Project Schedule

Date	L#	Topic	Goals
01-16 (T)	1	Introduction	Understand what is data science research
02-06 (T)	7	Proposal: Teaming and proposal	Submit your proposal paper: • What is your project topic/research problem? • How will you find your dataset? • What may be your proposed method?
03-08 (R)	15	Milestone	Submit your milestone paper: • Your topic, dataset, and method • Milestone progress: Some preliminary results • Challenges and proposed solutions • Plan for the next two months
04-26 (R)	27	Oral 1 (up to 20% additional credits)	Every team gives an oral presentation. Classmates, instructor, and invited faculty will evaluate your presentation.
05-01 (T)	28	Oral 2	
05-03 (R)		Paper due	Project final paper due: You have to submit your code package, data, and term paper at 11:59PM this date.

Project Evaluation

- Proposal paper (10 points)
- Milestone presentation/paper (15 points)

- **Final term oral presentation (25 points)**
  - **04/26 and 05/01**
  - **Graded by classmates, invited faculty, and instructor**
- **Final term paper (25 points)**
  - **05/03**
  - **Graded by instructor**
- **Code package and data (25 points)**
  - **05/03**
  - **Graded by instructor and TA**

Dr. Taeho Jung

Data Security and Privacy Lab (DSP-Lab)
CSE 20110 Discrete Mathematics (Fall 2017)
CSE 40622 Cryptography (Spring 2018)

Grading Code Package and Data

- README.md (20%: 5 points)
- Runnable? (40%: 10 points)
- Reproducible? (40%: 10 points)
- Jupyter Notebook is encouraged as supplementary materials. (+2 points)

- Example:
  - REAME.md and makefile: https://github.com/shangjingbo1226/AutoPhrase
  - Jupyter for word2vec: http://nbviewer.jupyter.org/github/danielfrg/word2vec/blob/master/examples/word2vec.ipynb

Grading Final Term Paper

Introduction:	15%	Provide context and motivation. What questions are being addressed? Why are these questions interesting or important?
Related Work:	10%	What other methods have addressed these or similar questions? How do these methods differ from your method?
Solution/Method:	25%	What did you do? What tools and techniques did you use? Was any innovation attempted?
Data and Experiments:	10%	What data did you use? Are your experimental methods reliable? What preprocessing was done the data?
Evaluation and Results:	25%	Did you properly evaluate your experiments? Did you test for statistical significance? Do your conclusions match your results?
Writing Quality:	15%	Clarity of writing (5%), organization (5%), and grammar (5%).

Grading Oral Presentation

Introduction:	15%	Provide context. What questions are being addressed?
Solution/Method:	30%	What did you do? Why did you choose this method? What tools and techniques did you use?
Data and Experiments:	10%	What data did you use? Are your experimental methods reliable?
Evaluation and Results:	30%	What evaluation did you do? Do your conclusions match your results?
Presentation Quality:	15%	Clarity of speaking (5%), organization (5%), and visuals (5%).

Grading Form

- Students (anonymized; skip your own team): 60%
- Invited faculty: 30%
- Instructor: 10%

	Intro (15)	Solution, method (30)	Data and experiments (10)	Evaluation, analysis, results (30)	Presentation quality (15)	Sum (100)
NPM						
ACC						
MLB						
MML						
EBM						
POW						
PBC						
DPH						
AFG						
MPT						

How to Have Grade A?

- Calculated score >= 93
  - **HW1*5% + HW2*5% + HW3*5% + HW4*5%**
  - **Mid exam*20%** (at most 100*20% though honor code bonus)
  - **<u>Final exam*30%</u>** (no honor code bonus)
  - Course project
    - **Proposal*3% + Milestone*4.5%**
    - Presentation (at most 100*7.5%, up to +20% for early-bird: Apr. 26)
      83.333 → 100 (may happen)
      - **<u>Students*4.5%</u>**
      - **<u>Invited faculty*2.25%</u>**
      - **<u>Instructor*0.75%</u>**
    - **<u>Final project paper*7.5%</u>**
      - Usually proportional to the presentation
    - **<u>Code/data package*7.5%</u>**

Letter Grades

- A: 93-100
- A-: 90, 91, 92
- B+: 87, 88, 89
- B: 84, 85, 86
- B-: 81, 82, 83
- C+: 78, 79, 90
- C: 75, 76, 77

Final Exam

- Time: May 8 (Tuesday) 10:30 am – 12:30 pm
- Location: 117 DeBartolo

- Write down your answers/solutions on the blue book.
- Return your exam paper after the exam.
- You can have a double-sided letter-size reference paper.

- You must bring a pen/pencil/writing tool.
- You had better bring a calculator.
- You are not allowed to use laptop/computer/cellphone!
- You are not allowed to bring text book.

Chapter 7. Advanced Frequent Pattern Mining: Diverse Patterns

Meng Jiang

Data Science

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

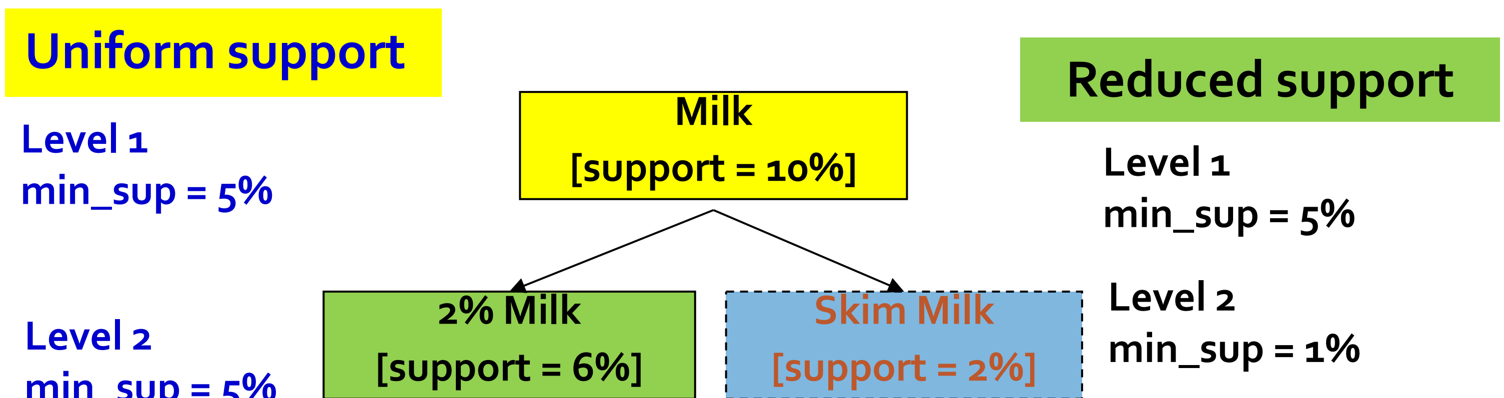
- **Mining Diverse Patterns**
- Constraint-Based Frequent Pattern Mining
- Sequential Pattern Mining
- Graph Pattern Mining

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



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") $\Rightarrow$ buys(X, "bread")
- Multi-dimensional rules (i.e., items in $\geq$ 2 dimensions or predicates)
  - Inter-dimension association rules (*no repeated predicates*)
    - age(X, "18-25") $\wedge$ occupation(X, "student") $\Rightarrow$ buys(X, "coke")
  - Hybrid-dimension association rules (*repeated predicates*)
    - age(X, "18-25") $\wedge$ buys(X, "popcorn") $\Rightarrow$ buys(X, "coke")

Mining Quantitative Associations

- Mining quantitative associations
  - Ex.: Gender = female $\Rightarrow$ 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) $\Rightarrow$ mean wage = $6.3/hr
  - Ex.: Education in [14-18] (yrs) $\Rightarrow$ 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)
- 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 U B) << sup (A) × 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 U B) = 0.005, s(A) × s(B) = 0.25, s(A U B) << s(A) × s(B)
  - But when there are 10^5 transactions, we have
    - s(A U B) = $1/10^5$, s(A) × s(B) = $1/10^3 \times 1/10^3$, s(A U B) > s(A) × 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

Pattern Mining Methods

Pattern	Closed Pattern (Concepts)	Idea 1: Pattern candidate generation and pruning	Idea 2: Pattern growth
Frequent pattern (itemset)	?	?	?
Sequential pattern	?	?	?
Graph pattern	?	?	?

Pattern Mining Methods

Pattern	Closed Pattern (Concepts)	Idea 1: Pattern candidate generation and pruning	Idea 2: Pattern growth
Frequent pattern (itemset)	Closed frequent itemset	Apriori (1994)	FP-Growth (2000)
Sequential pattern	Closed seq. pattern	GSP (1996)	PrefixSpan (2004)
Graph pattern	Closed graph pattern	FSG (2000-2001)	gSpan (2002)

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)
  - 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>

	<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>						

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

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

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

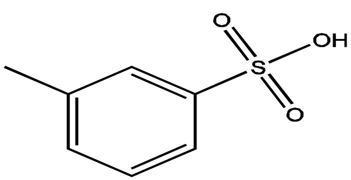
Advanced Frequent Pattern Mining

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

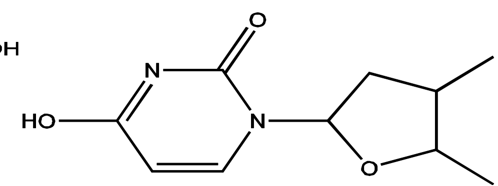
Frequent (Sub)Graph Patterns

- Given a labeled graph dataset $D = \{G_1, G_2, \ldots, G_n\}$, the supporting graph set of a subgraph g is $D_g = \{G_i \mid g \subseteq G_i, G_i \in D\}$.
  - $\text{support}(g) = |D_g| / |D|$
- 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

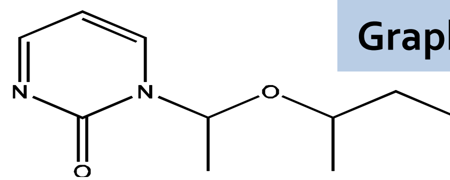
Graph Dataset



(A)



(B)

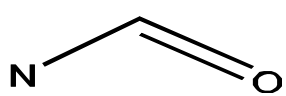


(C)

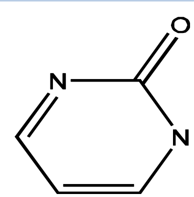
min_sup = 2

Frequent Graph Patterns

(1)



(2)

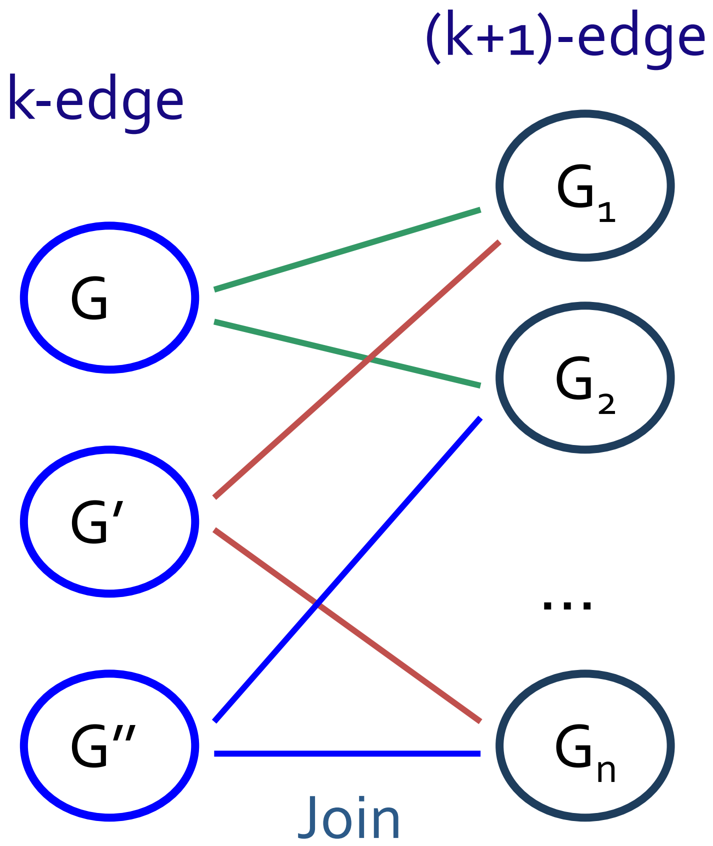


Graph Pattern Mining: Applications

- 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

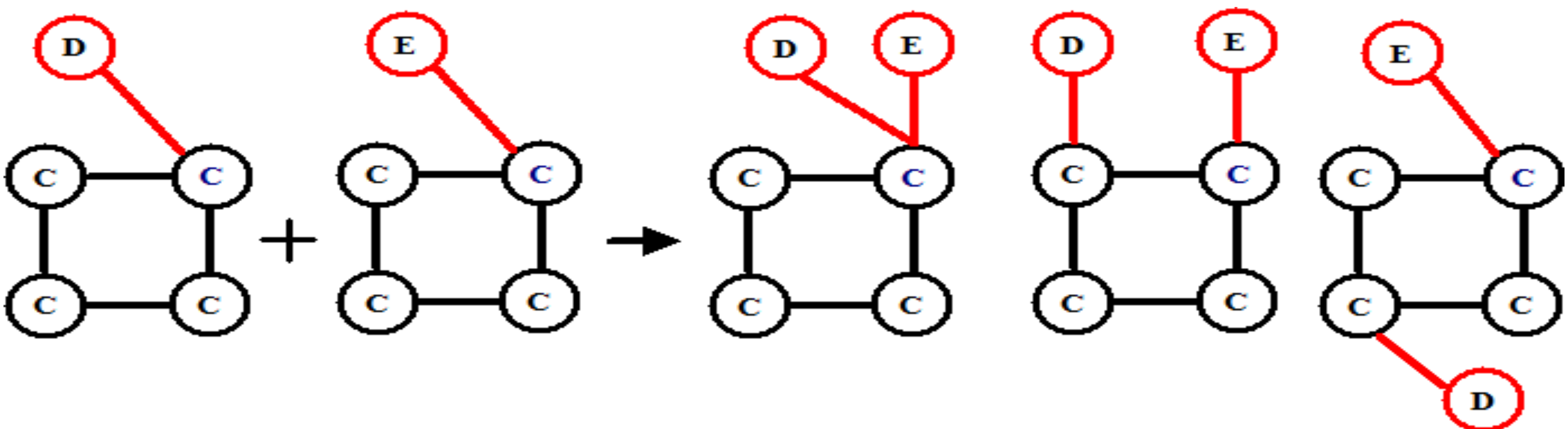
Apriori-Based Approach

- 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

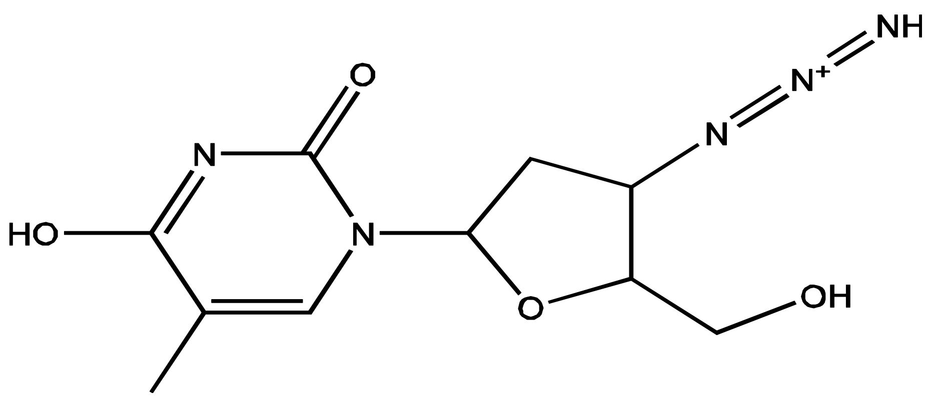
- Methodology: breadth-search, Apriori joining two size-k graphs
  - Many possibilities at generating size-(k+1) candidate graphs



- 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

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



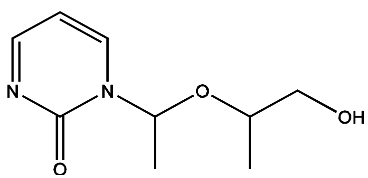
If this subgraph is *closed* in the graph dataset, it implies that none of its frequent super-graphs carries the same support

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

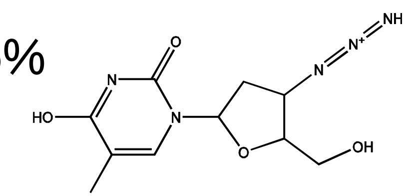
Experiment and Performance Comparison

- 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

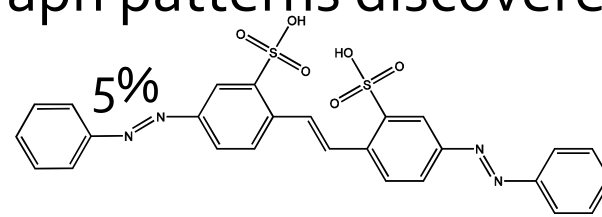
20%



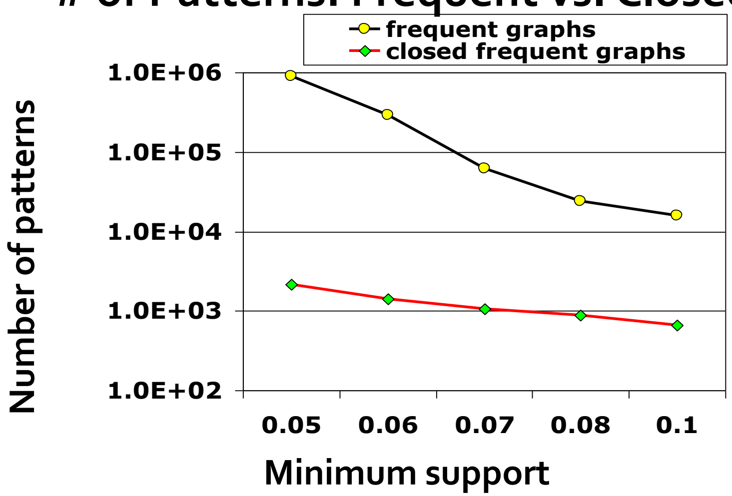
10%



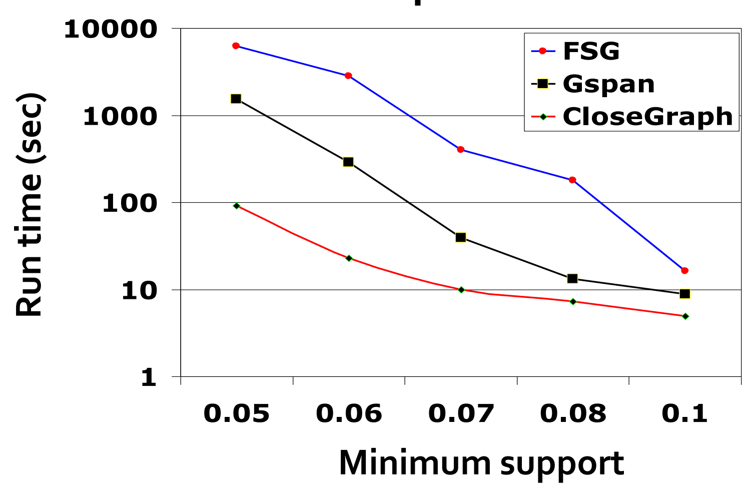
5%



# of Patterns: Frequent vs. Closed



Runtime: Frequent vs. Closed



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