# Session 6: Mining Colossal Patterns

# Mining Long Patterns: Challenges

- Mining long patterns is needed in bioinformatics, social network analysis, software engineering, …
  - But the methods introduced so far mine only short patterns (e.g., length < 10)
- Challenges of mining long patterns
  - The curse of "downward closure" property of frequent patterns
    - Any sub-pattern of a frequent pattern is frequent
    - If $\{a_1, a_2, \ldots, a_{100}\}$ is frequent, then $\{a_1\}, \{a_2\}, \ldots, \{a_{100}\}, \{a_1, a_2\}, \{a_1, a_3\}, \ldots, \{a_1, a_{100}\}, \{a_1, a_2, a_3\}, \ldots$ are all frequent! There are about $2^{100}$ such frequent itemsets!
  - No matter searching in breadth-first (e.g., Apriori) or depth-first (e.g., FPgrowth), if we still adopt the "small to large" paradigm, we have to examine so many patterns, which leads to combinatorial explosion!

# Colossal Patterns: A Motivating Example

**$T_1$ = 2 3 4 ….. 39 40**
**$T_2$ = 1 3 4 ….. 39 40**
**: .**
**: .**
**: .**
**: .**
**$T_{40}$=1 2 3 4 …… 39**
**$T_{41}$= 41 42 43 ….. 79**
**$T_{42}$= 41 42 43 ….. 79**
**: .**
**: .**
**$T_{60}$= 41 42 43 … 79**

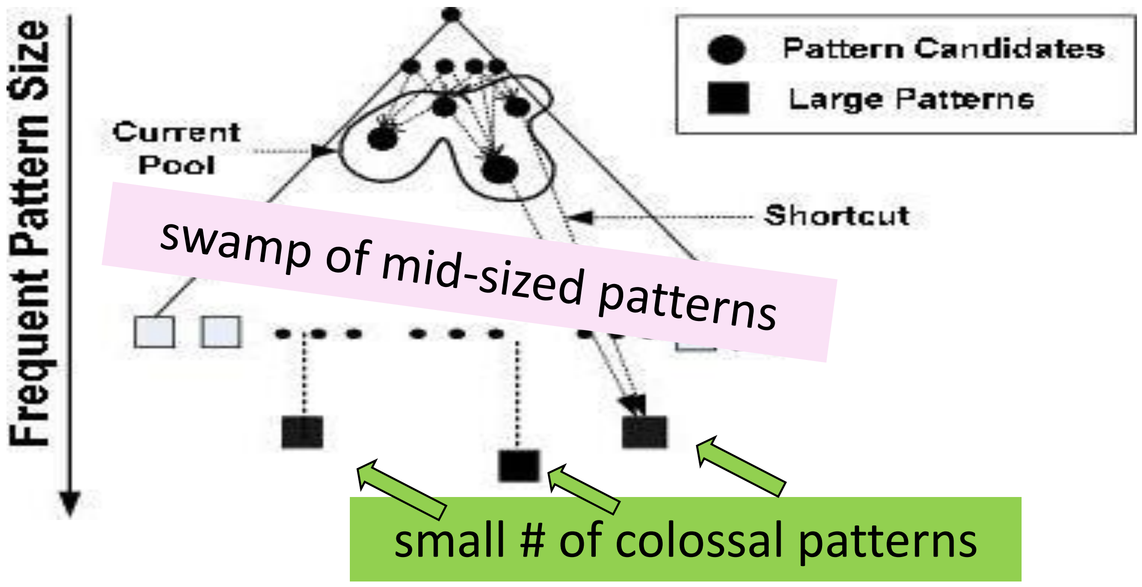
- Let min-support σ= 20
- # of closed/maximal patterns of size 20: $\binom{40}{20}$
- But there is only one pattern with size close to 40 (*i.e., long* or *colossal*)
  - α= {41,42,…,79} of size 39
- Q: How to find it without generating an exponential number of size-20 patterns?

The existing fastest mining algorithms (*e.g.,* FPClose, LCM) fail to complete running

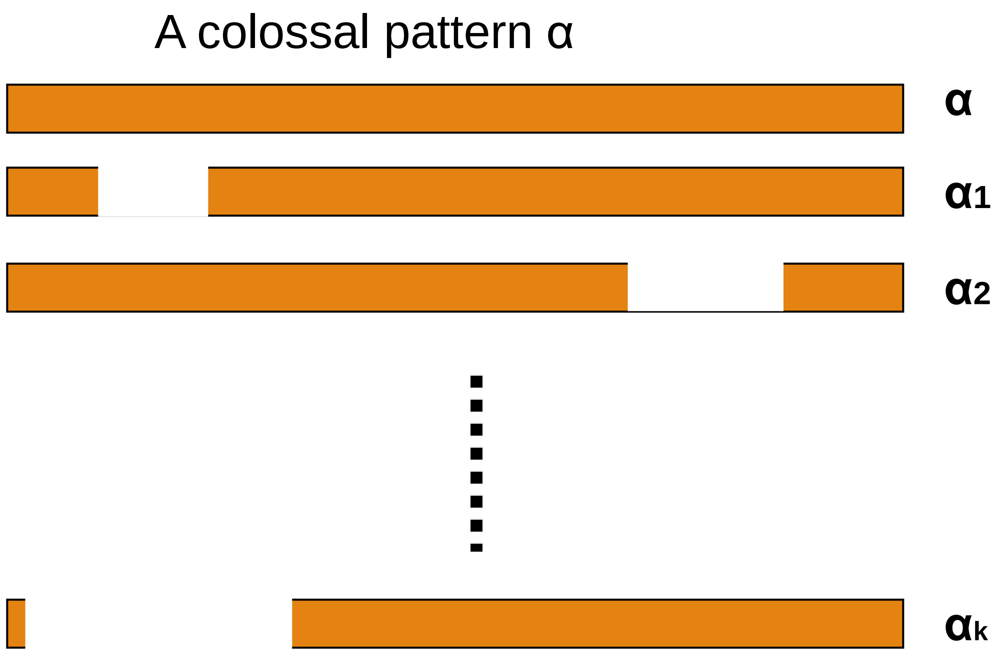
A new algorithm, *Pattern-Fusion*, outputs this colossal pattern in seconds

# What Is Pattern-Fusion?

- Not strive for completeness (why?)
- Jump out of the swamp of the mid-sized intermediate "results"
- Strive for mining almost complete and representative colossal patterns: identify "short-cuts" and take "leaps"
- Key observation
  - The larger the pattern or the more distinct the pattern, the greater chance it will be generated from small ones
- Philosophy: Collection of small patterns hints at the larger patterns
- Pattern fusion strategy: Fuse small patterns together in one step to generate new pattern candidates of significant sizes



# Observation: Colossal Patterns and Core Patterns




Subpatterns $\alpha_1$ to $\alpha_k$ cluster tightly around the colossal pattern α by sharing a similar support. Such subpatterns are *core patterns* of α

- A colossal pattern has far more core patterns than a small-sized pattern
- A colossal pattern has far more core descendants of a smaller size c
- A random draw from a complete set of pattern of size c would be more likely to pick a core descendant of a colossal pattern
- A colossal pattern can be generated by merging a set of core patterns

# Robustness of Colossal Patterns

- Core Patterns: For a frequent pattern α, a subpattern β is a τ-core pattern of α if β shares a similar support set with α, i.e.,

  $$\frac{|D_\alpha|}{|D_\beta|} \geq \tau \qquad 0 < \tau \leq 1$$ where τ is called the core ratio

- (d,τ)-robustness: A pattern α is *(d, τ)-robust* if *d* is the maximum number of items that can be removed from α for the resulting pattern to remain a τ-core pattern of α
- For a (d,τ)-robust pattern α, it has $\Omega(2^d)$ core patterns
- Robustness of Colossal Patterns: A colossal pattern tends to have much more core patterns than small patterns
- Such core patterns can be clustered together to form "dense balls" based on pattern distance defined by

  $$Dist(\alpha,\beta) = 1 - \frac{|D_\alpha \cap D_\beta|}{|D_\alpha \cup D_\beta|}$$

A random draw in the pattern space will hit somewhere in the ball with high probability

# The Pattern-Fusion Algorithm

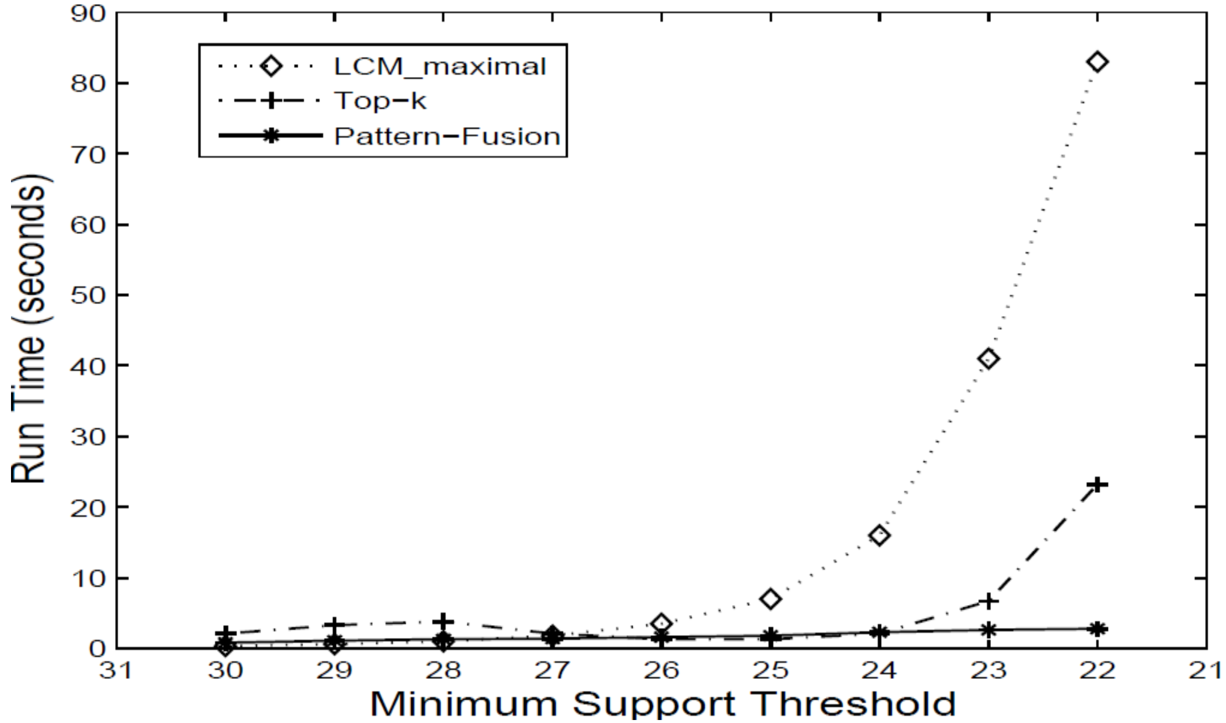
- Initialization (Creating initial pool): Use an existing algorithm to mine all frequent patterns up to a small size, e.g., 3
- Iteration (Iterative Pattern Fusion):
  - At each iteration, K seed patterns are randomly picked from the current pattern pool
  - For each seed pattern thus picked, we find all the patterns within a bounding ball centered at the seed pattern
  - All these patterns found are fused together to generate a set of super-patterns
  - All the super-patterns thus generated form a new pool for the next iteration
- Termination: when the current pool contains no more than K patterns at the beginning of an iteration

# Experimental Results on Data Set: ALL

- ALL: A popular gene expression clinical data set on ALL-AML leukemia, with 38 transactions, each with 866 columns. There are 1736 items in total.
  - When minimum support is high (e.g., 30), Pattern-Fusion gets all the largest colossal patterns with size greater than 85

| Pattern Size | 110 | 107 | 102 | 91 | 86 | 84 | 83 |
|---|---|---|---|---|---|---|---|
| The complete set | 1 | 1 | 1 | 1 | 1 | 2 | 6 |
| Pattern-Fusion | 1 | 1 | 1 | 1 | 1 | 1 | 4 |
| Pattern Size | 82 | 77 | 76 | 75 | 74 | 73 | 71 |
| The complete set | 1 | 2 | 1 | 1 | 1 | 2 | 1 |
| Pattern-Fusion | 0 | 2 | 0 | 1 | 1 | 1 | 1 |

Mining colossal patterns on a Leukemia dataset




Algorithm runtime comparison on another dataset

# Summary of the Lecture

- Efficient methods have been developed for mining various kinds of patterns
  - Mining Multiple-Level Associations
  - Mining Multi-Dimensional Associations
  - Mining Quantitative Associations
  - Mining Negative Correlations
  - Mining Compressed and Redundancy-Aware Patterns
  - Mining Long/Colossal Patterns

# Recommended Readings


- R. Srikant and R. Agrawal, “Mining generalized association rules”, VLDB'95
- Y. Aumann and Y. Lindell, “A Statistical Theory for Quantitative Association Rules”, KDD'99
- D. Xin, J. Han, X. Yan and H. Cheng, "On Compressing Frequent Patterns", Knowledge and Data Engineering, 60(1): 5-29, 2007
- D. Xin, H. Cheng, X. Yan, and J. Han, "Extracting Redundancy-Aware Top-K Patterns", KDD'06
- F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng, “Mining Colossal Frequent Patterns by Core Pattern Fusion”, ICDE'07
- J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent Pattern Mining: Current Status and Future Directions", Data Mining and Knowledge Discovery, 15(1): 55-86, 2007