
Kuliah 12

Sequential Pattern Mining

Sumber:

[http://www.is.informatik.uni-
duisburg.de/courses/im_ss09/fohlen/MiningSequentialPatterns.ppt](http://www.is.informatik.uni-
duisburg.de/courses/im_ss09/fohlen/MiningSequentialPatterns.ppt)

Outline

- Apa itu basis data sekuens dan sequential pattern mining
- Metode-metode untuk sequential pattern mining
- Constraint-based sequential pattern mining
- Periodicity analysis untuk data sekuens

Basis data Sekuens

- Sebuah basis data sekuen terdiri dari elemen-elemen atau kejadian-kejadian terurut
- Basis data transaksi vs basis data sekuens

Basis data transaksi

TID	itemsets
10	a, b, d
20	a, c, d
30	a, d, e
40	b, e, f

Basis data sekuens

SID	sequences
10	<a(<u>ab</u> c)(a <u>c</u>)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(<u>ab</u>)(df) <u>c</u> b>
40	<eg(af)cbc>

Aplikasi

- Aplikasi dari sequential pattern mining
 - Urutan pembelajaran dari konsumen:
 - Pertama memberi komputer, kemudian CD-ROM, dan kemudian digital camera, dalam 3 bulan.
 - Perawatan medis, bencana alam (contoh gempa bumi), proses sains dan rekayasa, stock dan pemasaran, dll
 - Pola panggilan telepon, Weblog click streams
 - Sekuens DNA dan struktur gen

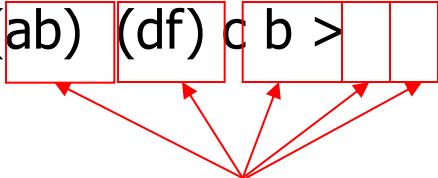
Sub sekuens vs. super sekuens

- Sebuah sekuens adalah daftar terurut dari kejadian dinotasikan $\langle e_1 e_2 \dots e_l \rangle$
- Diberikan 2 sekuens $\alpha = \langle a_1 a_2 \dots a_n \rangle$ dan $\beta = \langle b_1 b_2 \dots b_m \rangle$
- α disebut sub sekuens dari β , dinotasikan sebagai $\alpha \subseteq \beta$, jika terdapat integer $1 \leq j_1 < j_2 < \dots < j_n \leq m$ sedemikian sehingga $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \dots, a_n \subseteq b_{j_n}$
- β adalah super sekuens dari α
 - E.g. $\alpha = \langle (ab), d \rangle$ and $\beta = \langle (abc), (de) \rangle$

Apa itu Sequential Pattern Mining?

- Diberikan himpunan sekuens dan support threshold, temukan himpunan lengkap dari *frequent* subsequences

Sekuens: < (ef) (ab) (df) c b >



Basis data sekuens

SID	sequence
10	<a(<u>ab</u> c)(a <u>c</u>)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(<u>ab</u>)(df) <u>c</u> b>
40	<eg(af)cbc>

Sebuah elemen dapat mengandung sekumpulan A.

Items dalam sebuah elemen adalah tidak terurut dan dituliskan secara alfabetik.

<a(bc)dc> is a *subsequence* of <a(abc)(ac)d(cf)>

support threshold yang diberikan (*min_sup*) = 2,
<(ab)c> adalah *sequential pattern*

Tantangan dalam Sequential Pattern Mining (SPM)

- Jumlah yang besar dari pola sekuens yang mungkin diperoleh, yang tersimpan dalam basis data
- Algoritme SPM harus
 - Dapat menemukan **himpunan lengkap dari pola**, apabila memungkinkan, yang memenuhi minimum support (frekuensi) threshold
 - **efisien, scalable**, melibatkan sedikit scanning basis data
 - Mampu menangani berbagai jenis **user-specific constraints**

Penelitian terkait Sequential Pattern Mining

- Pengenalan konsep dan algoritme Apriori-like
 - Agrawal & Srikant. Mining sequential patterns, [ICDE'95]
- Metode berbasis Apriori: **GSP** (Generalized Sequential Patterns: Srikant & Agrawal [EDBT'96])
- Metode Pattern-growth : FreeSpan & **PrefixSpan** (Han et al. KDD'00; Pei, et al. [ICDE'01])
- Vertical format-based mining: **SPADE** (Zaki [Machine Learning'00])
- Sequential pattern mining berbasis kendala (SPIRIT: Garofalakis, Rastogi, Shim [VLDB'99]; Pei, Han, Wang [CIKM'02])
- Mining closed sequential patterns: **CloSpan** (Yan, Han & Afshar [SDM'03])

Metode-metode untuk sequential pattern mining

- **Pendekatan berbasis Apriori**
 - GSP
 - SPADE
- **Pendekatan berbasis Pattern-Growth**
 - FreeSpan
 - PrefixSpan

Sifat Apriori dari Pola Sequential

- Sifat dasar: Apriori (Agrawal & Sirkant'94)
 - Jika sebuah sekuens S adalah tidak frequent, maka tidak ada dari super-sequences dari S yang frequent
 - E.g, <hb> adalah infrequent begitu juga dengan <hab> dan <(ah)b>

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Diberikan support
threshold $min_sup = 2$

GSP—Generalized Sequential Pattern Mining

- Algoritme GSP (Generalized Sequential Pattern) mining
- Outline metode GSP
 - Langkah awal, setiap item dalam DB adalah kandidat dari sekuen dengan panjang 1
 - Untuk setiap level (i.e., sekuens dengan panjang- k) lakukan
 - scan basis data untuk menentukan support count untuk setiap sekuen kandidat
 - Bangkitkan kandidat sekuen dengan panjang $-(k+1)$ dari sekuen dengan panjang- k yang frequent menggunakan Apriori
 - Ulangi sampai tidak ada sekuen yang frequent atau tidak ada kandidat yang dapat ditemukan
- Kekuaran utama: pemangkasan kandidat oleh prinsip Apriori

Mendapatkan Pola Sequential dengan panjang 1

- Kandidat awal:
 - $\langle a \rangle$, $\langle b \rangle$, $\langle c \rangle$, $\langle d \rangle$, $\langle e \rangle$, $\langle f \rangle$, $\langle g \rangle$, $\langle h \rangle$
- Scan basis data satu kali, hitung support untuk setiap kandidat

$min_sup = 2$

Seq. ID	Sequence
10	$\langle (bd)cb(ac) \rangle$
20	$\langle (bf)(ce)b(fg) \rangle$
30	$\langle (ah)(bf)abf \rangle$
40	$\langle (be)(ce)d \rangle$
50	$\langle a(bd)bcb(ade) \rangle$

Kandidat	Sup
$\langle a \rangle$	3
$\langle b \rangle$	5
$\langle c \rangle$	4
$\langle d \rangle$	3
$\langle e \rangle$	3
$\langle f \rangle$	2
$\langle g \rangle$	1
$\langle h \rangle$	1

Membangkitkan Kandidat dengan panjang 2

51 kandidat
dengan panjang 2

	<a>		<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
	<ba>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<ca>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<da>	<db>	<dc>	<dd>	<de>	<df>
<e>	<ea>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fa>	<fb>	<fc>	<fd>	<fe>	<ff>

	<a>		<c>	<d>	<e>	<f>
<a>		<(ab)>	<(ac)>	<(ad)>	<(ae)>	<(af)>
			<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>				<(cd)>	<(ce)>	<(cf)>
<d>					<(de)>	<(df)>
<e>						<(ef)>
<f>						

Tanpa sifat Apriori,
 $8*8+8*7/2=92$
 kandidat

Apriori memangkas
 44.57% kandidat₁₃

Mendapatkan Pola Sequential dengan panjang 2

- Scan basis data sekali lagi, hitung support count untuk setiap kandidat dengan panjang 2
- Terdapat 19 kandidat dengan panjang 2 yang memenuhi minimum support threshold
 - Sekuens tersebut adalah pola sequential dengan panjang 2

Proses Mining menggunakan GSP

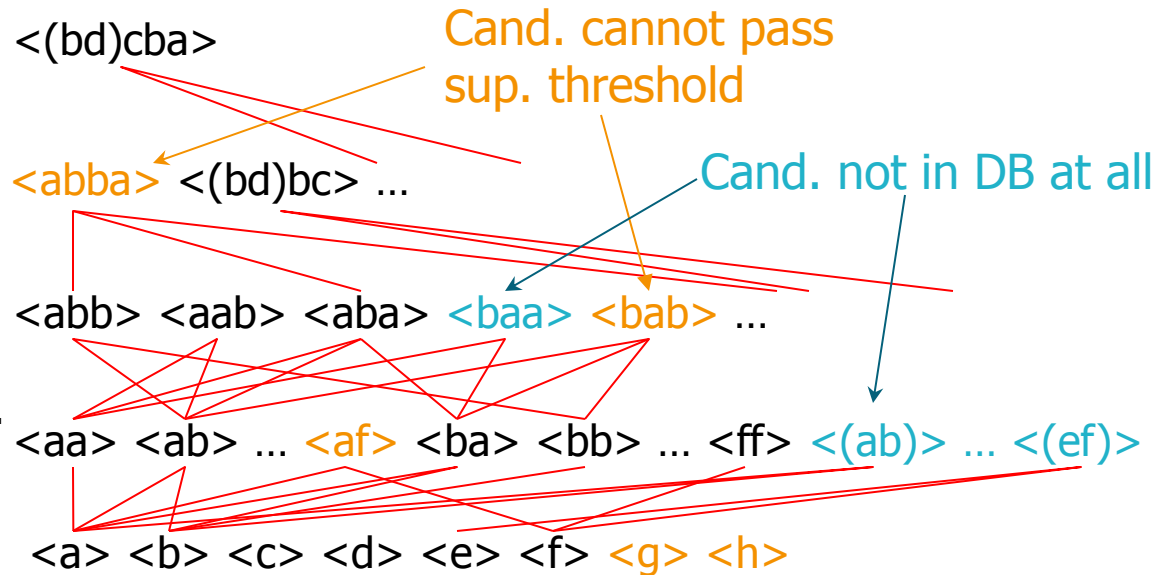
5th scan: 1 cand. 1 length-5 seq.
pat.

4th scan: 8 cand. 6 length-4 seq.
pat.

3rd scan: 46 cand. 19 length-3 seq.
pat. 20 cand. not in DB at all

2nd scan: 51 cand. 19 length-2 seq.
pat. 10 cand. not in DB at all

1st scan: 8 cand. 6 length-1 seq.
pat.



$min_sup = 2$

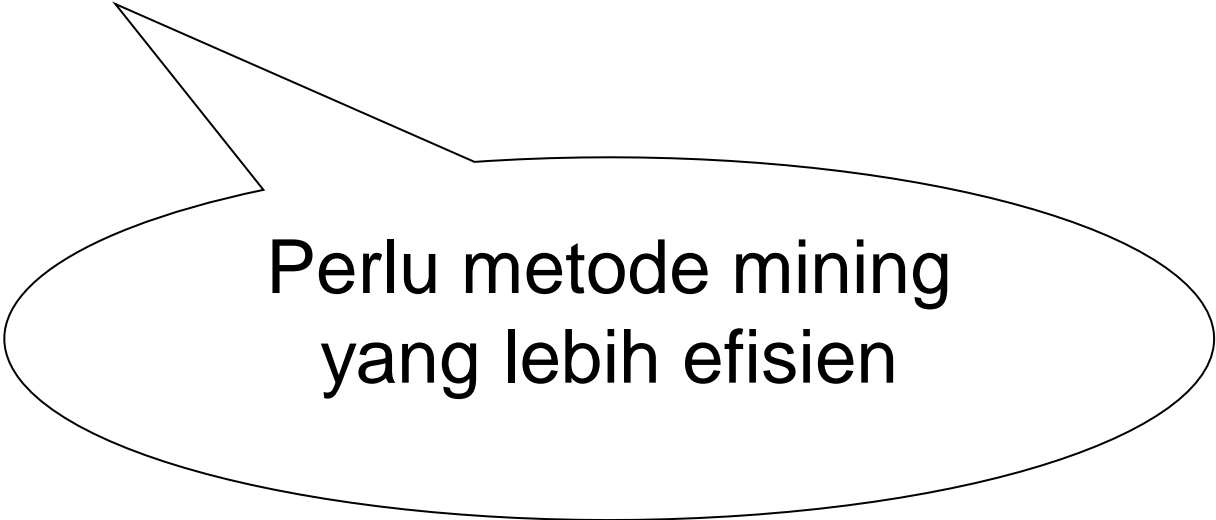
Seq. ID	Sequence
10	$\langle (bd)cb(ac) \rangle$
20	$\langle (bf)(ce)b(fg) \rangle$
30	$\langle (ah)(bf)abf \rangle$
40	$\langle (be)(ce)d \rangle$
50	$\langle a(bd)bcb(ade) \rangle$

Algoritme GSP

- Ambil sekuen-sekuen dalam bentuk $\langle x \rangle$ sebagai kandidat dengan panjang 1
- Scan basis data sekali, temukan F_1 , himpunan pola sekuens dengan panjang 1
- Misalkan $k=1$; selama F_k tidak kosong lakukan
 - Dari C_{k+1} , himpunan kandidat dengan panjang $(k+1)$ dari F_k ;
 - Jika C_{k+1} tidak kosong, scan basis data sekali, temukan F_{k+1} , himpunan pola sekuen dengan panjang $(k+1)$
 - Misalkan $k=k+1$;

Algoritme GSP

- Keuntungan dari Apriori pruning
 - Mengurangi ruang pencarian
- Bottlenecks
 - Scan basis data berkali-kali
 - Membangkitkan himpunan besar dari sekuens kandidat



Perlu metode mining
yang lebih efisien

Algoritme SPADE

- SPADE (Sequential Pattern Discovery using Equivalent Class) dibangun oleh Zaki 2001
- Merupakan metode sequential pattern mining dengan format vertikal
- Basis data sekuens dipetakan ke himpunan berukuran besar dari item: <SID, EID>
- Sequential pattern mining dilakukan dengan
 - Membangkitkan (pola) subsekuens 1 item setiap saat dengan pembangkitan kandidat berdasarkan prinsip Apriori

Algorithme SPADE

SID	EID	Items
1	1	a
1	2	abc
1	3	ac
1	4	d
1	5	cf
2	1	ad
2	2	c
2	3	bc
2	4	ae
3	1	ef
3	2	ab
3	3	df
3	4	c
3	5	b
4	1	e
4	2	g
4	3	af
4	4	c
4	5	b
4	6	c

a		b		...
SID	EID	SID	EID	...
1	1	1	2	
1	2	2	3	
1	3	3	2	
2	1	3	5	
2	4	4	5	
3	2			
4	3			

ab			ba			...
SID	EID (a)	EID(b)	SID	EID (b)	EID(a)	...
1	1	2	1	2	3	
2	1	3	2	3	4	
3	2	5				
4	3	5				

aba				...
SID	EID (a)	EID(b)	EID(a)	...
1	1	2	3	
2	1	3	4	

Bottleneck dalam pembangkitan dan pengujian kandidat

- Himpunan yang besar dari kandidat dibangkitkan.
 - Khususnya sekuen kandidat dengan 2-item.
- Banyak proses scanning dari basis data dalam mining.
 - Panjang setiap kandidat meningkat 1 pada setiap proses scan basis data.
- Tidak efisien untuk mining pola sekuens yang panjang.
 - Pola yang panjang dibuat dari pola yang pendek
 - Kandidat yang pendek jumlahnya eksponensial

PrefixSpan (Prefix-Projected Sequential Pattern Growth)

- PrefixSpan
 - Berbasis proyeksi
 - Tetapi hanya proyeksi berbasis prefiks: proyeksi yang sedikit dan jumlah sekuens menyusut dengan cepat
- J.Pei, J.Han,... PrefixSpan : Mining sequential patterns efficiently by prefix-projected pattern growth. ICDE'01.

Prefix dan Suffix (Proyeksi)

- ▶ $\langle a \rangle$, $\langle aa \rangle$, $\langle a(ab) \rangle$ and $\langle a(abc) \rangle$ adalah prefixes dari sekuens $\langle a(abc)(ac)d(cf) \rangle$
- ▶ Diberikan sekuens $\langle a(abc)(ac)d(cf) \rangle$

Prefix	<u>Suffix</u> (Proyeksi berbasis Prefix)
$\langle a \rangle$	$\langle (abc)(ac)d(cf) \rangle$
$\langle aa \rangle$	$\langle (_bc)(ac)d(cf) \rangle$
$\langle ab \rangle$	$\langle (_c)(ac)d(cf) \rangle$

Mining Sequential Pattern dengan Proyeksi Prefix

- Langkah 1: Tentukan semua pola sequential dengan panjang 1
 - $\langle a \rangle$, $\langle b \rangle$, $\langle c \rangle$, $\langle d \rangle$, $\langle e \rangle$, $\langle f \rangle$
- Langkah 2: bagi ruang pencarian. Himpunan lengkap pola kandidat pola sekuens dapat dipartisi ke dalam 6 subset:
 - Yang memiliki prefix $\langle a \rangle$;
 - Yang memiliki prefix $\langle b \rangle$;
 - ...
 - Yang memiliki prefix $\langle f \rangle$

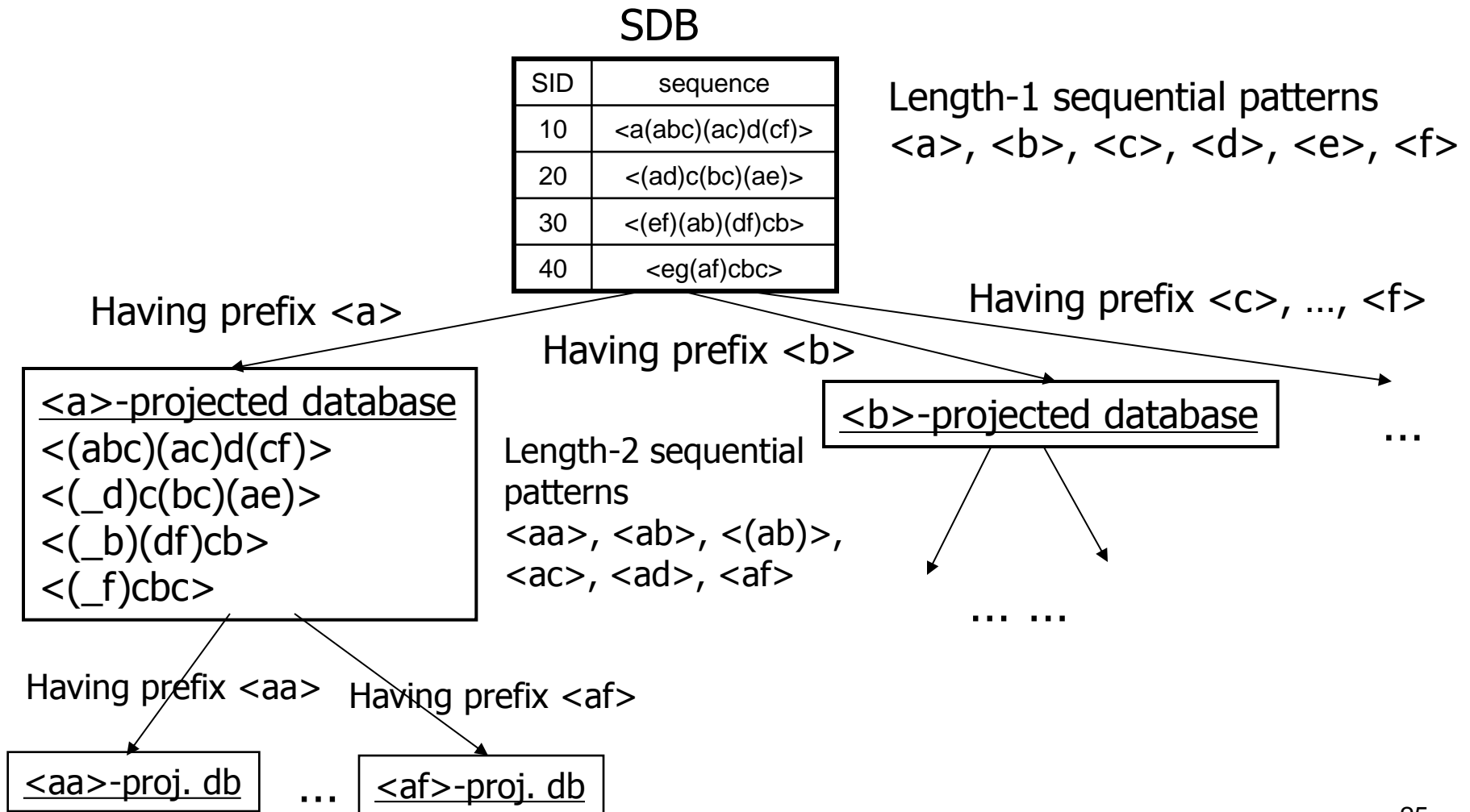
SID	sekuens
10	$\langle a(abc)(ac)d(cf) \rangle$
20	$\langle (ad)c(bc)(ae) \rangle$
30	$\langle (ef)(ab)(df)cb \rangle$
40	$\langle eg(af)cbc \rangle$

Tentukan pola sekuens dengan Prefix <a>

- Hanya perlu memperhatikan proyeksi terhadap <a>
 - <a>-projected database: <(abc)(ac)d(cf)>, <(_d)c(bc)(ae)>, <(_b)(df)cb>, <(_f)cbc>
- Tentukan semua pola sekuens dengan panjang 2 yang memiliki prefix <a>: <aa>, <ab>, <(ab)>, <ac>, <ad>, <af>
 - Selanjutnya dibagi ke dalam 6 subset
 - Yang memiliki prefix <aa>;
 - ...
 - Yang memiliki prefix <af>

SID	sekuens
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

Cara Kerja PrefixSpan



Algoritme PrefixSpan

- **Input:** Basis data sekuens S , minimum support threshold min_sup
- **Output:** himpunan lengkap dari pola sekuens
- **Metode:** Panggil fungsi $\text{PrefixSpan}(\langle \rangle, 0, S)$
- **Subrutin:** $\text{PrefixSpan}(\alpha, l, S|\alpha)$
- **Parameter:**
 - α : pola sekuens,
 - l : panjang dari α ;
 - $S|\alpha$: α -projected database, jika $\alpha \neq \langle \rangle$; selainnya; basis data sekuens S

Algoritme PrefixSpan (2)

Metode

1. Scan $S|\alpha$ sekali, temukan himpunan frequent item b sedemikian sehingga:
 - a) b dapat dirangkai ke elemen terakhir dari α untuk membentuk pola sekuens; atau
 - b) $\langle b \rangle$ dapat ditambahkan ke α untuk membentuk pola sekuens.
2. Untuk setiap frequent item b , tambahkan b ke α untuk membentuk pola sequential α' , dan output α' ;
3. Untuk setiap α' , buat α' -projected database $S|\alpha'$, dan panggil PrefixSpan(α' , $l+1$, $S|\alpha'$).

Efisiensi PrefixSpan

- Tidak ada kandidat sekuen yang perlu dibandingkan
- Projected databases tetapi berukuran kecil
- Biaya dalam PrefixSpan: membuat projected databases
 - Dapat diperbaiki dengan bi-level projection

Optimasi dalam PrefixSpan

- Proyeksi Single level vs. bi-level
 - Proyeksi Bi-level projection dengan 3 cara pemeriksaan dapat mengurangi jumlah dan ukuran projected databases
- Proyeksi fisikal vs. pseudo-projection
 - Pseudo-projection dapat mengurangi kerja dari projection ketika projected database sesuai ukurannya dalam main memory
- Parallel projection vs. partition projection
 - Partition projection dapat menghindari 'blowup' ruang disk

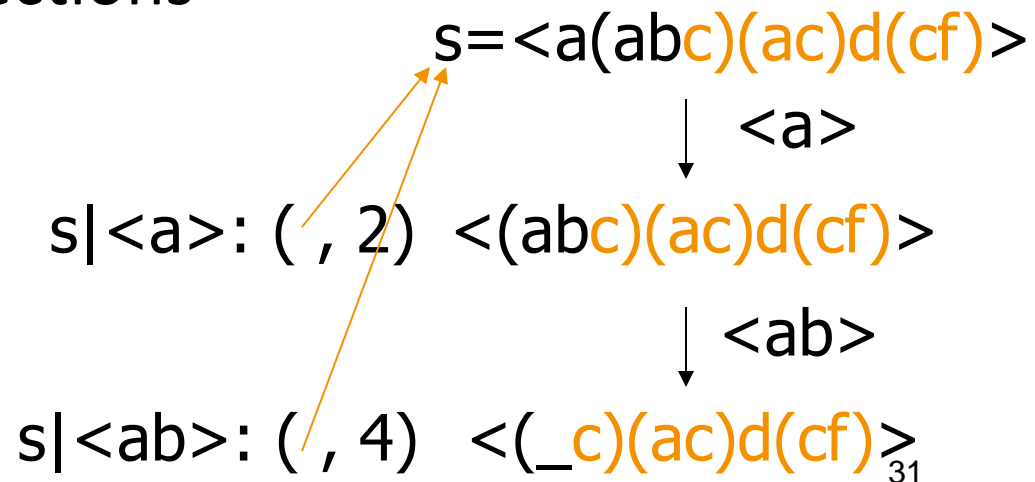
Scaling Up dengan Proyeksi Bi-Level

- Membagi ruang pencarian berdasarkan pola sekuens dengan panjang 2
- Hanya membentuk projected databases dan melakukan recursive mining pada bi-level projected databases

Speed-up dengan Pseudo-projection

- Biaya dalam PrefixSpan: proyeksi
 - Postfix dari sekuens sering muncul secara berulang dalam recursive projected databases
- Ketika (projected) database dapat dilakukan dalam main memory, gunakan pointers untuk membentuk projections

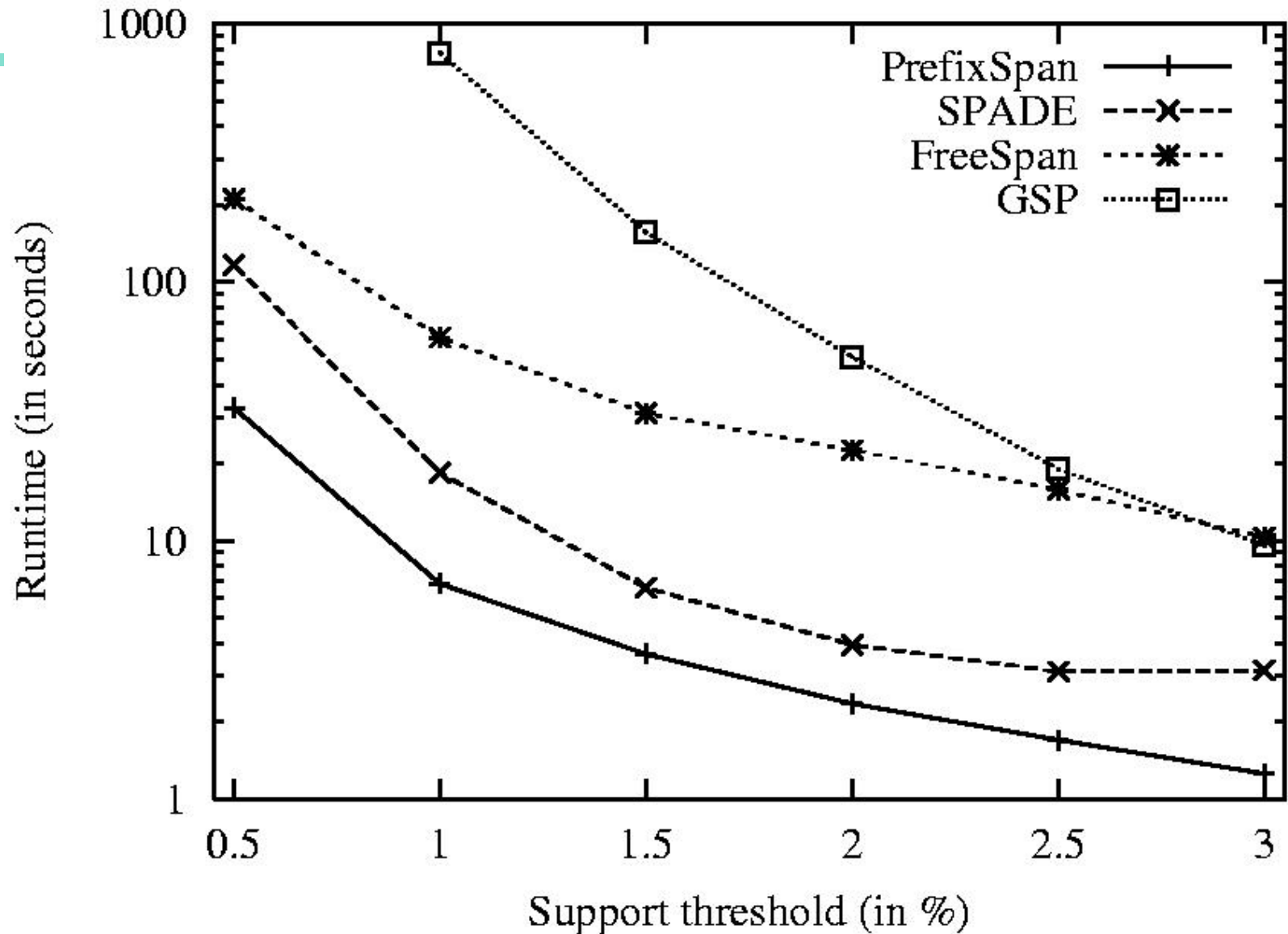
- Pointer ke sekuens
- Mengimbangi postfix



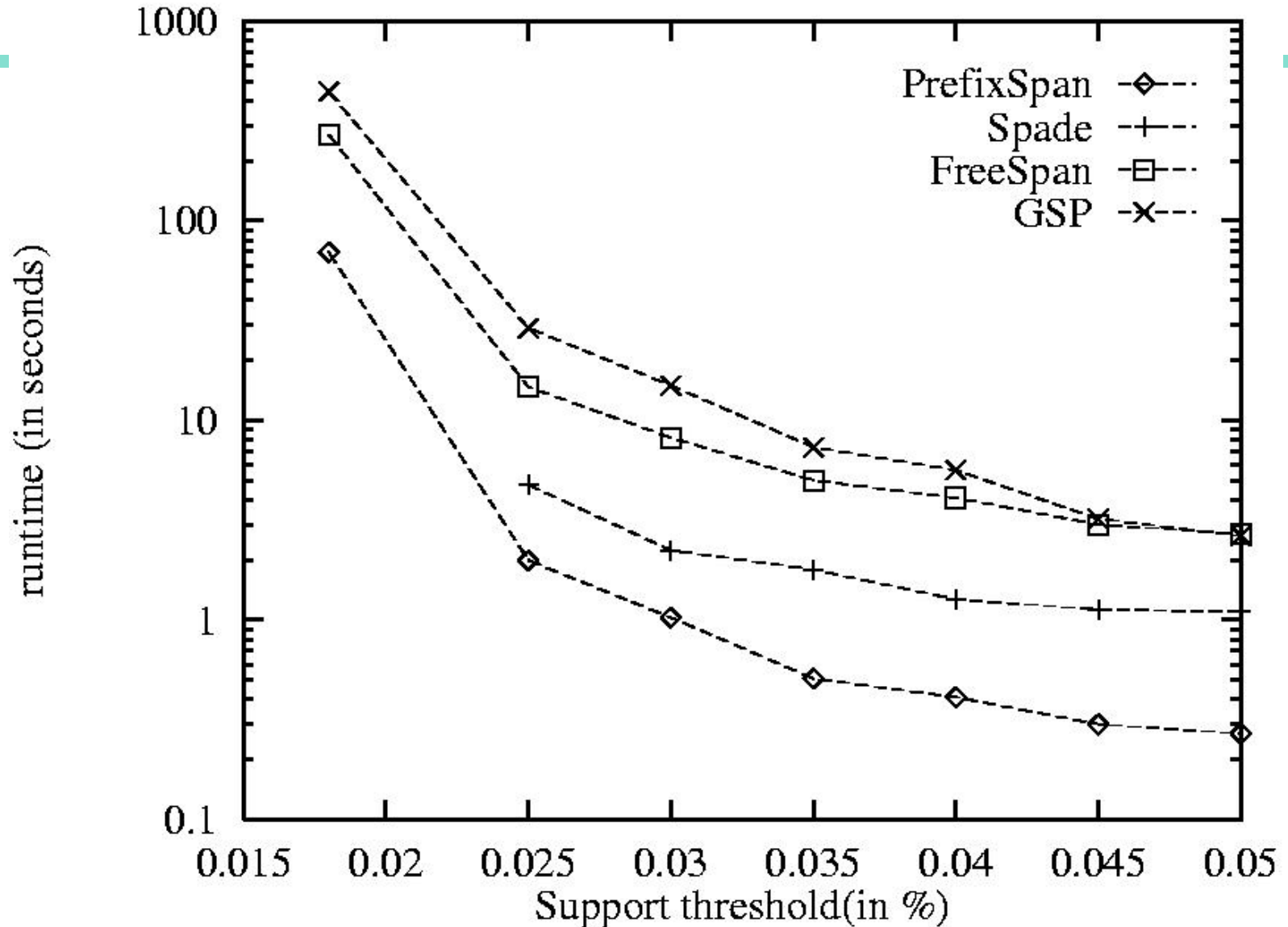
Pseudo-Projection vs. Physical Projection

- ▶ Pseudo-projection menghindari penyalinan postfix secara fisik
 - ▶ Efisien dalam running time dan space ketika basis data dapat disimpan dalam main memory
- ▶ Walaupun demikian, pendekatan ini tidak efisien ketika basis data tidak muat dalam main memory
 - ▶ Pengaksesan secara acak berbasis disk memerlukan biaya yang tinggi
- ▶ Pendekatan yang disarankan:
 - ▶ Integrasi physical dan pseudo-projection
 - ▶ Swapping ke pseudo-projection ketika data set tidak muat dalam memori

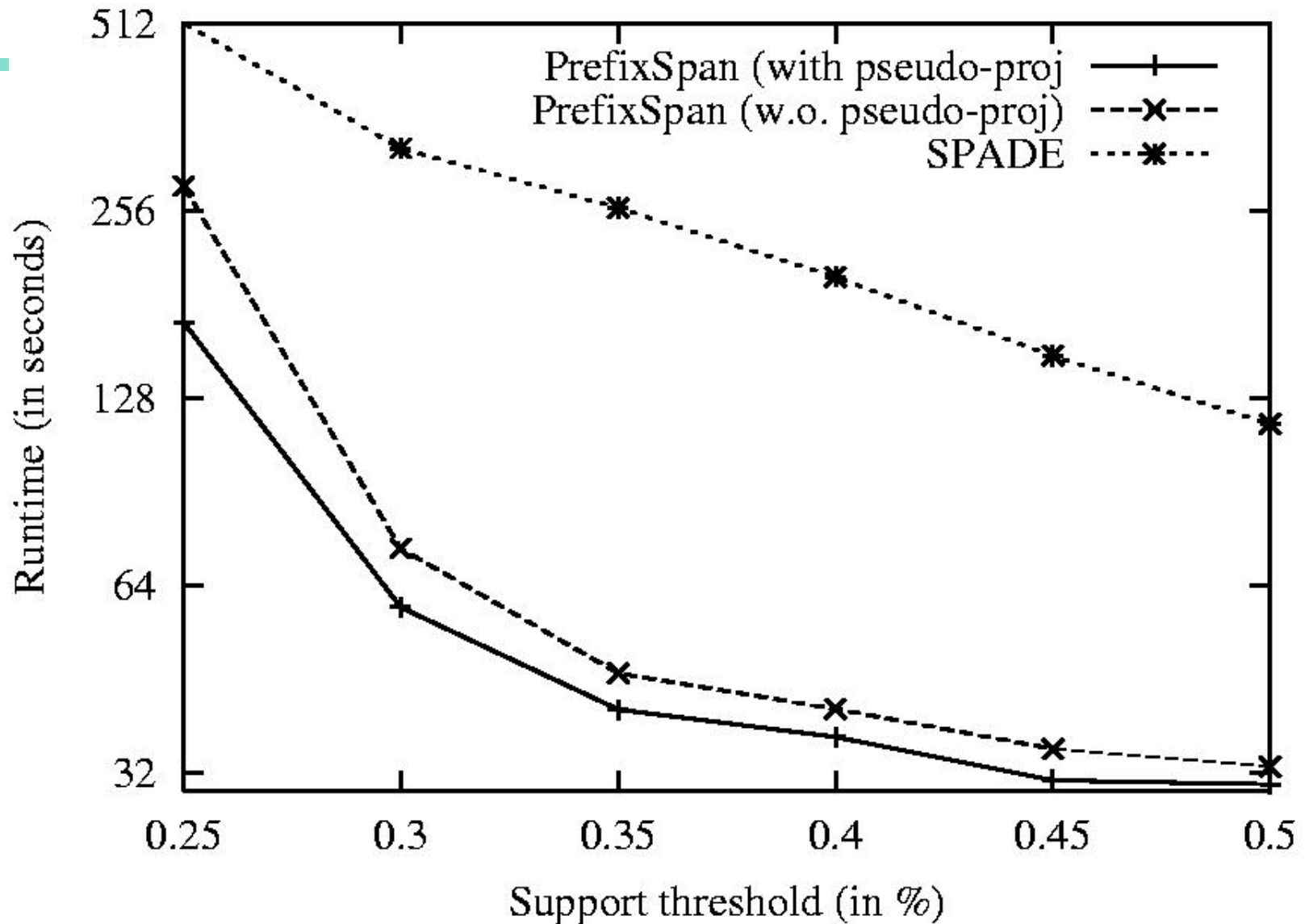
Kinerja algoritme pada Data Set C10T8S8I8



Kinerja algoritme pada Data Set Gazelle

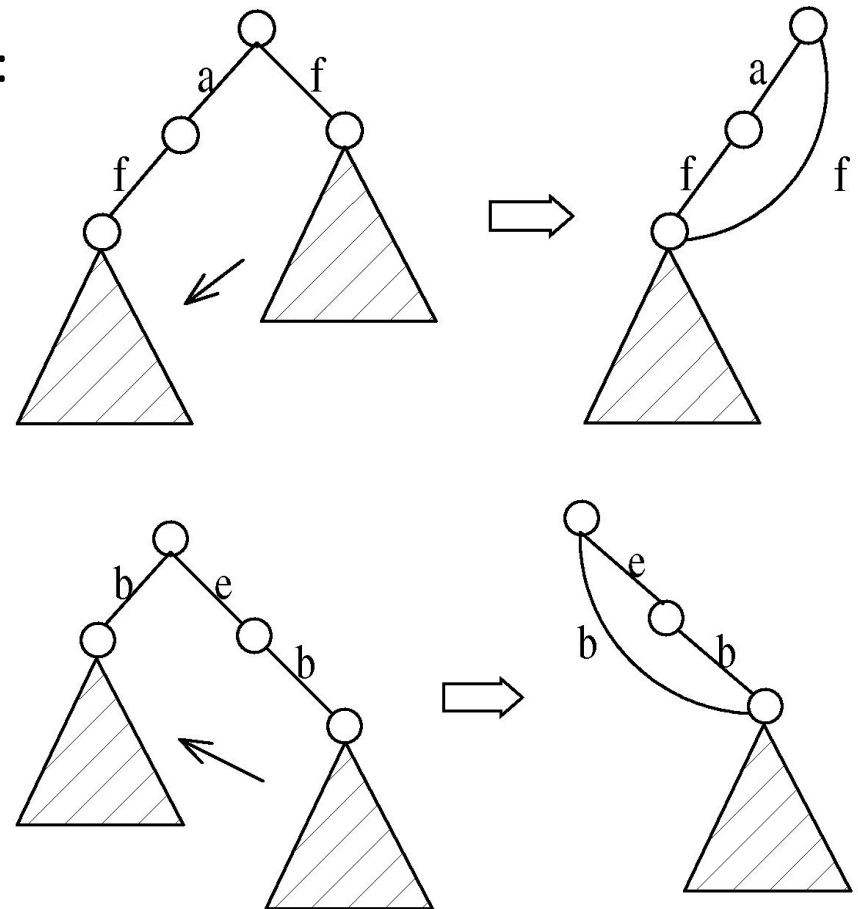


Pengaruh Pseudo-Projection

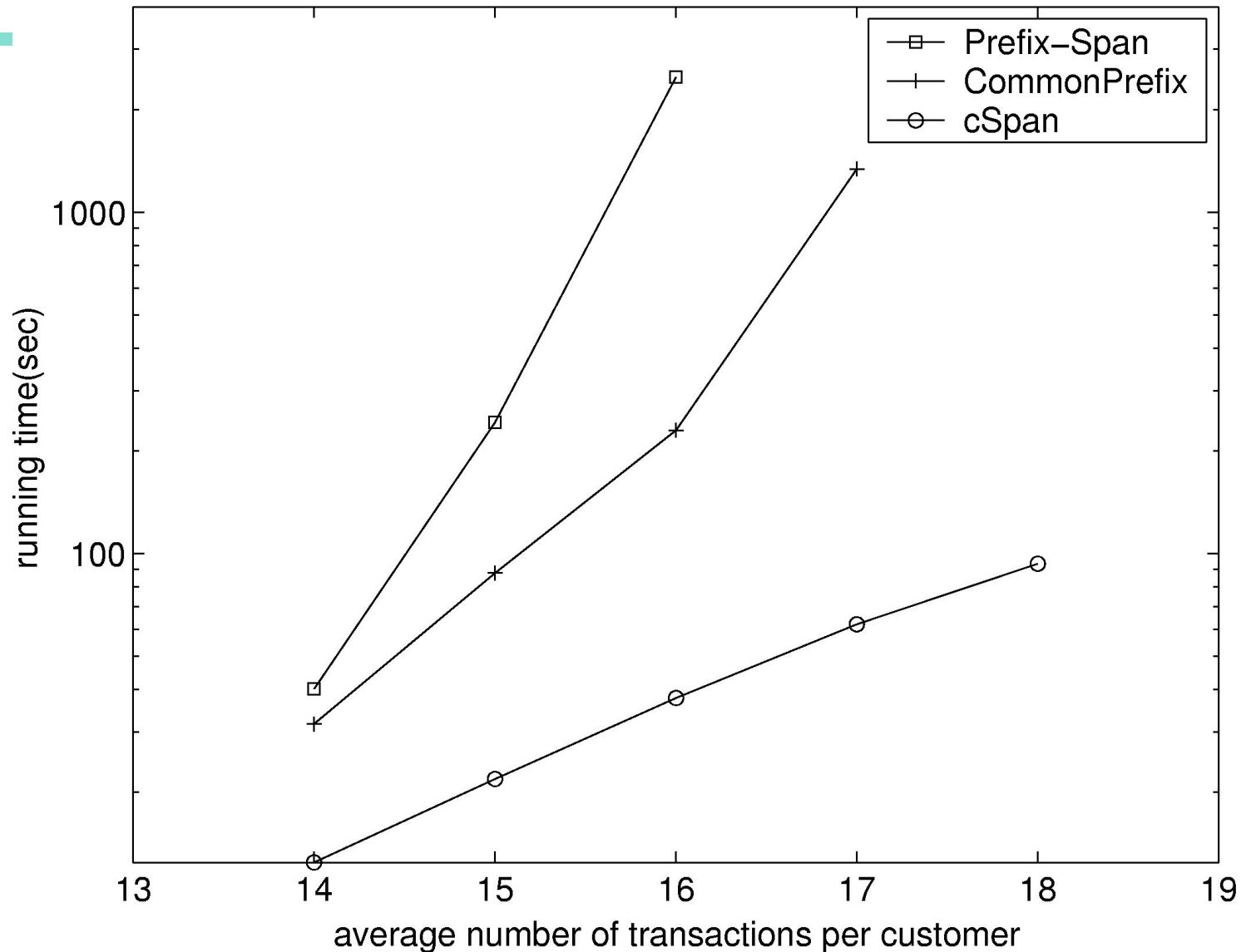


CloSpan: Mining Closed Sequential Patterns

- Sebuah **closed sequential pattern** s : tidak ada superpattern s' sedemikian sehingga $s' \supset s$, dan s' dan s yang memiliki support yang sama
- Motivasi: mengurangi banyaknya pola (yang redundan) tapi memerlukan sumberdaya yang sama
- Menggunakan pemangkasan Backward Subpattern dan Backward Superpattern untuk memangksa ruang pencarian yang redundan



CloSpan: Perbandingan Kinerja dengan PrefixSpan



Kendala pada Sequential Pattern Mining

- Kendala Item
 - Carilah pola web log yang hanya terkait online-bookstores
- Kendala panjang (length)
 - Cari pola yang memiliki sedikitnya 20 barang
- Kendala super pattern
 - Cari super patterns dari “PC → digital camera”
- Kendala agregat
 - Cari pola yang rata-rata harga barangnya melebihi \$100

Kendala pada Sequential Pattern Mining

- Kendala ekspresi regular
 - Carilah pola “starting from Yahoo homepage, search for hotels in Washington DC area”
 - Yahootravel(WashingtonDC|DC)(hotel|motel|lodging)
- Kendala durasi
 - Carilah pola sekitar ± 24 jam dari pengambilan
- Kendala gap
 - Carilah pola pembelian sedemikian sehingga “gap antara setiap pembelian yang berurutan kurang dari 1 bulan”

Dari Sequential Pattern ke Structured Patterns

- Sets, sequences, trees, graphs, dan struktur yang lain
 - Transaksi DB: Sets of items
 - $\{\{i_1, i_2, \dots, i_m\}, \dots\}$
 - Seq. DB: Urutan dari himpunan:
 - $\{\langle\{i_1, i_2\}, \dots, \{i_m, i_n, i_k\}\rangle, \dots\}$
 - Himpunan dari sekuens:
 - $\{\{\langle i_1, i_2 \rangle, \dots, \langle i_m, i_n, i_k \rangle\}, \dots\}$
 - Himpunan dari tree: $\{t_1, t_2, \dots, t_n\}$
 - Himpunan dari graf (mining untuk frequent subgraphs):
 - $\{g_1, g_2, \dots, g_n\}$
- Mining pola terstruktur dalam dokumen XML, struktur bio-chemical, etc.

Episode dan Episode Pattern Mining

- Metode lain untuk menentukan macam-macam pola
 - Serial episodes: $A \rightarrow B$
 - Parallel episodes: $A \& B$
 - Regular expressions: $(A \mid B)C^*(D \rightarrow E)$
- Metode untuk episode pattern mining
 - Variasi dari algoritme Apriori-like contoh, GSP
 - Pembangkitan pola berdasarkan proyeksi basis data
 - Mirip dengan pembangkitan frequent pattern tanpa pembangkitan kandidat

Periodicity Analysis

- Periodicity ada dimanapun: tides, musim, konsumsi listrik harian, etc.
- **Full periodicity**
 - Setiap titik pada waktunya berkontribusi (secara tepat atau pendekatan) untuk periodicity
- **Partial periodicit:** Notasi umum yang lebih general
 - Hanya beberapa segmen berkontribusi pada periodicity
 - Jim reads NY Times 7:00-7:30 am every week day
- **Cyclic association rules**
 - Asosiasi yang membentuk cycle
- **Metode**
 - Full periodicity: FFT, metode analisis statistika
 - Partial dan cyclic periodicity: variasi dari metode Apriori-like

Ringkasan

- Sequential Pattern Mining berguna dalam banyak aplikasi contohnya weblog analysis, financial market prediction, BioInformatics, dll.
- Pendekatan ini mirip dengan frequent itemsets mining, *tetapi* dengan memperhatikan urutan.
- Pendekatan yang telah dibahas diturunkan dari 2 algoritme yang populer dalam mining frequent itemsets
 - Pembangkitan kandidat : AprioriAll dan GSP
 - Pattern Growth: FreeSpan dan PrefixSpan