Chapter 7. Advanced Frequent Pattern Mining: Diverse Patterns

Meng Jiang

CSE 40647/60647 Data Science Fall 2017

Introduction to Data Mining

Research on Pattern Mining: A Road Map

- Research on Pattern Mining
  - Kinds of patterns and rules
    - Basic Patterns
      - frequent pattern
      - association rule
      - closed/max pattern
      - generator
    - Multilevel & Multidimensional Patterns
      - multilevel (uniform, varied, or itemset -based support)
      - multidimensional pattern (incl . high-dimensional pattern)
      - continuous data (discretization -based, or statistical)
    - Extended Patterns
      - approximate pattern
      - uncertain pattern
      - compressed pattern
      - rare pattern/negative pattern
      - high-dimensional and colossal patterns
  - Mining Methods
    - Basic Mining Methods
      - candidate generation (Apriori , partitioning, sampling, ...)
      - Pattern growth (FPgrowth , HMine , FPMax, Closet+, ...)
      - vertical format (EClat , CHARM, ...)
    - Mining Interesting Patterns
      - interestingness (subjective vs. objective)
      - constraint-based mining
      - correlation rules
      - exception rules
    - Distributed, parallel & incremental
      - distributed/parallel mining
      - incremental mining
      - stream pattern
  - Extensions & Application
    - Extended Data Types
      - sequential ad time-series patterns
      - structural (e.g., tree, lattice, graph) patterns
      - spatial (e.g., co-location) pattern
      - temporal (evolutionary, periodic)
      - image, video and multimedia patterns
      - network patterns
    - Applications
      - pattern-based classification
      - pattern-based clustering
      - pattern-based semantic annotation
      - collaborative filtering
      - privacy-preserving

Advanced Frequent Pattern Mining

- **Mining Diverse Patterns**
- Constraint-Based Frequent Pattern Mining
- Sequential Pattern Mining
- Graph Pattern Mining

Mining Diverse Patterns

- Mining Multiple-Level Associations
- Mining Multi-Dimensional Associations
- Mining Quantitative Associations
- Mining Negative Correlations

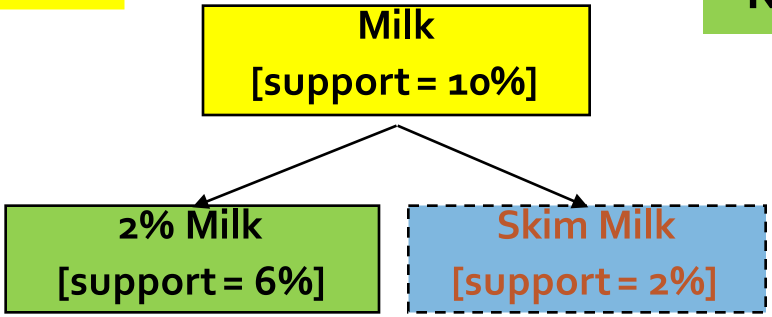
Mining Multiple-Level Frequent Patterns

- Items often form hierarchies
  - Ex.: Dairyland 2% milk; Wonder wheat bread
- How to set min-support thresholds?
  - Uniform min-support across multiple levels (reasonable?)
  - Level-reduced min-support: Items at the lower level are expected to have lower support

Uniform support

Level 1
min_sup = 5%

Level 2
min_sup = 5%



Reduced support

Level 1
min_sup = 5%

Level 2
min_sup = 1%

Redundancy Filtering at Mining Multi-Level Associations

- Multi-level association mining may generate many redundant rules
- Redundancy filtering: Some rules may be redundant due to "ancestor" relationships between items

  (Suppose the 2% milk sold is about ¼ of milk sold in gallons)

  – milk $\Rightarrow$ wheat bread [support = 8%, confidence = 70%] (1)

  – 2% milk $\Rightarrow$ wheat bread [support = 2%, confidence = 72%] (2)

- A rule is *redundant* if its support is close to the "expected" value, according to its "ancestor" rule, and it has a similar confidence as its "ancestor"
  – Rule (1) is an ancestor of rule (2), which one to prune?

Customized Min-Supports for Different Kinds of Items

- We have used the same min-support threshold for all the items or item sets to be mined in each association mining
- In reality, some items (e.g., diamond, watch, …) are valuable but less frequent
- It is necessary to have customized min-support settings for different kinds of items
- One Method: Use group-based "individualized" min-support
  - E.g., {diamond, watch}: 0.05%; {bread, milk}: 5%; …

Mining Multi-Dimensional Associations

- Single-dimensional rules (e.g., items are all in "product" dimension)
  - buys(X, "milk") ⇒ buys(X, "bread")
- Multi-dimensional rules (i.e., items in ≥ 2 dimensions or predicates)
  - Inter-dimension association rules (*no repeated predicates*)
    - age(X, "18-25") ∧ occupation(X, "student") ⇒ buys(X, "coke")
  - Hybrid-dimension association rules (*repeated predicates*)
    - age(X, "18-25") ∧ buys(X, "popcorn") ⇒ buys(X, "coke")
- Attributes can be categorical or numerical
  - Categorical Attributes (e.g., *profession*, *product*: no ordering among values): Data cube for inter-dimension association
  - Quantitative Attributes: Numeric, implicit ordering among values—discretization, clustering, and gradient approaches

Mining Quantitative Associations

- Mining quantitative associations
  - Ex.: Gender = female ⇒ Wage: mean=$7/hr (overall mean = $9)
  - LHS: a subset of the population
  - RHS: an ***extraordinary*** behavior of this subset
- Rule condition can be categorical or numerical
  - Ex.: (Gender = female) ^ (South = yes) ⇒ mean wage = $6.3/hr
  - Ex.: Education in [14-18] (yrs) ⇒ mean wage = $11.64/hr
- Data cube technology?

Rare Patterns vs. Negative Patterns

- Rare patterns
  - Very low support but interesting (e.g., buying Rolex watches)
  - How to mine them? Setting individualized, group-based min-support thresholds for different groups of items
- Negative patterns
  - Negatively correlated: Unlikely to happen together
  - Ex.: Since it is unlikely that the same customer buys both a Ford Expedition (an SUV car) and a Ford Fusion (a hybrid car), buying a Ford Expedition and buying a Ford Fusion are likely negatively correlated patterns
  - How to define negative patterns?

Defining Negative Correlated Patterns

- A support-based definition
  - If itemsets A and B are both frequent but rarely occur together, i.e., $sup(A \cup B) \ll sup(A) \times sup(B)$
  - Then A and B are negatively correlated

Does this remind you the definition of *lift*?

- Is this a good definition for large transaction datasets?
- Ex.: Suppose a store sold two needle packages A and B 100 times each, but only one transaction contained both A and B
  - When there are in total 200 transactions, we have
    - $s(A \cup B) = 0.005$, $s(A) \times s(B) = 0.25$, $s(A \cup B) \ll s(A) \times s(B)$
  - But when there are 10^5 transactions, we have
    - $s(A \cup B) = 1/10^5$, $s(A) \times s(B) = 1/10^3 \times 1/10^3$, $s(A \cup B) > s(A) \times s(B)$
  - What is the problem? — Null transactions: The support-based definition is not null-invariant!

Defining Negative Correlation: Need Null-Invariance in Definition

- A good definition on negative correlation should take care of the null-invariance problem
  - Whether two itemsets A and B are negatively correlated should not be influenced by the number of null-transactions
- A Kulczynski measure-based definition
  - If itemsets A and B are frequent but $(P(A|B) + P(B|A))/2 < \epsilon$, where ϵ is a negative pattern threshold, then A and B are negatively correlated
- For the same needle package problem:
  - No matter there are in total 200 or 10^5 transactions
  - If $\epsilon = 0.01$, we have $(P(A|B) + P(B|A))/2 = (0.01 + 0.01)/2 < \epsilon$

Advanced Frequent Pattern Mining

- Mining Diverse Patterns
- **Constraint-Based Frequent Pattern Mining**
- Sequential Pattern Mining
- Graph Pattern Mining

Why Constraint-Based Mining?

- Finding all the patterns in a dataset autonomously? — unrealistic!
  - Too many patterns but not necessarily user-interested!
- Pattern mining should be an interactive process
  - User directs what to be mined using a data mining query language (or a graphical user interface)
- Constraint-based mining
  - User flexibility: provides constraints on what to be mined
  - Optimization: explores such constraints for efficient mining
    - Constraint-based mining: Constraint-pushing, similar to push selection first in DB query processing

Meta-Rule Guided Mining

- A meta-rule can contain partially instantiated predicates & constants
  - $P_1(X, Y) \wedge P_2(X, W) \Rightarrow$ buys(X, "iPad")
- The resulting mined rule can be
  - age(X, "15-25") ^ profession(X, "student") ⇒ buys(X, "iPad")
- In general, (meta) rules can be in the form of
  - $P_1 \wedge P_2 \wedge \ldots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \ldots \wedge Q_r$
- Method to find meta-rules
  - Find frequent (l + r) predicates (based on *min-support*)
  - Push constants deeply when possible into the mining process
  - Also, push min_sup, min_conf, and other measures as early as possible (measures acting as constraints)

Different Kinds of Constraints Lead to Different Pruning Strategies

- Constraints can be categorized as
  - **Pattern space** pruning constraints vs. **data space** pruning constraints
- Pattern space pruning constraints
- Data space pruning constraints

Pattern Space Pruning with **Pattern Anti-Monotonicity**

- Constraint c is anti-monotone
  - If an itemset S violates constraint c, so does any of its superset
  - That is, mining on itemset S can be terminated
- Ex. 1: c1: sum(S.price) ≤ v is anti-monotone
- Ex. 2: c2: range(S.profit) ≤ 15 is anti-monotone
  - Itemset ab violates c2 (range(ab) = 40)
  - So does every superset of ab
- Ex. 3. c3: sum(S.Price) ≥ v is not anti-monotone
- Ex. 4. Is c4: support(S) ≥ σ anti-monotone?
  - Yes! Apriori pruning is essentially pruning with an anti-monotonic constraint!

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

min_sup = 2

price(item) > 0

Item	Profit
a	40
b	0
c	-20
d	-15
e	-30
f	-10
g	20
h	5

Pattern Monotonicity and Its Roles

- A constraint c is monotone: if an itemset S satisfies the constraint c, so does any of its superset
  - That is, we do not need to check c in subsequent mining
- Ex. 1: c1: sum(S.Price) ≥ v is monotone
- Ex. 2: c2: min(S.Price) ≤ v is monotone
- Ex. 3: c3: range(S.profit) ≥ 15 is monotone
  - Itemset ab satisfies c3
  - So does every superset of ab

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

min_sup = 2

price(item) > 0

Item	Profit
a	40
b	0
c	-20
d	-15
e	-30
f	-10
g	20
h	5

Data Space Pruning with **Data Anti-Monotonicity**

- A constraint c is *data anti-monotone*: In the mining process, if a data entry (transaction) *t* cannot satisfy constraint *c*, *t* cannot satisfy any pattern *p* under *c*
  - Data space pruning*:* Data entry *t* can be pruned
- Ex. 1: c_1: *sum(S.Profit)* ≥ *v* is data anti-monotone
  - Let constraint c_1 be: sum{S.Profit} ≥ 25
    - T_{30}: {b, c, d, f, g} can be removed since none of their combinations can make an S whose sum of the profit is ≥ 25
- Ex. 2: c_2: *min(S.Price)* ≤ *v* is data anti-monotone
    - Consider *v* = 5 but every item in transaction T_{50} has a price higher than 10

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

min_sup = 2

price(item) > 10

Item	Profit
a	40
b	0
c	-20
d	-15
e	-30
f	-10
g	20
h	5

Different Kinds of Constraints Lead to Different Pruning Strategies

- Constraints can be categorized as
  - Pattern space pruning constraints vs. data space pruning constraints
- Pattern space pruning constraints
  - **Anti-monotonic:** If constraint c is violated, its further mining can be terminated (=no superset)
  - **Monotonic:** If c is satisfied, no need to check c again (=all supersets)
  - Succinct: If c can be enforced by directly manipulating the data
  - Convertible: c can be converted to monotonic or anti-monotonic if items can be properly ordered in processing
- Data space pruning constraints
  - **Data anti-monotonic:** If a transaction t does not satisfy c, then t can be pruned to reduce data processing effort (=no that transaction)
  - Data succinct: Data space can be pruned at the initial pattern mining process

Pattern Mining Methods

Pattern	Closed Pattern (Concepts)	Idea 1: Pattern candidate generation and pruning	Idea 2: Vertical format to accelerate mining	Idea 3: Pattern growth
Frequent pattern (itemset)	?	?	?	?
Sequential pattern	?	?	?	?
Graph pattern	?	?	x	?

Pattern Mining Methods

Pattern	Closed Pattern (Concepts)	Idea 1: Pattern candidate generation and pruning	Idea 2: Vertical format to accelerate mining	Idea 3: Pattern growth
Frequent pattern (itemset)	Closed frequent itemset	Apriori	ECLAT	FP-Growth
Sequential pattern	Closed seq. pattern	?	?	?
Graph pattern	Closed graph pattern	?	x	?

Advanced Frequent Pattern Mining

- Mining Diverse Patterns
- Constraint-Based Frequent Pattern Mining
- **Sequential Pattern Mining**
- Graph Pattern Mining

Sequential Patterns: Applications

- Sequential pattern mining has broad applications
  - Customer shopping sequences
    - Purchase a laptop first, then a digital camera, and then a smartphone, within 6 months
  - Medical treatments, natural disasters (e.g., earthquakes), science & engineering processes, stocks and markets, ...
  - Weblog click streams, calling patterns, ...
  - Software engineering: Program execution sequences, ...
  - Biological sequences: DNA, protein, ...

Sequential Pattern and Sequential Pattern Mining

- Sequential pattern mining: Given a set of sequences, find the complete set of frequent subsequences (i.e., satisfying the min_sup threshold)

A *sequence database*

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

A *sequence*: < (ef) (ab) (df) c b >

- An element may contain a set of items (also called events)
- Items within an element are unordered and we list them alphabetically

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

- Given support threshold min_sup = 2, <(ab)c> is a sequential pattern

http://hanj.cs.illinois.edu/pdf/span01.pdf

Sequence vs Element/Itemset/Event vs Item/Instance

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of all **items**. An **itemset** is a subset of items. A **sequence** is an ordered list of itemsets. A sequence s is denoted by $\langle s_1 s_2 \cdots s_l \rangle$, where s_j is an itemset, i.e., $s_j \subseteq I$ for $1 \leq j \leq l$. s_j is also called an **element** of the sequence, and denoted as $(x_1 x_2 \cdots x_m)$, where x_k is an item, i.e., $x_k \in I$ for $1 \leq k \leq m$. For brevity, the brackets are omitted if an element has only one item. That is, element (x) is written as x. An item can occur at most once in an element of a sequence, but can occur multiple times in different elements of a sequence. The

Sequential Pattern Mining Algorithms

- Algorithm requirement: Efficient, scalable, finding complete set, incorporating various kinds of user-specific constraints
- The Apriori property still holds: If a subsequence s_1 is infrequent, none of s_1's super-sequences can be frequent
- Representative algorithms
  - Apriori-based Generalized Sequential Patterns: GSP (Srikant & Agrawal @ EDBT'96)
  - Vertical format-based mining: SPADE (Zaki@Machine Leanining'00)
  - Pattern-growth methods: PrefixSpan (Pei, et al. @TKDE'04)
- Mining **closed** sequential patterns: CloSpan (Yan, et al. @SDM'03)
- Constraint-based sequential pattern mining

GSP: Apriori-Based Sequential Pattern Mining

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

- Initial candidates: All singleton sequences
  - <a>, <b>, <c>, <d>, <e>, <f>, <g>, <h>
- Scan DB once, count support for each candidate
- Generate length-2 candidate sequences

min_sup = 2

Cand.	sup
<a>	3
<b>	5
<c>	4
<d>	3
<e>	3
<f>	2
<g>	1
<h>	1

	<a>	<b>	<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
<b>	<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>

Length-2 candidates:
36 + 15= 51
Without Apriori pruning:
8*8+8*7/2=92 candidates

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

GSP (Generalized Sequential Patterns): Srikant & Agrawal @ EDBT'96)

GSP Mining and Pruning

- Repeat (for each level (i.e., length-k))
  - Scan DB to find length-k frequent sequences
  - Generate length-(k+1) candidate sequences from length-k frequent sequences using Apriori
  - set k = k+1
- Until no frequent sequence or no candidate can be found

Sequential Pattern Mining in Vertical Data Format: The SPADE Algorithm

- A sequence database is mapped to: <SID, EID>
- Grow the subsequences (patterns) one item at a time by Apriori candidate generation

SID	Sequence
1	<a(abc)(ac)d(cf)>
2	<(ad)c(bc)(ae)>
3	<(ef)(ab)(df)cb>
4	<eg(af)cbc>

min_sup = 2

Ref: SPADE (Sequential PAttern Discovery using Equivalent Class) [M. Zaki 2001]

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	

https://pdfs.semanticscholar.org/39a0/80c17dec400a6f04af5fe5746dab3a5eb0dc.pdf

PrefixSpan: A Pattern-Growth Approach

- Prefix and suffix
  - Given <a(abc)(ac)d(cf)>
  - Prefixes: <a>, <aa>, <a(ab)>, <a(abc)>, …
  - Prefixes-based projection
- PrefixSpan Mining: Prefix Projections
  - Step 1: Find length-1 sequential patterns
    - <a>, <b>, <c>, <d>, <e>, <f>
  - Step 2: Divide search space and mine each projected DB
    - <a>-projected DB,
    - <b>-projected DB,
    - …
    - <f>-projected DB, …

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

Prefix	Suffix (Projection)
<a>	<(abc)(ac)d(cf)>
<aa>	<(_bc)(ac)d(cf)>
<ab>	<(_c)(ac)d(cf)>

PrefixSpan (Prefix-projected Sequential pattern mining) Pei, et al. @TKDE'04

http://hanj.cs.illinois.edu/pdf/span01.pdf

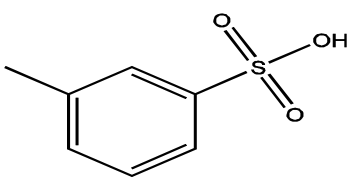
Advanced Frequent Pattern Mining

- Mining Diverse Patterns
- Constraint-Based Frequent Pattern Mining
- Sequential Pattern Mining
- **Graph Pattern Mining**

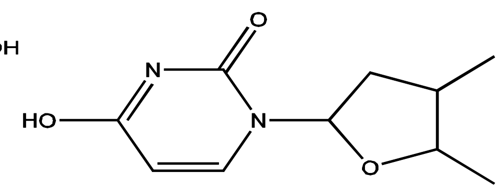
Frequent (Sub)Graph Patterns

- Given a labeled graph dataset $D = \{G_1, G_2, \ldots, G_n\}$, the supporting graph set of a subgraph g is $D_g = \{G_i \mid g \subseteq G_i, G_i \in D\}$.
  - support(g) = $|D_g| / |D|$
- A (sub)graph g is ***frequent*** if *support*(g) ≥ *min_sup* Ex.: Chemical structures
- Alternative:
  - Mining frequent subgraph patterns from a single large graph or network

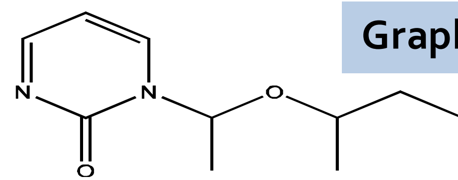
Graph Dataset



(A)



(B)

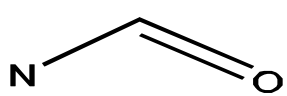


(C)

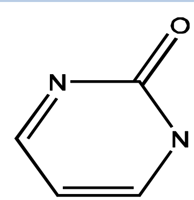
min_sup = 2

Frequent Graph Patterns

(1)



(2)



Graph Pattern Mining: Applications

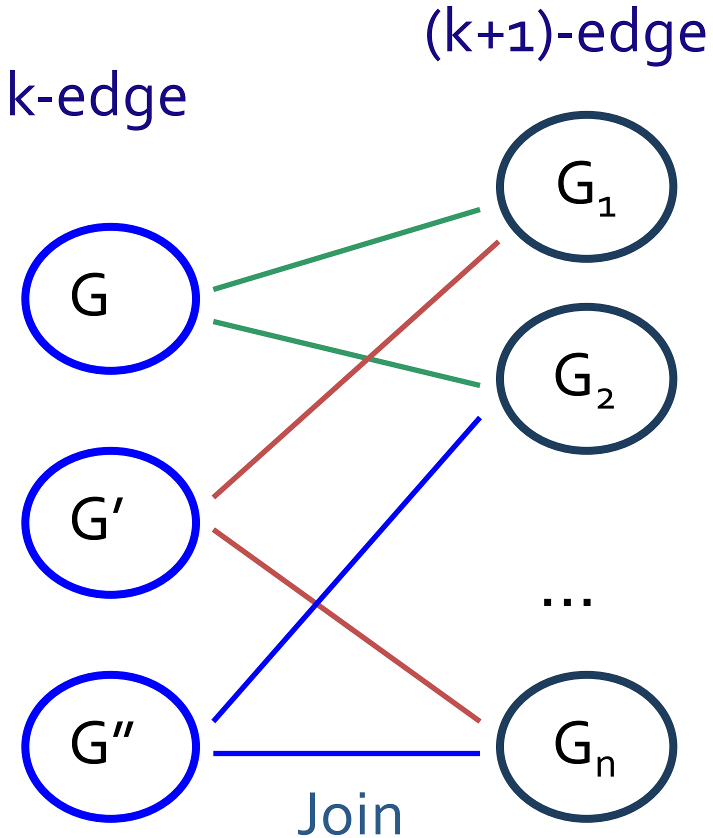
- Bioinformatics
  - Gene networks, protein interactions, metabolic pathways
- Chem-informatics: Mining chemical compound structures
- Social networks, web communities, tweets, …
- Cell phone networks, computer networks, …
- Web graphs, XML structures, semantic Web, information networks
- Software engineering: program execution flow analysis
- Building blocks for graph classification, clustering, compression, comparison, and correlation analysis
- Graph indexing and graph similarity search

Graph Pattern Mining Algorithms: Different Methodologies

- Generation of candidate subgraphs
  - Apriori vs. pattern growth (e.g., FSG vs. gSpan)
- Search order
  - Breadth vs. depth
- Elimination of duplicate subgraphs
  - Passive vs. active (e.g., gSpan (Yan&Han'02))
- Order of pattern discovery
  - Path → tree → graph (e.g., GASTON (Nijssen&Kok'04))

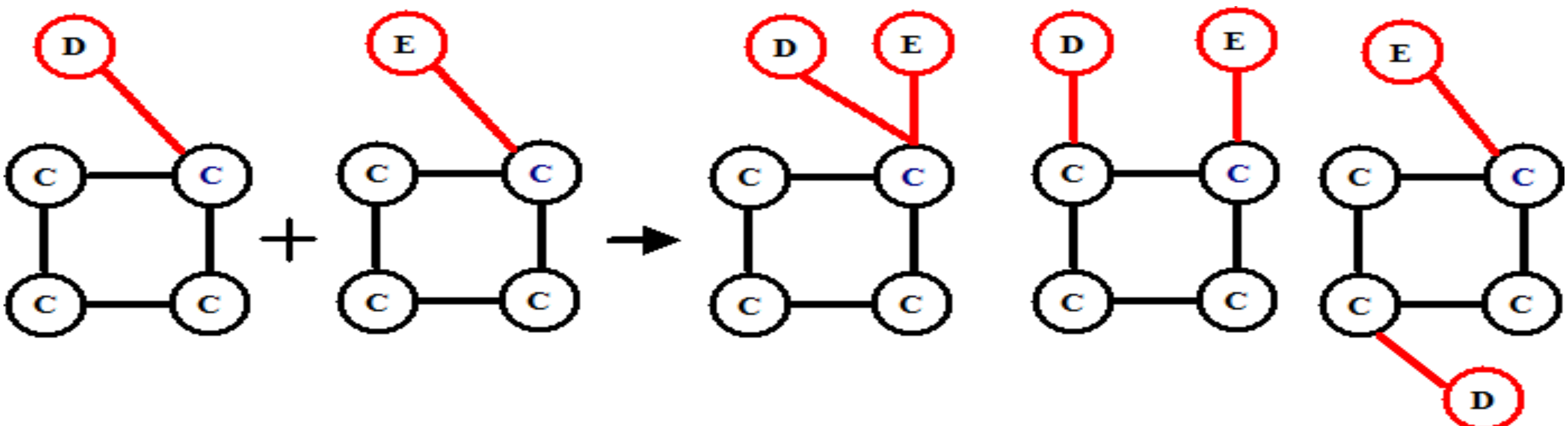
Apriori-Based Approach

- The Apriori property (anti-monotonicity): A size-*k* subgraph is frequent if and only if all of its subgraphs are frequent
- A candidate size-(*k*+1) edge/vertex subgraph is generated if its corresponding two *k*-edge/vertex subgraphs are frequent
- Iterative mining process:
  - Candidate-generation → candidate pruning → support counting → candidate elimination



Candidate Generation: Vertex Growing vs. Edge Growing

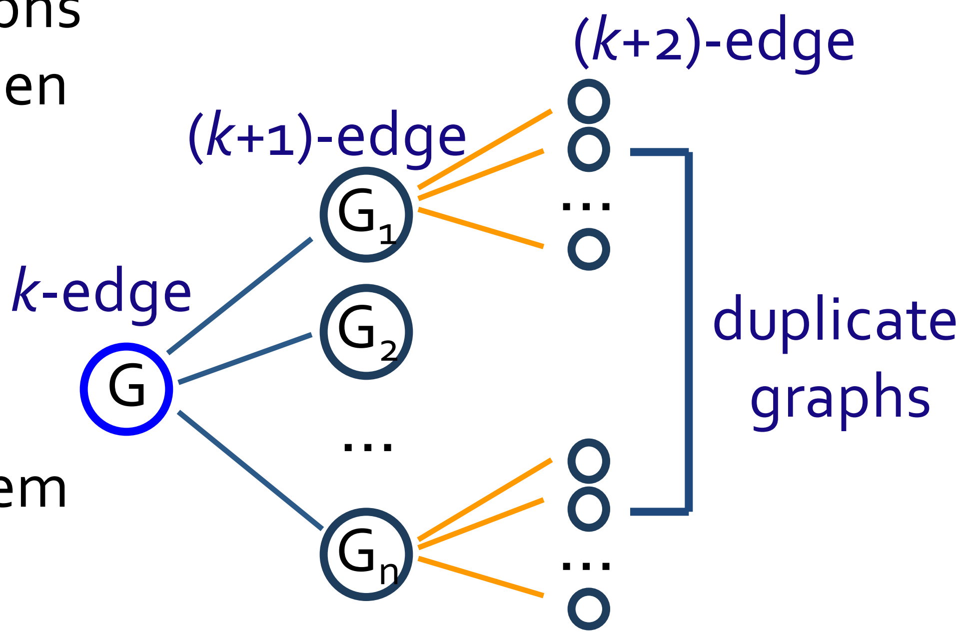
- Methodology: breadth-search, Apriori joining two size-k graphs
  - Many possibilities at generating size-(k+1) candidate graphs



- Generating new graphs with one more vertex
  - AGM (Inokuchi, et al., PKDD'00)
- Generating new graphs with one more edge
  - FSG (Kuramochi and Karypis, ICDM'01)
- Performance shows via edge growing is more efficient

Pattern-Growth Approach

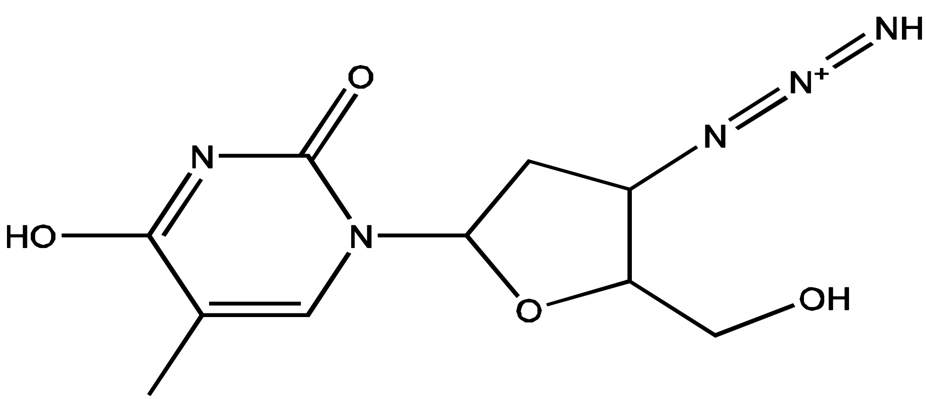
- Depth-first growth of subgraphs from *k*-edge to (*k*+1)-edge, then (*k*+2)-edge subgraphs
- Major challenge
  - Generating many duplicate subgraphs
- Major idea to solve the problem
  - Define an order to generate subgraphs
  - DFS spanning tree: Flatten a graph into a sequence using depth-first search
  - gSpan (Yan & Han: ICDM'02)



http://cs.ucsb.edu/~xyan/papers/gSpan-short.pdf

Why Mining Closed Graph Patterns?

- Challenge: An **n**-edge frequent graph may have 2^n subgraphs
- Motivation: Explore *closed frequent subgraphs* to handle graph pattern explosion problem
- A frequent graph G is *closed* if there exists no supergraph of G that carries the same support as G

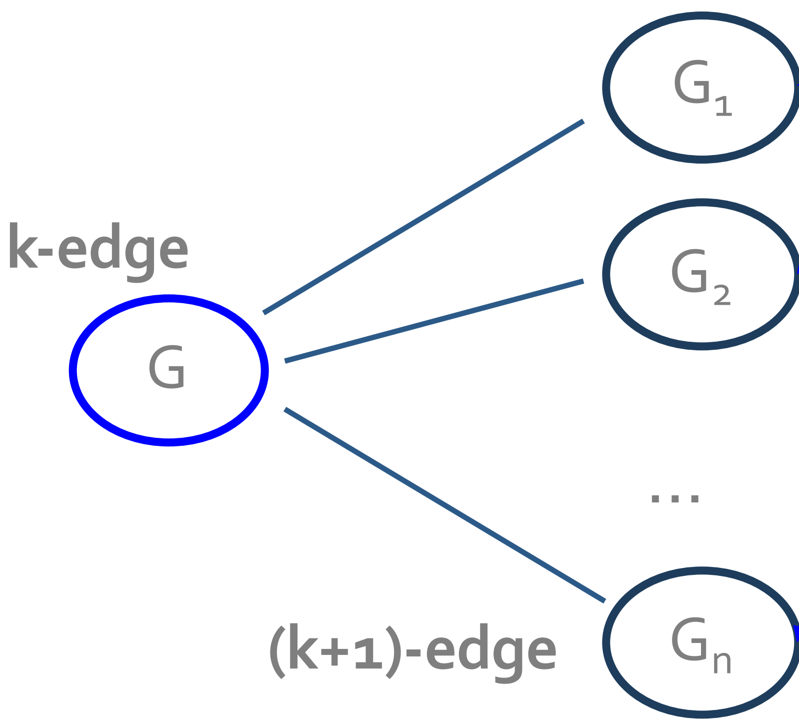


If this subgraph is *closed* in the graph dataset, it implies that none of its frequent super-graphs carries the same support

- *Lossless compression:* Does not contain non-closed graphs, but still ensures that the mining result is complete
- Algorithm CloseGraph: Mines closed graph patterns directly

CloseGraph: Directly Mining Closed Graph Patterns

- CloseGraph: Mining closed graph patterns by extending gSpan
- Suppose G and G_1 are frequent, and G is a subgraph of G_1
- If **in any part of the graph in the dataset where G occurs, G_1 also occurs**, then we need not grow G (except some special, subtle cases), since none of G's children will be closed except those of G_1



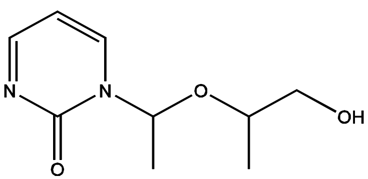
At what condition can we stop searching their children, i.e., early termination?

https://pdfs.semanticscholar.org/a1c1/5e63690c774b725fc91dcc77a629d01c3733.pdf

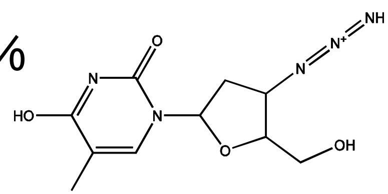
Experiment and Performance Comparison

- The AIDS antiviral screen compound dataset from NCI/NIH
- The dataset contains 43,905 chemical compounds
- Discovered Patterns: The smaller minimum support, the bigger and more interesting subgraph patterns discovered

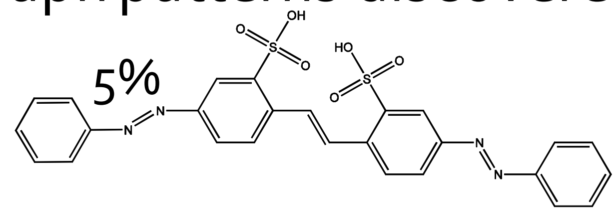
20%

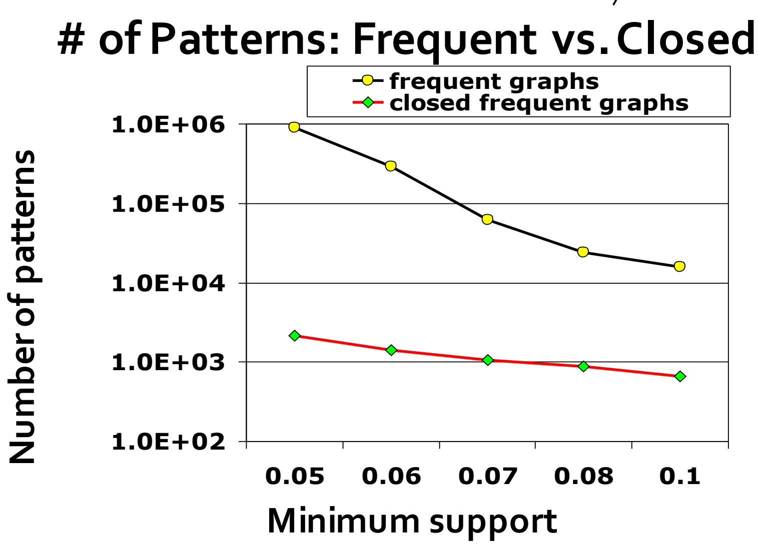


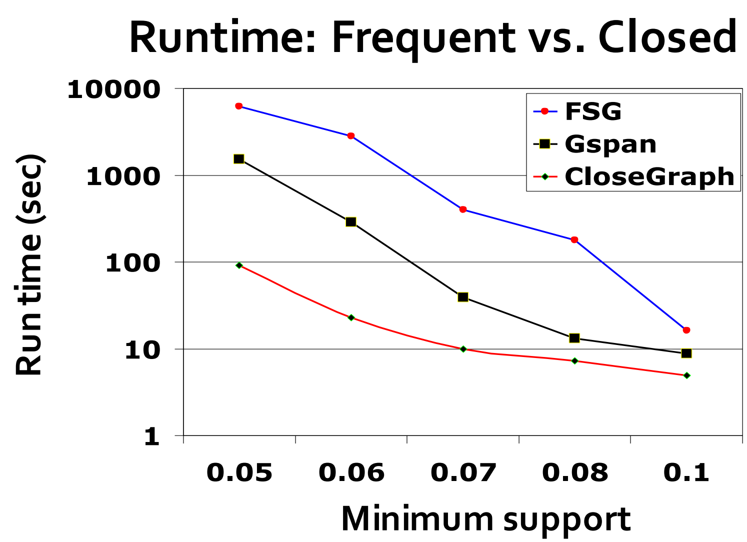
10%



5%







References: Mining Diverse Patterns

- 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
- K. Wang, Y. He, J. Han, “Pushing Support Constraints Into Association Rules Mining”, IEEE Trans. Knowledge and Data Eng. 15(3): 642-658, 2003
- 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
- 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
- F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng, “Mining Colossal Frequent Patterns by Core Pattern Fusion”, ICDE'07

References: Constraint-Based Frequent Pattern Mining

- R. Srikant, Q. Vu, and R. Agrawal, “Mining association rules with item constraints”, KDD'97
- R. Ng, L.V.S. Lakshmanan, J. Han & A. Pang, “Exploratory mining and pruning optimizations of constrained association rules”, SIGMOD’98
- G. Grahne, L. Lakshmanan, and X. Wang, “Efficient mining of constrained correlated sets”, ICDE'00
- J. Pei, J. Han, and L. V. S. Lakshmanan, “Mining Frequent Itemsets with Convertible Constraints”, ICDE'01
- J. Pei, J. Han, and W. Wang, “Mining Sequential Patterns with Constraints in Large Databases”, CIKM'02
- F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, “ExAnte: Anticipated Data Reduction in Constrained Pattern Mining”, PKDD'03
- F. Zhu, X. Yan, J. Han, and P. S. Yu, “gPrune: A Constraint Pushing Framework for Graph Pattern Mining”, PAKDD'07

References: Sequential Pattern Mining

- R. Srikant and R. Agrawal, “Mining sequential patterns: Generalizations and performance improvements”, EDBT’96
- M. Zaki, “SPADE: An Efficient Algorithm for Mining Frequent Sequences”, Machine Learning, 2001
- J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach", IEEE TKDE, 16(10), 2004
- X. Yan, J. Han, and R. Afshar, “CloSpan: Mining Closed Sequential Patterns in Large Datasets”, SDM'03
- J. Pei, J. Han, and W. Wang, "Constraint-based sequential pattern mining: the pattern-growth methods", J. Int. Inf. Sys., 28(2), 2007
- M. N. Garofalakis, R. Rastogi, K. Shim: Mining Sequential Patterns with Regular Expression Constraints. IEEE Trans. Knowl. Data Eng. 14(3), 2002
- H. Mannila, H. Toivonen, and A. I. Verkamo, “Discovery of frequent episodes in event sequences”, Data Mining and Knowledge Discovery, 1997

References: Graph Pattern Mining

- C. Borgelt and M. R. Berthold, Mining molecular fragments: Finding relevant substructures of molecules, ICDM'02
- J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism, ICDM'03
- A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data, PKDD'00
- M. Kuramochi and G. Karypis. Frequent subgraph discovery, ICDM'01
- S. Nijssen and J. Kok. A Quickstart in Frequent Structure Mining can Make a Difference. KDD'04
- N. Vanetik, E. Gudes, and S. E. Shimony. Computing frequent graph patterns from semistructured data, ICDM'02
- X. Yan and J. Han, gSpan: Graph-Based Substructure Pattern Mining, ICDM'02
- X. Yan and J. Han, CloseGraph: Mining Closed Frequent Graph Patterns, KDD'03
- X. Yan, P. S. Yu, J. Han, Graph Indexing: A Frequent Structure-based Approach, SIGMOD'04
- X. Yan, P. S. Yu, and J. Han, Substructure Similarity Search in Graph Databases, SIGMOD'05