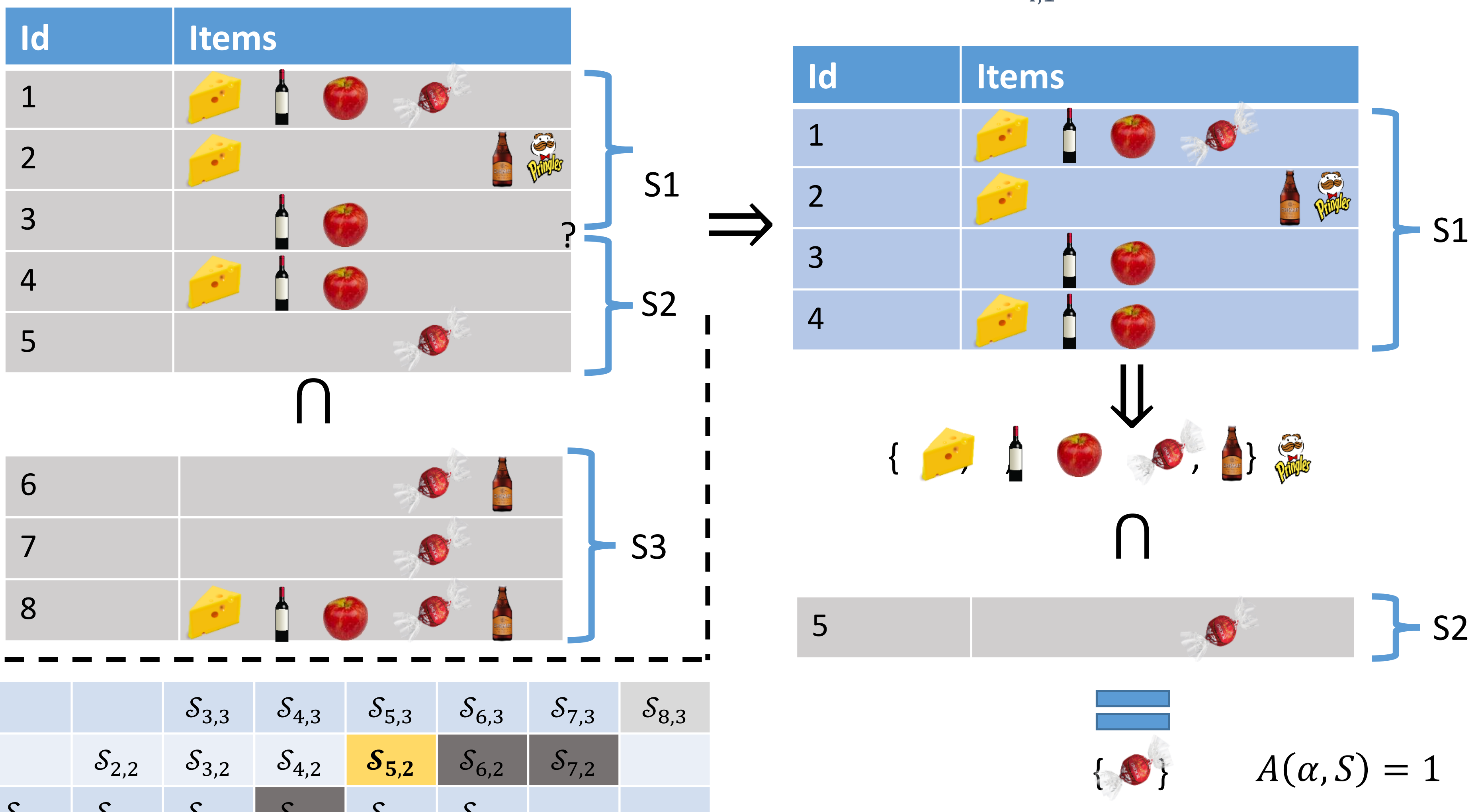


		$\mathcal{S}_{3,3}$	$\mathcal{S}_{4,3}$	$\mathcal{S}_{5,3}$	$\mathcal{S}_{6,3}$	$\mathcal{S}_{7,3}$	$\mathcal{S}_{8,3}$
	$\mathcal{S}_{2,2}$	$\mathcal{S}_{3,2}$	$\mathcal{S}_{4,2}$	$\mathcal{S}_{5,2}$	$\mathcal{S}_{6,2}$	$\mathcal{S}_{7,2}$	
$\mathcal{S}_{1,1}$	$\mathcal{S}_{2,1}$	$\mathcal{S}_{3,1}$	$\mathcal{S}_{4,1}$	$\mathcal{S}_{5,1}$	$\mathcal{S}_{6,1}$		

Dynamic programming (level 2.1)

$\underset{S \in \mathcal{S}_{5,2}}{\operatorname{argmax}} A(\alpha, S)$

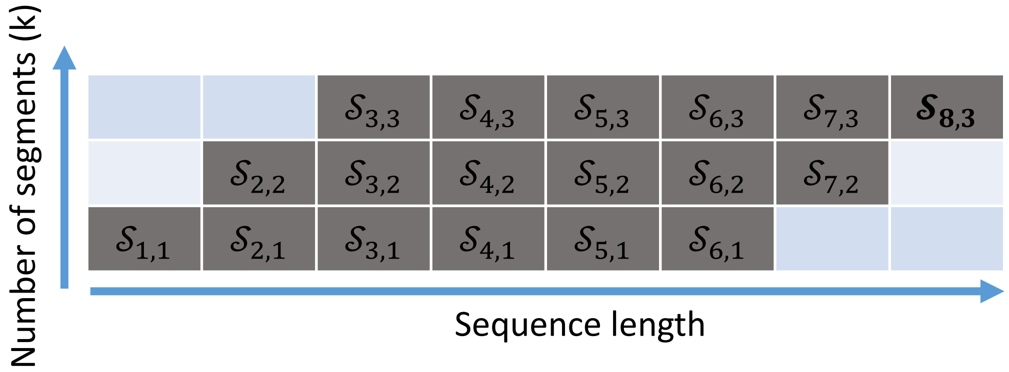
$\underset{S \in \mathcal{S}_{4,1}}{\operatorname{argmax}} A(\alpha, S) = 6$

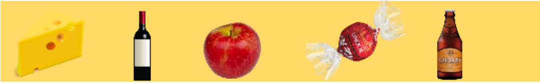


		$\mathcal{S}_{3,3}$	$\mathcal{S}_{4,3}$	$\mathcal{S}_{5,3}$	$\mathcal{S}_{6,3}$	$\mathcal{S}_{7,3}$	$\mathcal{S}_{8,3}$
	$\mathcal{S}_{2,2}$	$\mathcal{S}_{3,2}$	$\mathcal{S}_{4,2}$	$\boldsymbol{\mathcal{S}_{5,2}}$	$\mathcal{S}_{6,2}$	$\mathcal{S}_{7,2}$	
$\mathcal{S}_{1,1}$	$\mathcal{S}_{2,1}$	$\mathcal{S}_{3,1}$	$\mathcal{S}_{4,1}$	$\mathcal{S}_{5,1}$	$\mathcal{S}_{6,1}$		

Dynamic programming (solution)

- In practice
  - We build the matrix row by row(increasing k)
  - The signature is in cell $[n, k]$



Time	Items
May 3	
May 5	
May 10	
May 17	
May 18	
May 20	
May 24	
May 31	

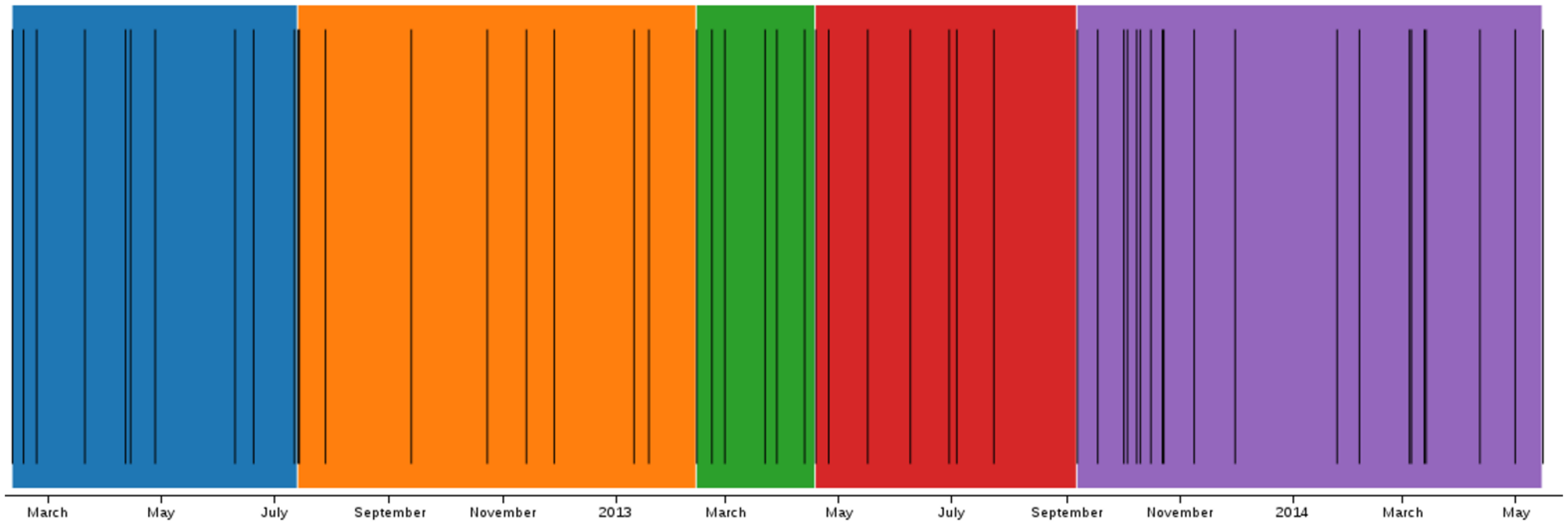
Extensions

- Sky-signatures: based on pareto dominance / skypatterns
  - See EMNLP’18 paper
  - With an interesting analysis of Trump/Clinton campaign speeches!

- MDL signatures: find THE best signature
  - Joint work with Matthijs van Leeuwen
  - Paper should be accepted at some point…
  - *For the impatient: see PhD of Clément Gautrais*

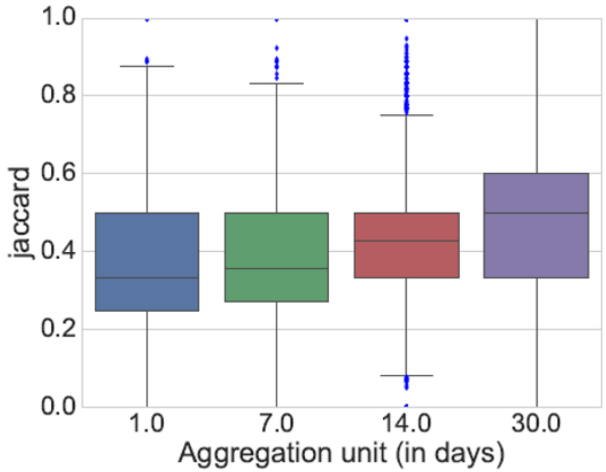
Example signature (real customer data)

- JOKER MULTIFRUIT BRK OVALINE1L
- SIROP SPORT CITROR BTL 1L
- BRETS CHIPS POULET BRAISE 6X25
- RANOU ROTI PORC 6TR 240G
- MINI BABYBEL X12 264G
- IDS CREME CASSIS 20D 70CL
- MT BLANC VANILLE MINI 6X125G
- J.ROZE S.HACHE LETENDR X10 1K
- 1ER PRIX BEURRE 1/2S PQ 500G
- ECR/AD COLOSSE CHOC.BLC4X120
- RANOU ROTI DE PORC 4TR 160G
- PASQUIER BISCOTTE MINC.36T 300
- RANOU JBON MON PARIS DD6T270G
- KINDER PINGUI CHOCOLAT 8X30G
- PASQUIER 12 CROISSANTS 480G

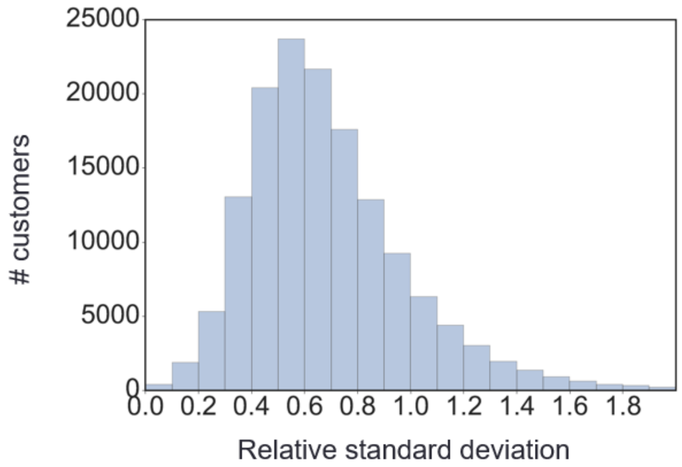


Signature VS periodic patterns

- Periodic pattern
  - Should the repetition constraint be more constrained?



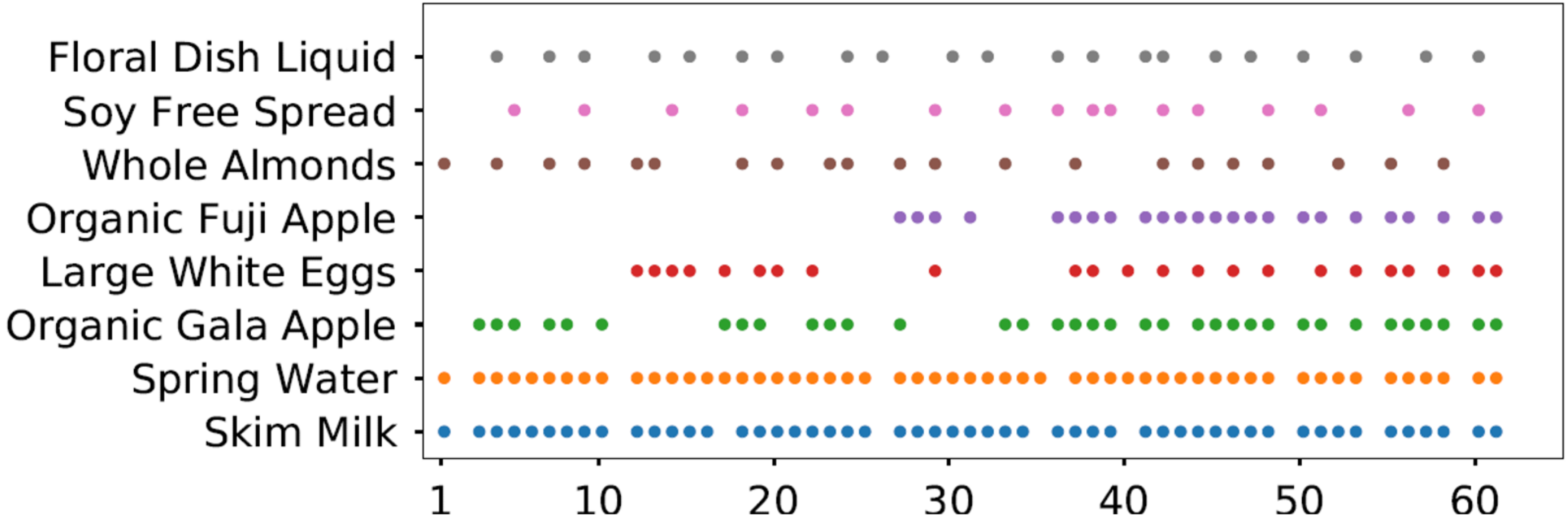
Jaccard similarity between the signature and the longest periodic pattern.



Relative standard deviation of the segment length

  - 50% of the signature is composed of products from the longest periodic pattern
  - The signature detects periodic products, along with non periodic regular products
  - The signature produces stable segments

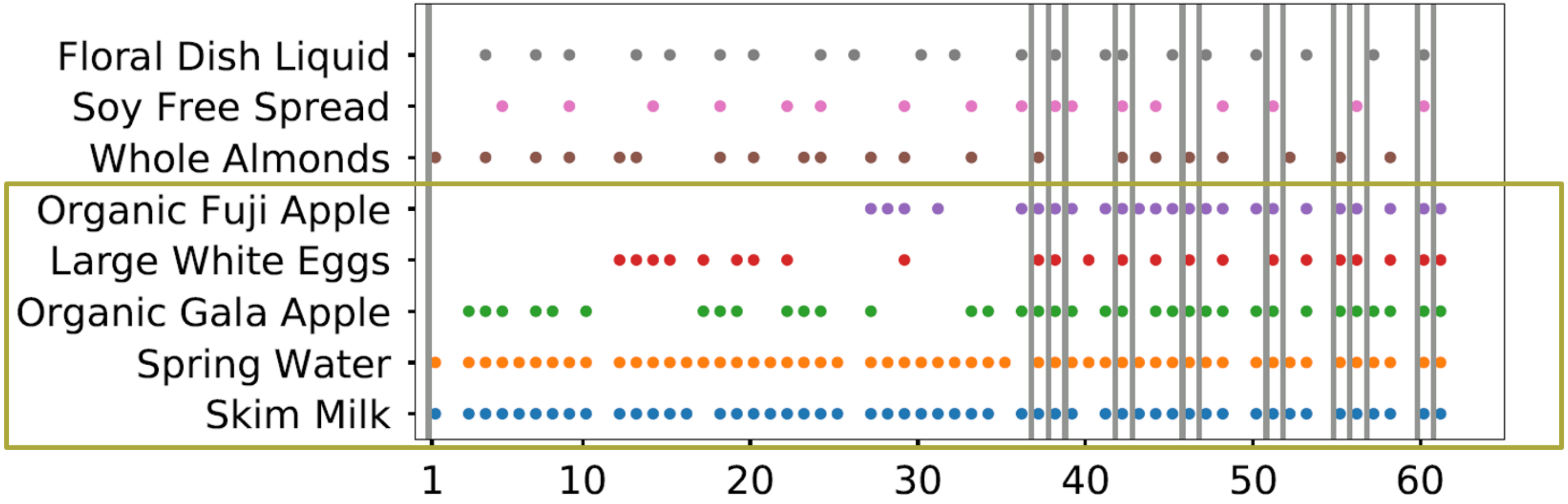
Real use case – Instacart data (Kaggle)



Real use case – Instacart data (Kaggle)

- Best signature found
  - Lowest encoded length
  - Fast and recent purchase habits

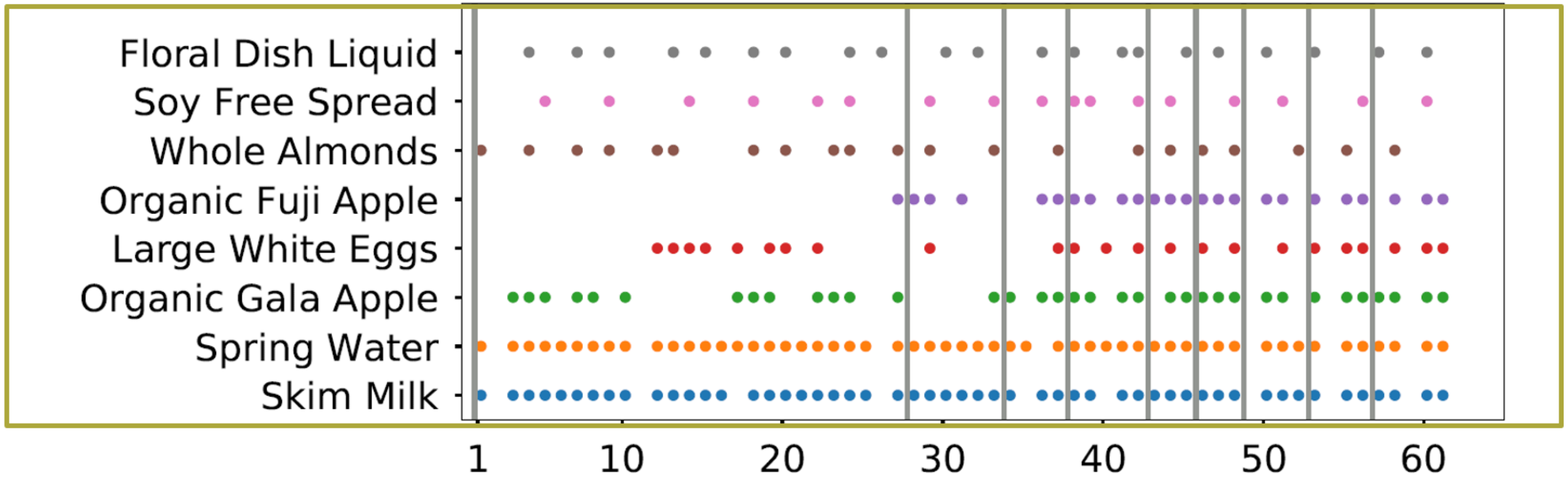
Length	Signature products	Segmentation
1924.72	Skim Milk, Spring Water, Organic Gala Apple, Large White Eggs, Organic Fuji Apple	1,36‖37, 37‖38, 38‖39,41‖42, 42‖43, 45‖46,46‖47, 50‖51, 51‖52,54‖55, 55‖56, 56‖57,59‖60, 60‖61, 61



Real use case – Instacart data (Kaggle)

- Second best signature found
  - Slower purchase rhythm

Length	Signature products	Segmentation
1924.72	Skim Milk, Spring Water, Organic Gala Apple, Large White Eggs, Organic Fuji Apple	1,36‖37, 37‖38, 38‖ 39,41‖42, 42‖43, 45‖ 46,46‖47, 50‖51, 51‖ 52,54‖55, 55‖56, 56‖ 57,59‖60, 60‖61, 61
1983.30	First signature + Whole Almonds +Soy Free Spread +Floral Dish Liquid	1,27‖28, 33‖34, 37‖ 38,42‖43, 45‖46, 48‖ 49,52‖53, 56‖57, 61‖



Conclusion

- Three approaches for mining temporal regularities presented
  - Quite strict cycles, gaps allowed between cycles, transaction data, condensed representation
  - Tolerant + nested cycles, sequence data, MDL
  - Segmentation, transaction data, optimisation/Pareto/MDL

- Many other interesting problems await

- Surprisingly few people in that research area (since 1999)

Perspectives

- Robustness, robustness, robustness
  - Most periodic pattern definitions break to easily
  - -> prevent the discovery of more general/covering patterns
- Take into account domain knowledge
- Provide easy to use implementations
  - Introducing the Scikit-Mine project

Thank you for your attention!