Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window

Chih-Hsiang Lin, Ding-Ying Chiu,
Yi-Hung Wu

Department of Computer Science
National Tsing Hua University

Arbee L.P. Chen

Department of Computer Science
National Chengchi University

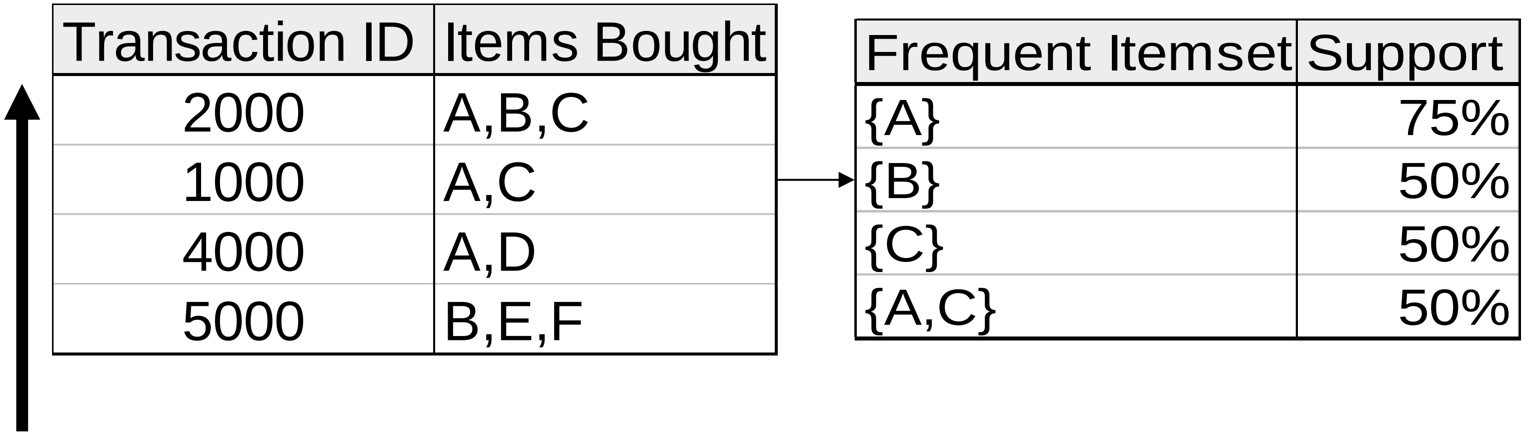
Outline

- ❑ **Introduction**
- ❑ **Related Work**
- ❑ **Our Approach**
  - ➢ Time-sensitive Sliding-window Model
  - ➢ Mining and Discounting
  - ➢ Self-adjusting Discounting Table
- ❑ **Performance Evaluation**
- ❑ **Conclusion**

Introduction

❑ Background

- ➢ Mining frequent itemsets in transaction databases
- ➢ Minimum support threshold



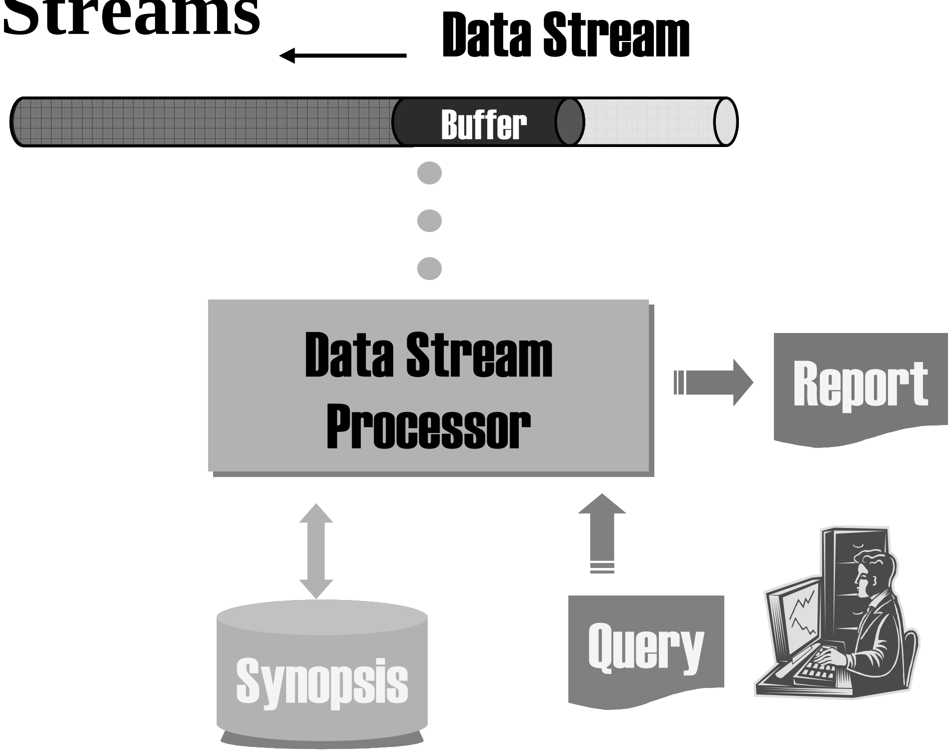
Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

A data stream is formed by transactions arriving in series.

Introduction

- ❑ **Various Forms of Data Streams**
  - ➢ Call detail records
  - ➢ Sensor network data
  - ➢ Web click streams
- ❑ **Three Characteristics**
  - ➢ Continuity
  - ➢ Expiration
  - ➢ Infinity

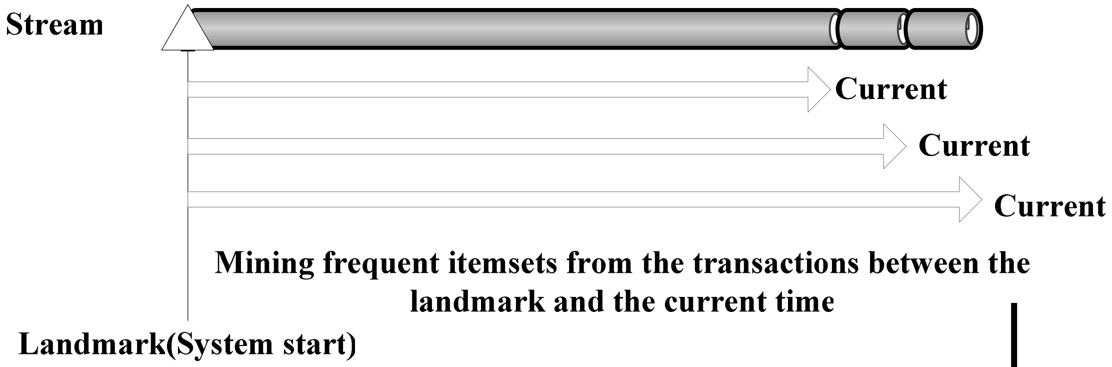


Introduction

- ❑ **Three Requirements**
  - ➢ Time-sensitivity
  - ➢ Approximation
  - ➢ Adaptability
- ❑ **Inability of Traditional Mining Algorithms**
  - ➢ Designed for only static databases
  - ➢ Multiple database scans
  - ➢ No approximate answering
  - ➢ Huge memory consumption

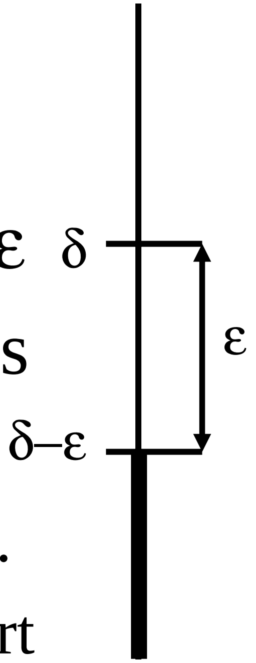
Related Work

❑ Landmark Model


Stream
Current
Current
Current
Mining frequent itemsets from the transactions between the landmark and the current time
Landmark(System start)

❑ Problem Definition in [MM02]

- ➢Given support threshold δ and error parameter ε
- ➢Output a list of itemsets with estimated supports
  - (1) Each itemset with true support ≥ δ is output.
  - (2) Each itemset with true support < δ–ε is not output.
  - (3) True support – ε ≤ estimated support ≤ true support



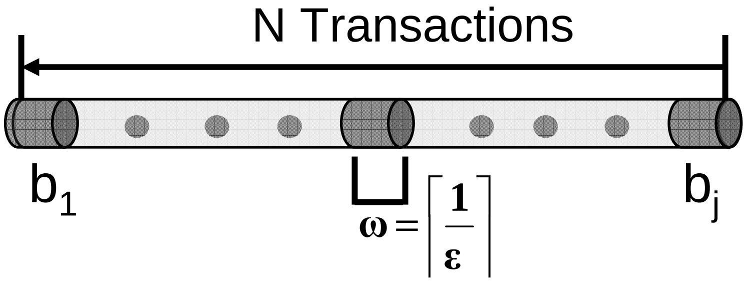
Related Work

❑ **Lossy-counting Algorithm [MM02]**

- ➢ Consider a data stream as a sequence of buckets
- ➢ In each b_j, maintain the set of (e, f, ∇)

  (e: itemset, f: estimated count, ∇: maximum error)

- ➢ Insert new (e, 1, j–1) or update old (e, f+1, ∇)
- ➢ At the end of b_j, delete (e, f, ∇) if $f+\nabla \le j$
  - **True count $\le f+\nabla \le j \le \varepsilon N < \delta N \Rightarrow$ *no false deletion***



b_j	b_1	b_2	b_3	…	b_{j-1}	b_j	…
∇	0	1	2	…	j-2	j-1	…

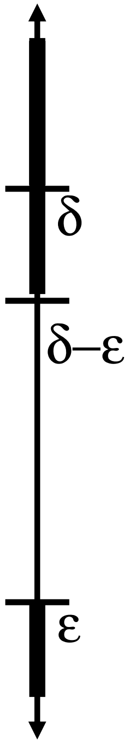
Related Work

❑ Lossy-counting Algorithm (continued)

- ➢ Output (e, f, ∇) if $f \geq (\delta - \varepsilon)N$
  - $f \leq$ **true count** $\leq f + \nabla \leq f + (j-1) \leq f + \varepsilon N$

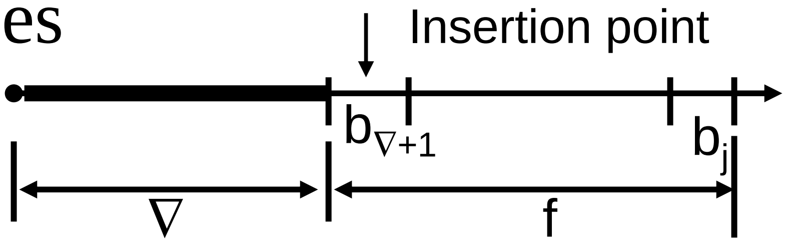
  $\Rightarrow 0 \leq$ **true count** $- f \leq \varepsilon N$ ***(3)***

  $\Rightarrow$ ***(2), (1), no false dismissal***



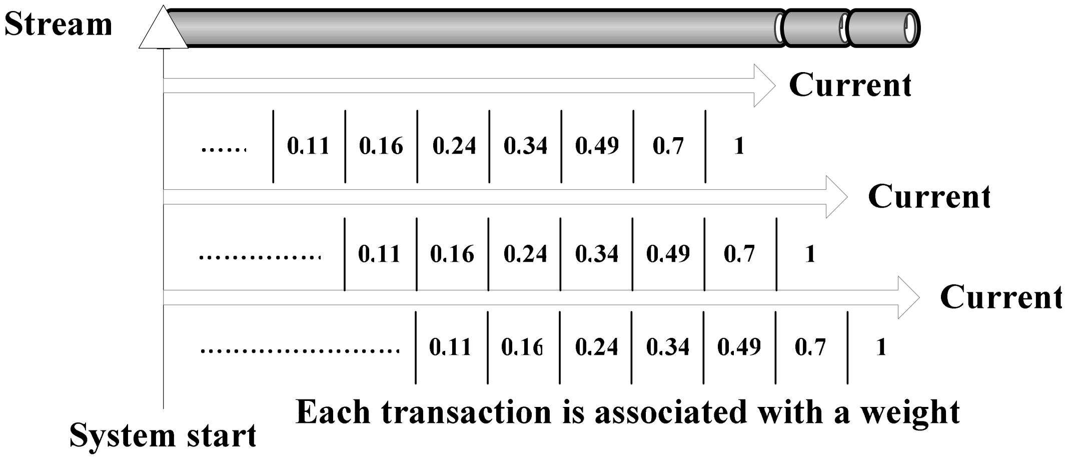
❑ Remarks

- ➢ The arrival time of data is not considered
- ➢ As time goes by, εN increases
- ➢ No adaptability to memory



Related Work

- **Time-fading Model**



Stream

Current

…… 0.11 0.16 0.24 0.34 0.49 0.7 1

Current

…………… 0.11 0.16 0.24 0.34 0.49 0.7 1

Current

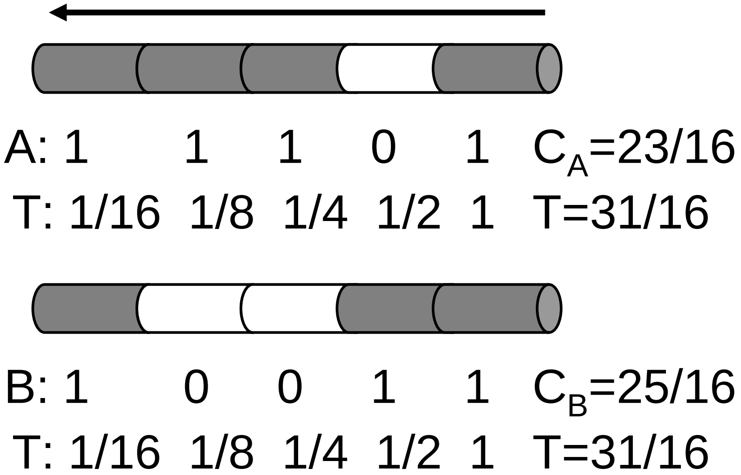
…………………… 0.11 0.16 0.24 0.34 0.49 0.7 1

System start

Each transaction is associated with a weight

- **Decay Rate in [CL03]**
  - $d=b^{-(1/h)}$, $b>1$, $h\geq 1$, $b^{-1}\leq d<1$
    - $C_N=C_{N-1}\times d + 1$ **(or 0)**
    - $T_N=T_{N-1}\times d + 1$

$\Rightarrow T_N \rightarrow 1/(1\text{-}d)$ **as** $N\rightarrow\infty$



A: 1 1 1 0 1 C_A=23/16

T: 1/16 1/8 1/4 1/2 1 T=31/16

B: 1 0 0 1 1 C_B=25/16

T: 1/16 1/8 1/4 1/2 1 T=31/16

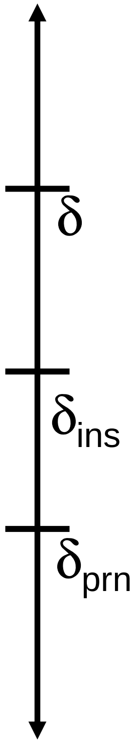
Related Work

❑ **estDec Method [CL03]**

- ➢ For each transaction, maintain the set of (e, f, ∇, tid)
- ➢ Update old (e, f, ∇, tid); Delete if f < δ_{prn}
- ➢ Insert (e, f, ∇, tid) if (e is 1-itemset) or f ≥ δ_{ins}
  - **Estimate the count of a new k-itemset based on the counts of all its (k-1)-subsets: an example**
- ➢ Output (e, f, ∇) if f ≥ δ

❑ **Remarks**

- ➢ δ_{ins} and δ_{prn} are significant to the performance
- ➢ No adaptability to memory



Related Work

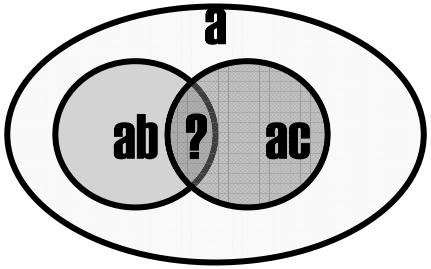
- ❑ **estDec Method (example** e=abc**)**
  - ➢Given f_{ab}, f_{ac}, f_{bc}, f_a, f_b, f_c, estimate f_{abc} and ∇_{abc}

$$f_{abc}=C_{max}^{abc}=\min\{f_{ab}, f_{ac}, f_{bc}\}$$

$$C_{min}^{ab\cup ac}=\max\{0, f_{ab}+f_{ac}-f_a\}$$

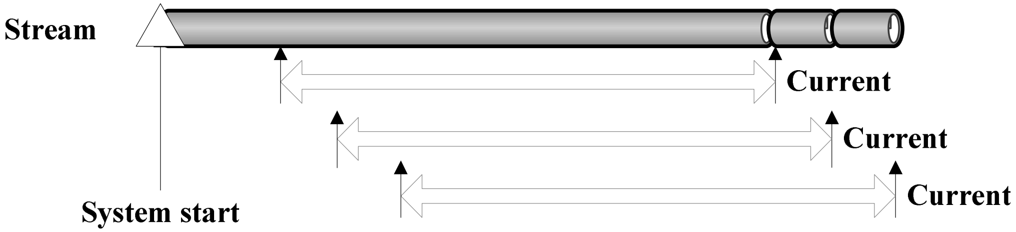
$$C_{min}^{abc}=\max\{C_{min}^{ab\cup ac}, C_{min}^{ab\cup bc}, C_{min}^{ac\cup bc}\}$$

$$\nabla_{abc}=C_{max}^{abc}-C_{min}^{abc}$$



Time-sensitive Sliding-window Model

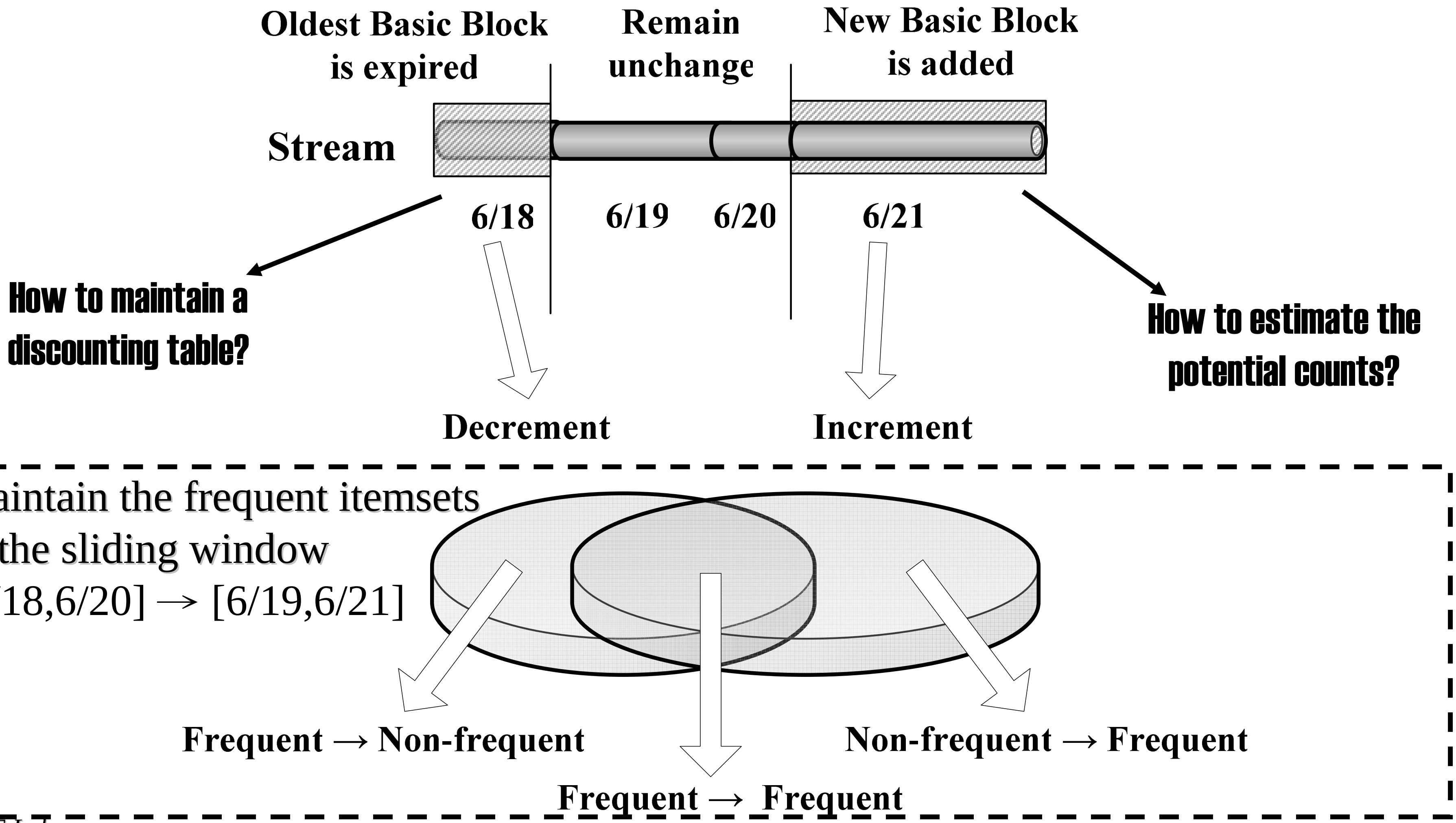
- ❑ **Sliding-window Model**



Mining frequent itemsets from only the latest W transactions

- ❑ **Our Goals**
  - ➢Time-sensitive sliding-window model
    - **Divide the data stream into blocks by time**
  - ➢Fast mining and discounting method
  - ➢Self-adjusting discounting table
    - **Guarantees: No false dismissal or No false alarm**

Time-sensitive Sliding-window Model



Oldest Basic Block is expired

Remain unchange

New Basic Block is added

Stream

6/18 6/19 6/20 6/21

How to maintain a discounting table?

How to estimate the potential counts?

Decrement

Increment

Maintain the frequent itemsets in the sliding window [6/18,6/20] → [6/19,6/21]

Frequent → Non-frequent

Non-frequent → Frequent

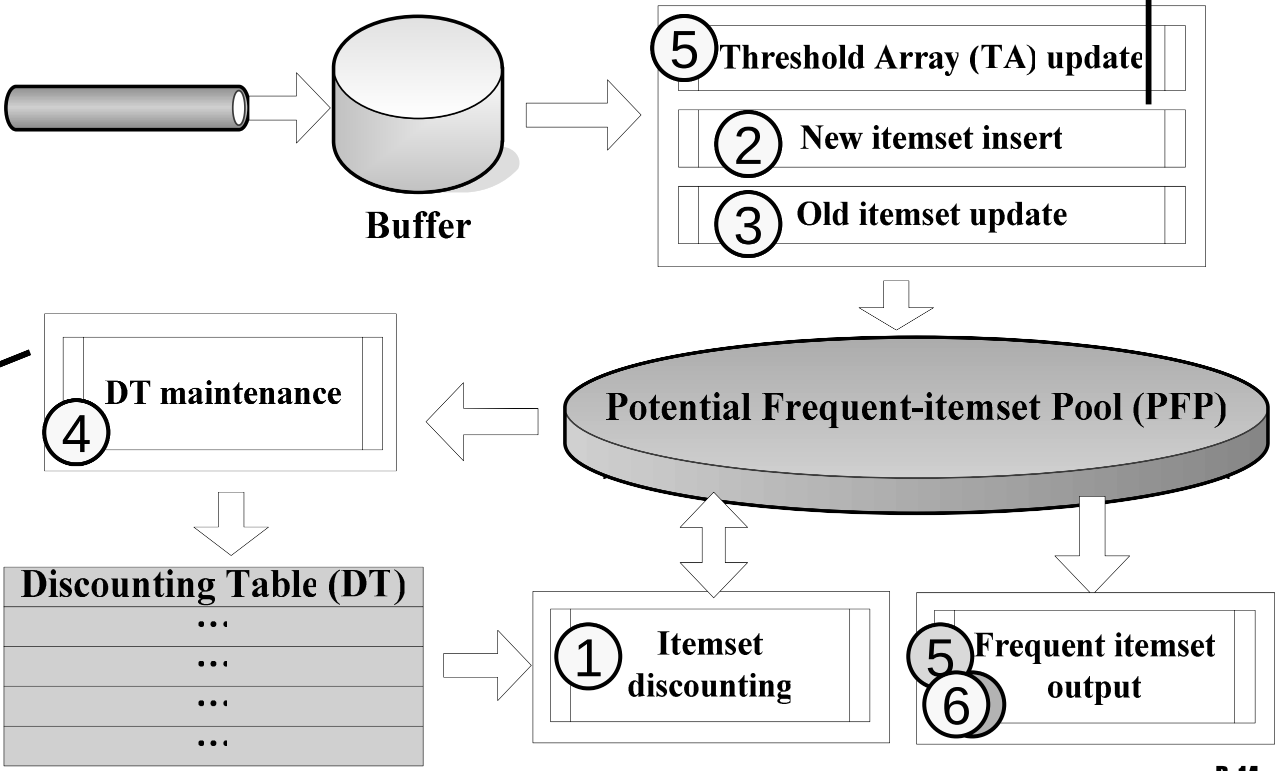
Frequent → Frequent

Mining and Discounting

Estimate potential counts by TA

- ❑ **Accuracy Guarantees of Outputs**
  - ➢ No false dismissal
  - ➢ No false alarm

Ignore potential counts



Buffer

5 Threshold Array (TA) update

2 New itemset insert

3 Old itemset update

Potential Frequent-itemset Pool (PFP)

4 DT maintenance

Discounting Table (DT)

Discounting Table (DT)
…
…
…
…

1 Itemset discounting

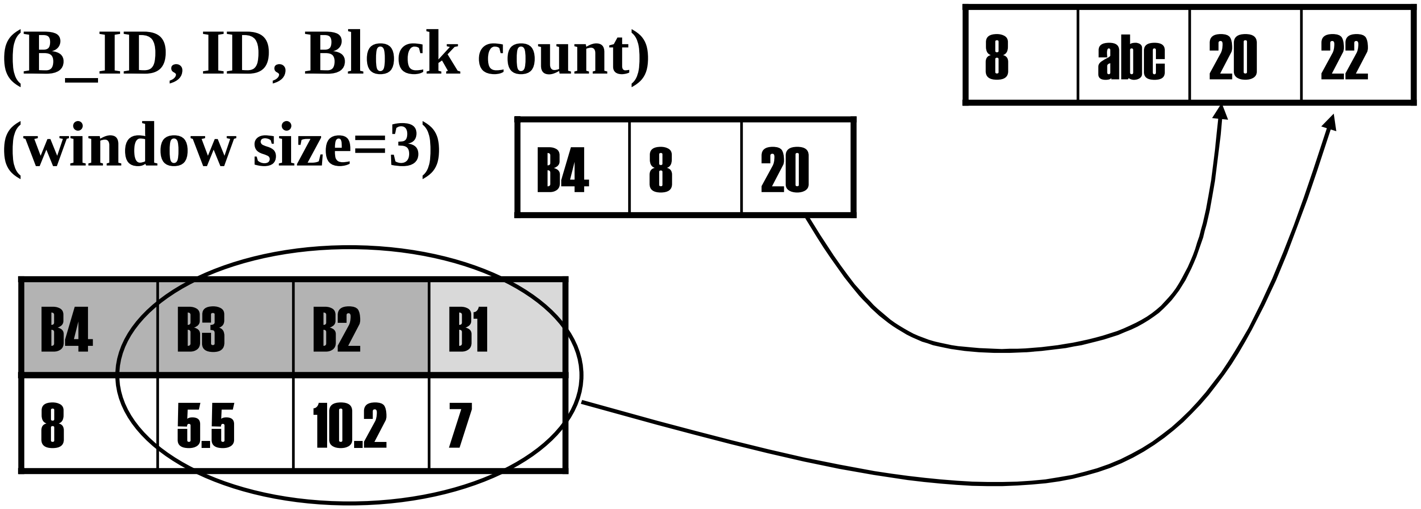
5 6 Frequent itemset output

Self-adjustment:
Merge by min/max functions

Mining and Discounting

❑ Main Storage Formats

- ➢ Ex. abc is a new frequent itemset in B4
  - PFP (ID, Items, Actual count, Potential count)
  - DT (B_ID, ID, Block count)
  - TA (window size=3)



8	abc	20	22

B4	8	20

B4	B3	B2	B1
8	5.5	10.2	7

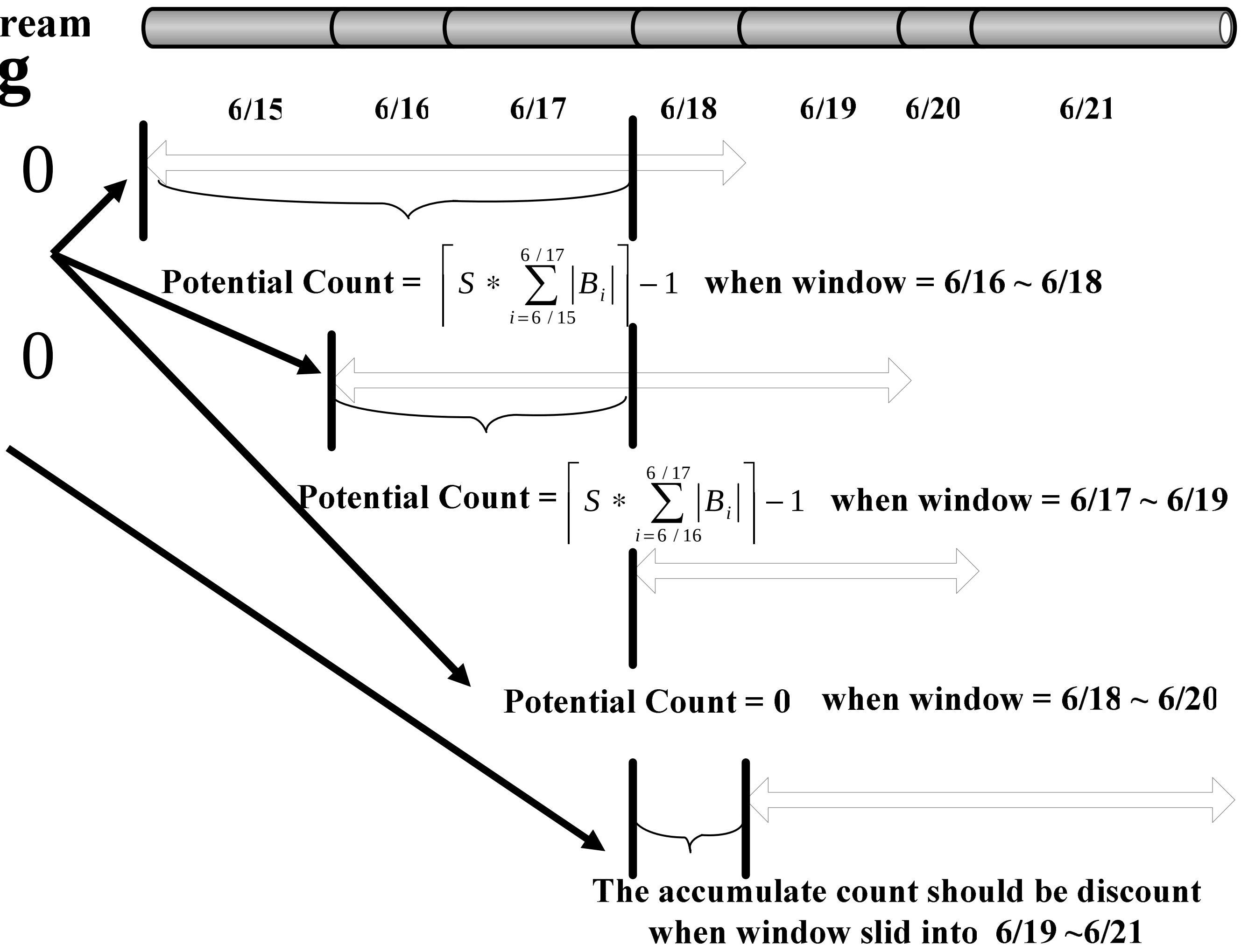
❑ Remark

- ➢ The potential count cannot bound the maximum error if only two thresholds (B2 and B3) are considered.

Mining and Discounting

❑ **Discounting**

- ➢ Pcount > 0
  - **By TA**
- ➢ Pcount = 0
  - **By DT**



Potential Count = $\left\lceil S * \sum_{i=6/15}^{6/17} |B_i| \right\rceil - 1$ **when window = 6/16 ~ 6/18**

Potential Count = $\left\lceil S * \sum_{i=6/16}^{6/17} |B_i| \right\rceil - 1$ **when window = 6/17 ~ 6/19**

Potential Count = 0 when window = 6/18 ~ 6/20

The accumulate count should be discount when window slid into 6/19 ~6/21

Mining and Discounting

- ❑ An Example (threshold=0.4, window size=3)

	Time period	Number of transactions	Frequent Itemset in a block (and its count)
B_1	09:00~09:59	27	a(11),b(20),c(2),ab(6)
B_2	10:00~10:59	20	a(20),c(13),ac(13)
B_3	11:00~11:59	27	a(19),b(8),c(7),ac(7)
B_4	12:00~12:59	23	a(10),c(3),d(10)

- ❑ After B_1 passes

TA

10.8	0	0	0

DT

B_ID	ID	Bcount
1	1	11
1	2	20

PFP

(1,a,11,0) (2,b,20,0)

Mining and Discounting

	Number of transactions	Frequent Itemset in a block (and its count)
B_1	27	a(11),b(20),c(2),ab(6)
B_2	20	a(20),c(13),ac(13)
B_3	27	a(19),b(8),c(7),ac(7)
B_4	23	a(10),c(3),d(10)

- After B_2 passes

TA	8	10.8	0	0

DT

B_ID	ID	Bcount
1	1	11
1	2	20
2	1	20
2	2	0
2	3	13
2	4	13

PFP (1,a,31,0) (2,b,20,0) (3,c,13,10) (4,ac,13,10)

Mining and Discounting

	Number of transactions	Frequent Itemset in a block (and its count)
B_1	27	a(11),b(20),c(2),ab(6)
B_2	20	a(20),c(13),ac(13)
B_3	27	a(19),b(8),c(7),ac(7)
B_4	23	a(10),c(3),d(10)

- ❑ After B_3 passes

TA

10.8	8	10.8	0

DT

B_ID	ID	Bcount
1	1	11
1	2	20
2	1	20
2	2	0
2	3	13
2	4	13
3	1	19
3	3	7
3	4	7

PFP (1,a,50,0) (3,c,20,10) (4,ac,20,10)

Mining and Discounting

	Number of transactions	Frequent Itemset in a block (and its count)
B_1	27	a(11),b(20),c(2),ab(6)
B_2	20	a(20),c(13),ac(13)
B_3	27	a(19),b(8),c(7),ac(7)
B_4	23	a(10),c(3),d(10)

- After B_4 passes

TA				
	9.2	10.8	8	10.8

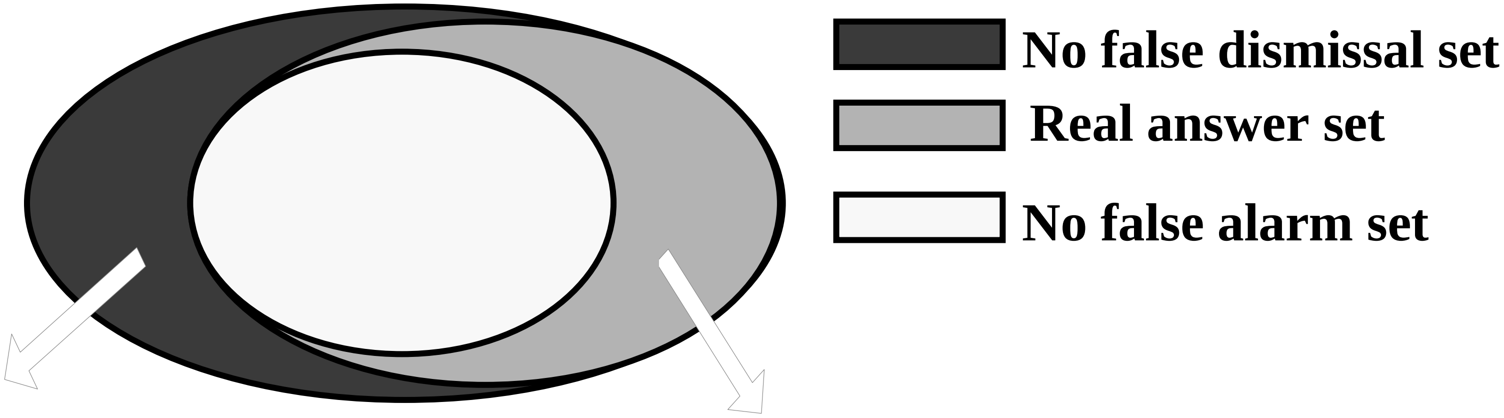
DT

B_ID	ID	Bcount
2	1	20
2	2	0
2	3	13
2	4	13
3	1	19
3	3	7
3	4	7
4	1	10
4	2	10

PFP (1,a,49,0) (2,d,10,29)

Mining and Discounting

- ❑ Accuracy Guarantees



No false dismissal set

Real answer set

No false alarm set

False alarms because of considering potential count

False dismissals because of not considering potential count

Self-adjusting Discounting Table

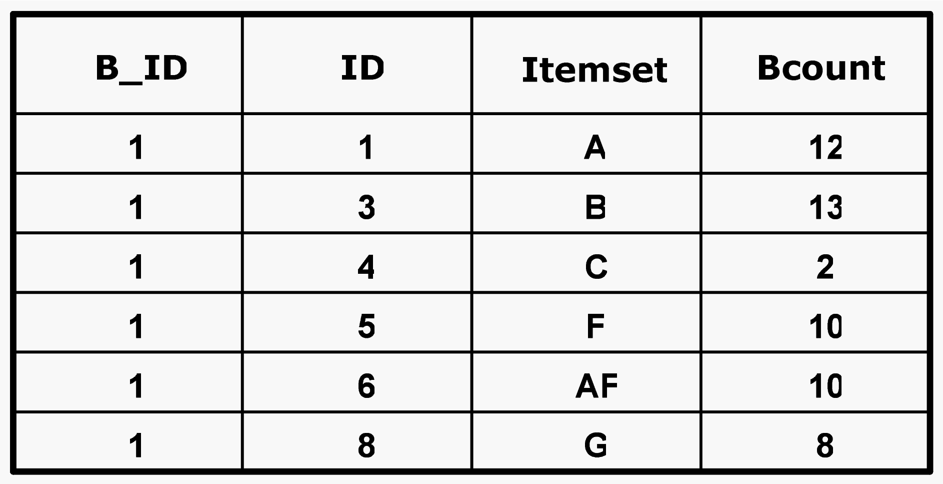
❑ **Requirements**

- ➢A huge number of itemsets → limited memory
- ➢Providing approximate support counts
- ➢Still keep the accuracy guarantees

❑ **Rationale**

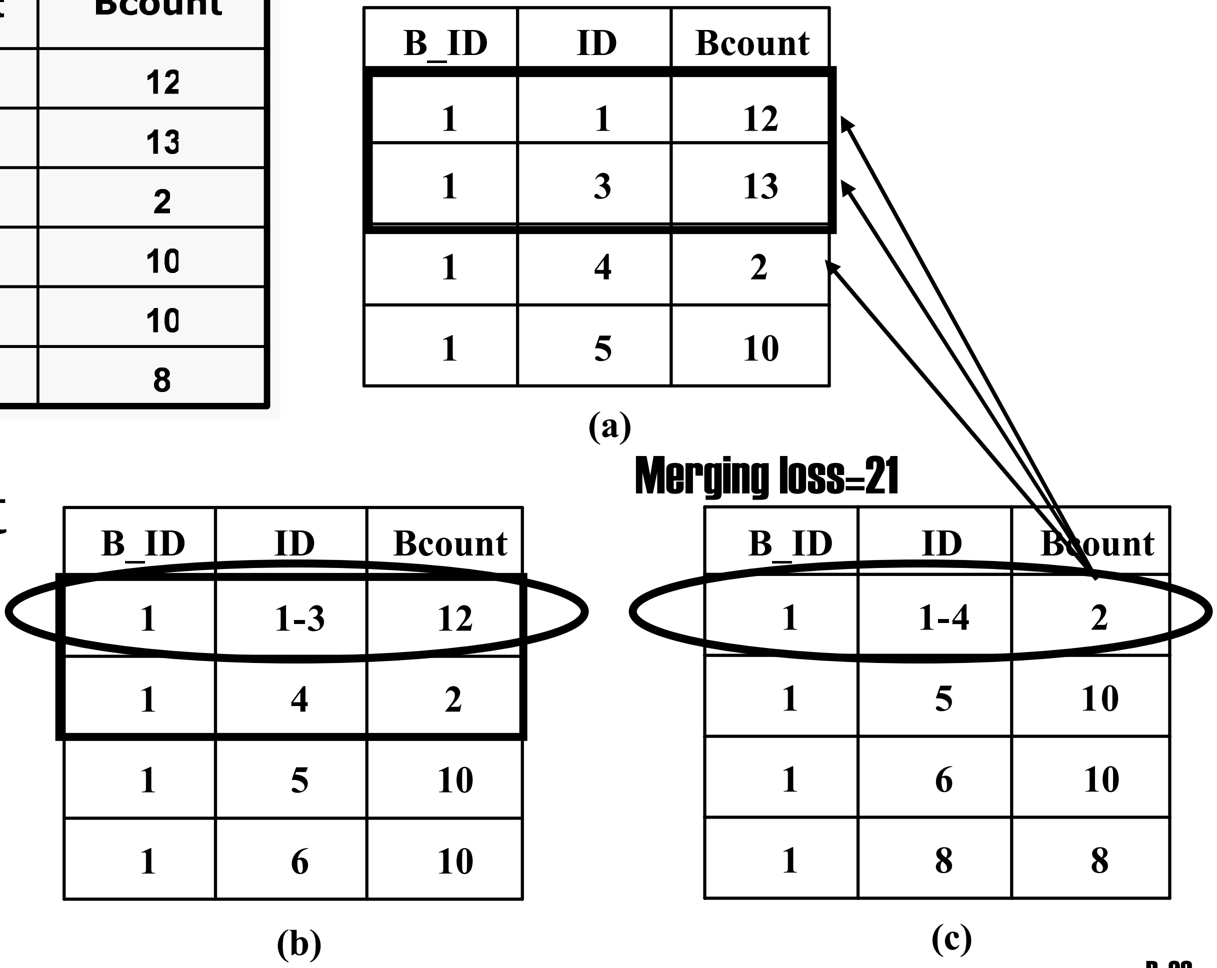
- ➢Merge different entries of DT (different itemsets) into one and represent their support counts by using the minimum/maximum support counts in them.

Self-adjusting Discounting Table



B_ID	ID	Itemset	Bcount
1	1	A	12
1	3	B	13
1	4	C	2
1	5	F	10
1	6	AF	10
1	8	G	8

- ❑ **Naïve Adjustment**
  - ➢ Merge the first two entries
  - ➢ Ex. DT_limit=4



B_ID	ID	Bcount
1	1	12
1	3	13
1	4	2
1	5	10

(a)

Merging loss=21

B_ID	ID	Bcount
1	1-3	12
1	4	2
1	5	10
1	6	10

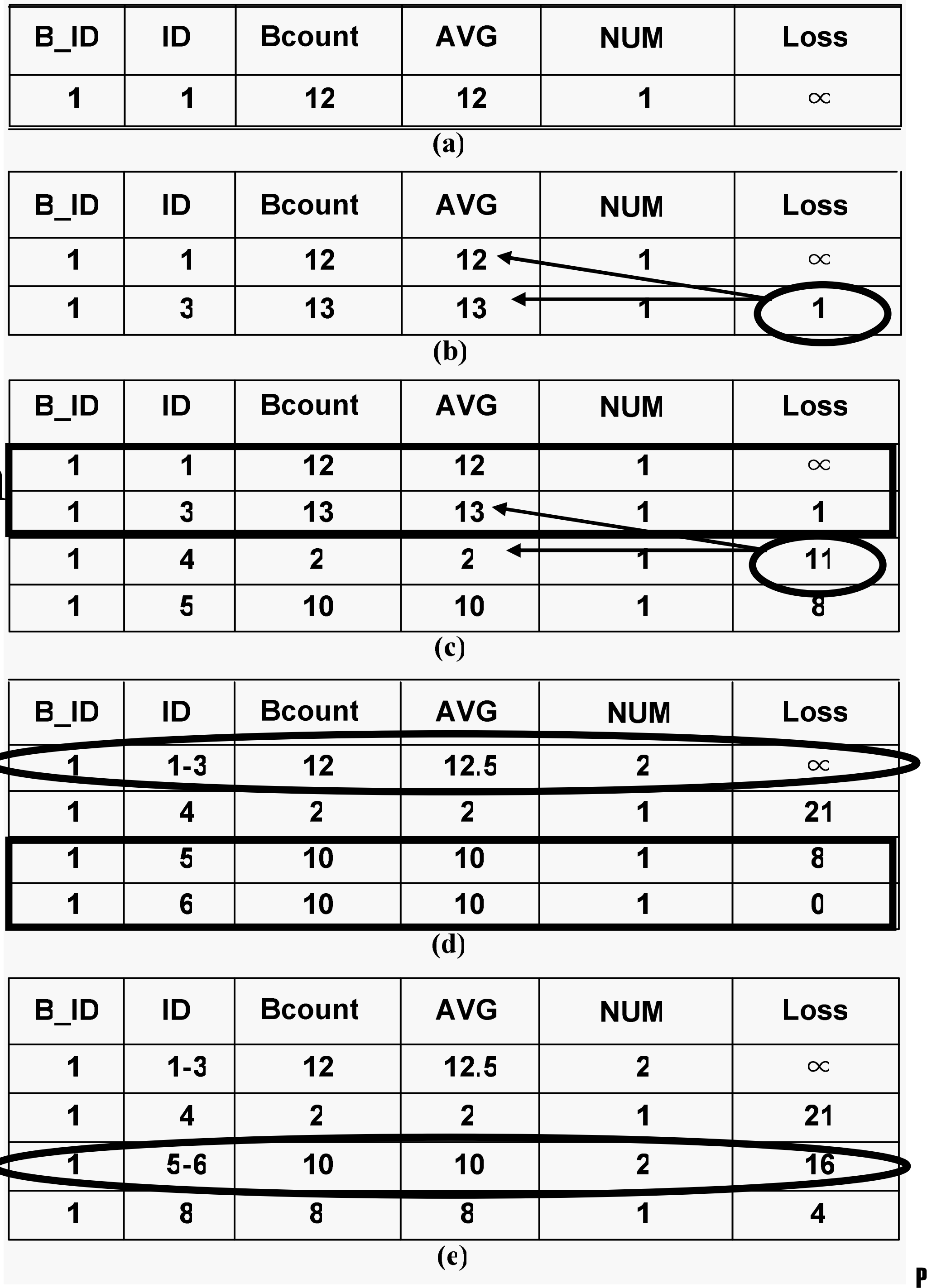
(b)

B_ID	ID	Bcount
1	1-4	2
1	5	10
1	6	10
1	8	8

(c)

❑ Selective Adjustment

- ➢ Merge the entry with the smallest merging loss with the entry above it
- ➢ Ex. DT_limit=4



B_ID	ID	Bcount	AVG	NUM	Loss
1	1	12	12	1	∝

(a)

B_ID	ID	Bcount	AVG	NUM	Loss
1	1	12	12	1	∝
1	3	13	13	1	1

(b)

B_ID	ID	Bcount	AVG	NUM	Loss
1	1	12	12	1	∝
1	3	13	13	1	1
1	4	2	2	1	11
1	5	10	10	1	8

(c)

B_ID	ID	Bcount	AVG	NUM	Loss
1	1-3	12	12.5	2	∝
1	4	2	2	1	21
1	5	10	10	1	8
1	6	10	10	1	0

(d)

B_ID	ID	Bcount	AVG	NUM	Loss
1	1-3	12	12.5	2	∝
1	4	2	2	1	21
1	5-6	10	10	2	16
1	8	8	8	1	4

(e)

Merging loss=1

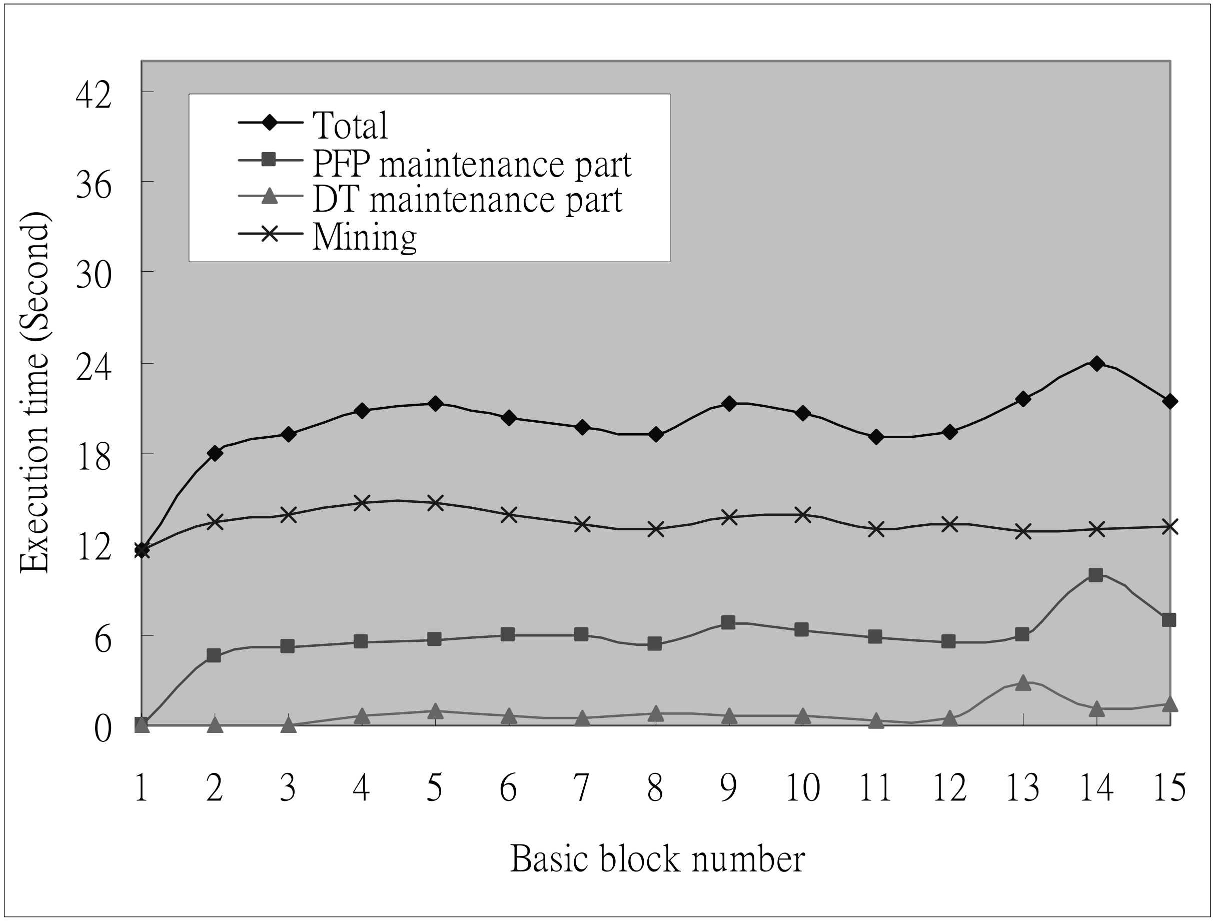
Performance Evaluation

❑ Experimental Setting

Parameter	**Value**
Number of distinct items	1K
DT_limit	10K
θ (support threshold)	0.0025
|W| (window size)	4
T (average transaction length)	3~7
I (the average length of the maximum pattern)	4
D (the total number of transactions)	150K

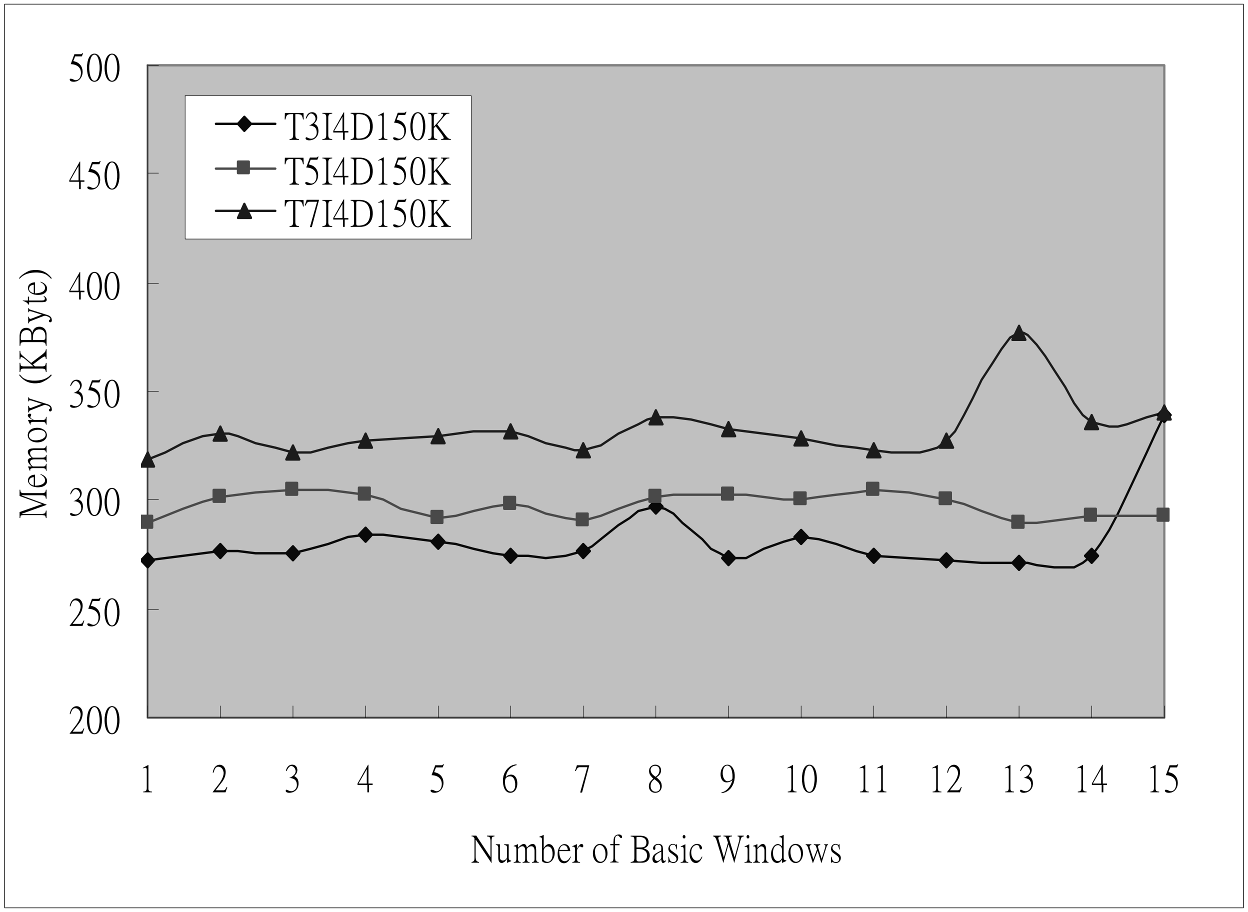
Performance Evaluation

- ❑ **Time Efficiency**
  - ➢ T=7



Performance Evaluation

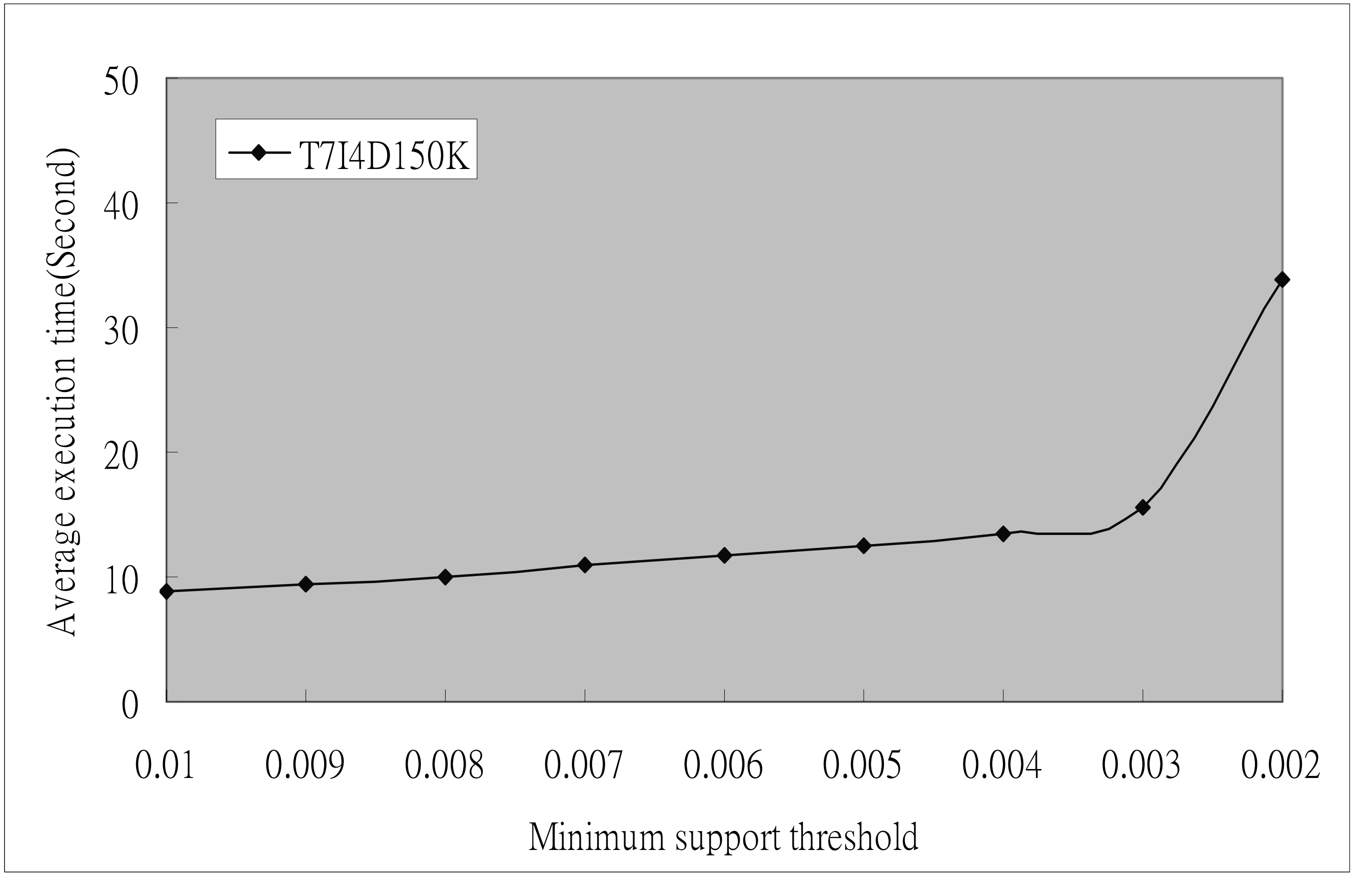
❑ Space Efficiency



MAKE Lab

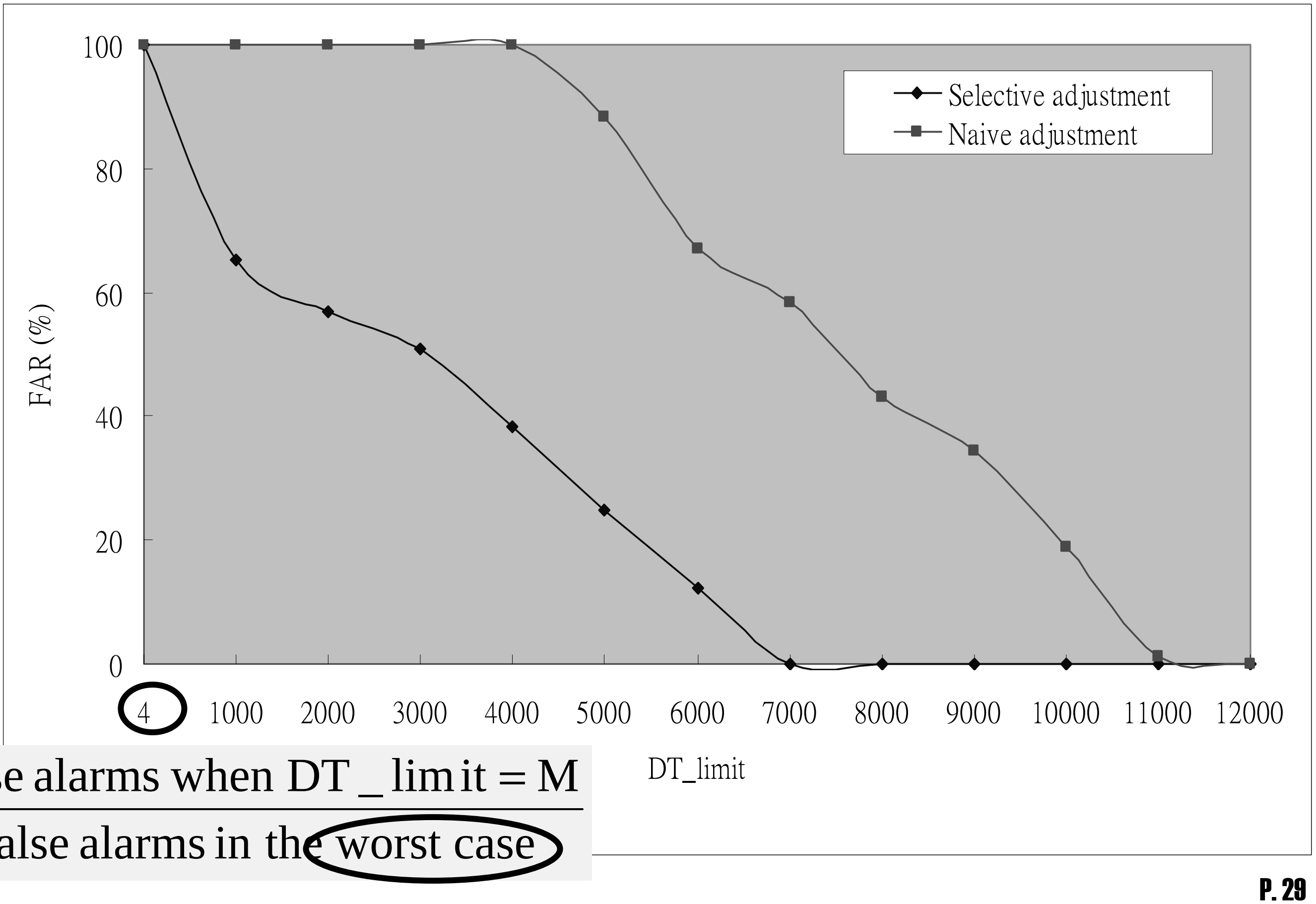
Performance Evaluation

❑ Scalability



Performance Evaluation

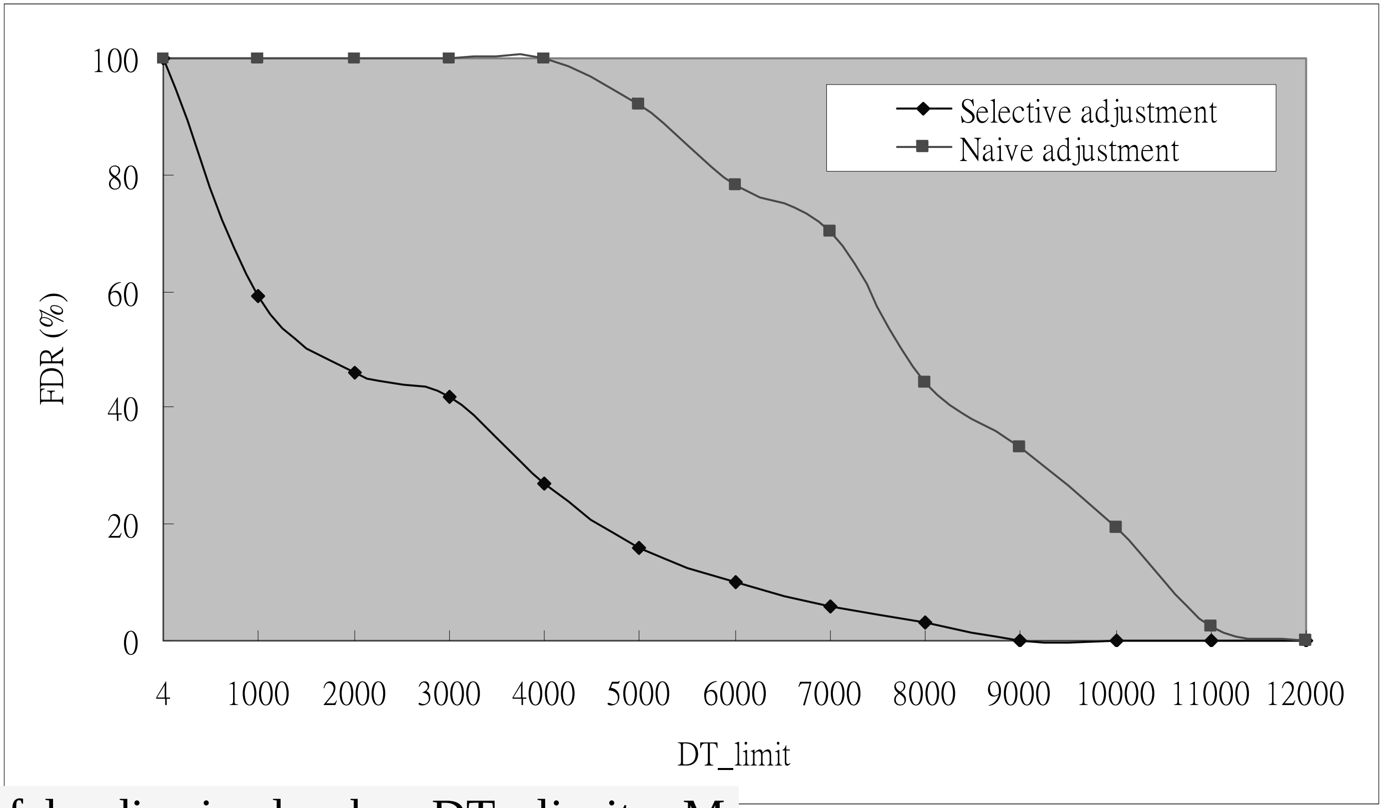
❑ Effectiveness on No False Dismissal



$$FAR_{M} = \frac{\text{The number of false alarms when } DT\_\lim it = M}{\text{The number of false alarms in the worst case}}$$

Performance Evaluation

❑Effectiveness on No False Alarm



$$\mathrm{FDR}_{M} = \frac{\text{The number of false dismissals when DT\_limit} = \text{M}}{\text{The number of false dismissals in the worst case}}$$

Conclusion

❑ **Our Contributions**

- An efficient algorithm for mining frequent itemsets over data streams under the time-sensitive sliding-window model
- Data structures and methods for mining and discounting the support counts of the frequent itemsets when the window slides
- Two strategies for maintaining the self-adjusting discounting table under the limited memory

Conclusion

❑ **Future Works**

- ➢ The error estimation that can help the ranking of frequent itemsets if only the top-k frequent itemsets are needed
- ➢ The other types of frequent patterns such as the sequential patterns
- ➢ The constraints recently discussed in the data mining field such as the closed frequent patterns

Thank You!

Reference

- [BBD+02] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and Issues in Data Stream Systems,” *ACM PODS Conference*, 2002.
- [CL03] J.H. Chang and W.S. Lee, “Finding Recent Frequent Itemsets Adaptively over Online Data Streams,” *ACM SIGKDD Conference*, 2003.
- [CS03] E. Cohen and M. Strauss, “Maintaining Time-Decaying Stream Aggregates,” *ACM PODS Conference*, 2003.
- [DGIM02] M. Datar, A. Gionis, P. Indyk, and R. Motwani, “Maintaining Stream Statistics over Sliding Windows,” *ACM-SIAM Symposium on Discrete Algorithms*, 2002.
- [GGR01] V. Ganti, J. Gehrke, and R. Ramakrishnan, “Demon: Mining and Monitoring Evolving Data,” *IEEE Transactions on Data Engineering*, 13(1), 2001.
- [GGR02] V. Ganti, J. Gehrke, and R. Ramakrishnan, “Mining Data Streams under Block Evolution,” *ACM SIGKDD Explorations*, 3(2), 2002.

Reference

- [GÖ03] L. Golab and M. T. Özsu, “Issues in Data Stream Management,” *ACM SIGMOD Record*, 32(2), 2003.
- [MM02] G. Manku and R. Motwani, “Approximate Frequency Counts over Data Streams,” *VLDB Conference*, 2002.
- [TCY03] W.G. Teng, M.S. Chen, and P.S. Yu, “A Regression-based Temporal Pattern Mining Scheme for Data Streams,” *VLDB Conference*, 2003.
- [TCY04] W.G. Teng, M.S. Chen, and P.S. Yu, “Resource-Aware Mining with Variable Granularities in Data Streams,” *SIAM Conference on Data Mining*, 2004.
- [YCLZ04] J.X. Yu, Z. Chong, H. Lu, and A. Zhou, “False Positive or False Negative: Mining Frequent Itemsets from High Speed Transactional Data Streams,” *VLDB Conference*, 2004.
- [ZS03] Y. Zhu and D. Shasha, “Efficient Elastic Burst Detection in Data Streams,” *ACM SIGKDD Conference*, 2003.