An Introduction to Graph Mining

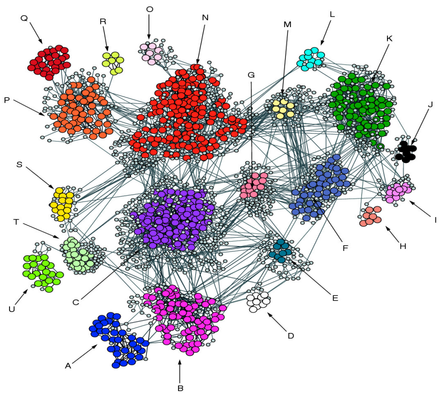
Karsten Borgwardt and Oliver Stegle

Machine Learning and
Computational Biology Research Group,
Max Planck Institute for Biological Cybernetics and
Max Planck Institute for Developmental Biology, Tübingen

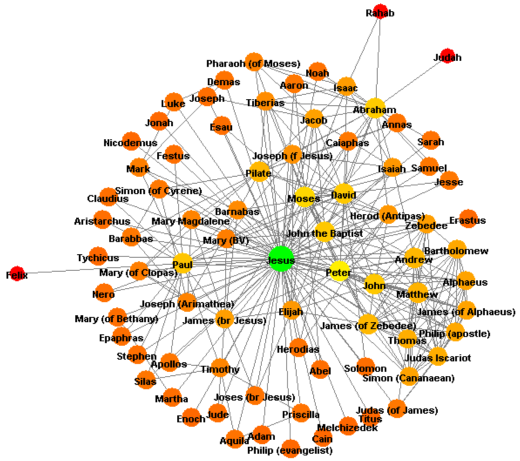
based upon K. Borgwardt and X. Yan: Graph Kernels and Graph Mining. KDD 2008, with permission from Xifeng Yan.

Graphs are everywhere

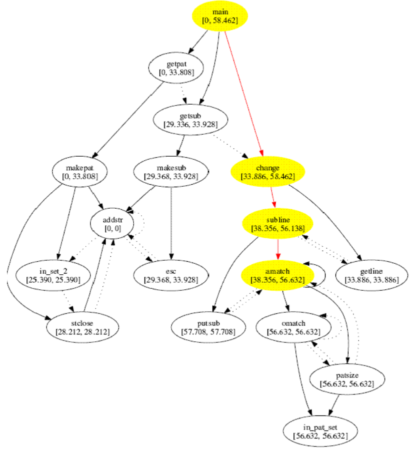
Magwene *et al.* *Genome Biology* 2004 **5**:R100



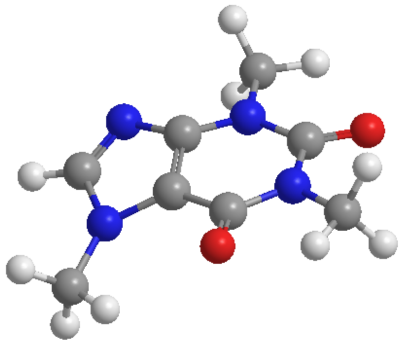
Co-expression Network



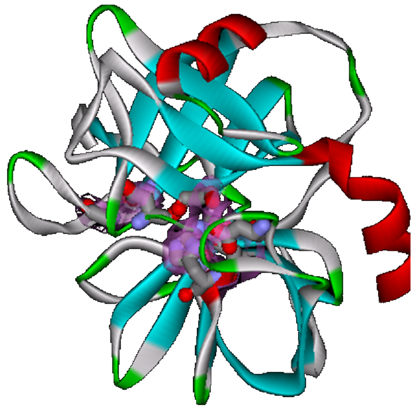
Social Network



Program Flow



Chemical Compound



Protein Structure

Part I: Graph Mining

Graph Pattern Mining

- Frequent graph patterns
- Pattern summarization
- Optimal graph patterns
- Graph patterns with constraints
- Approximate graph patterns

Graph Classification

- Pattern-based approach
- Decision tree
- Decision stumps

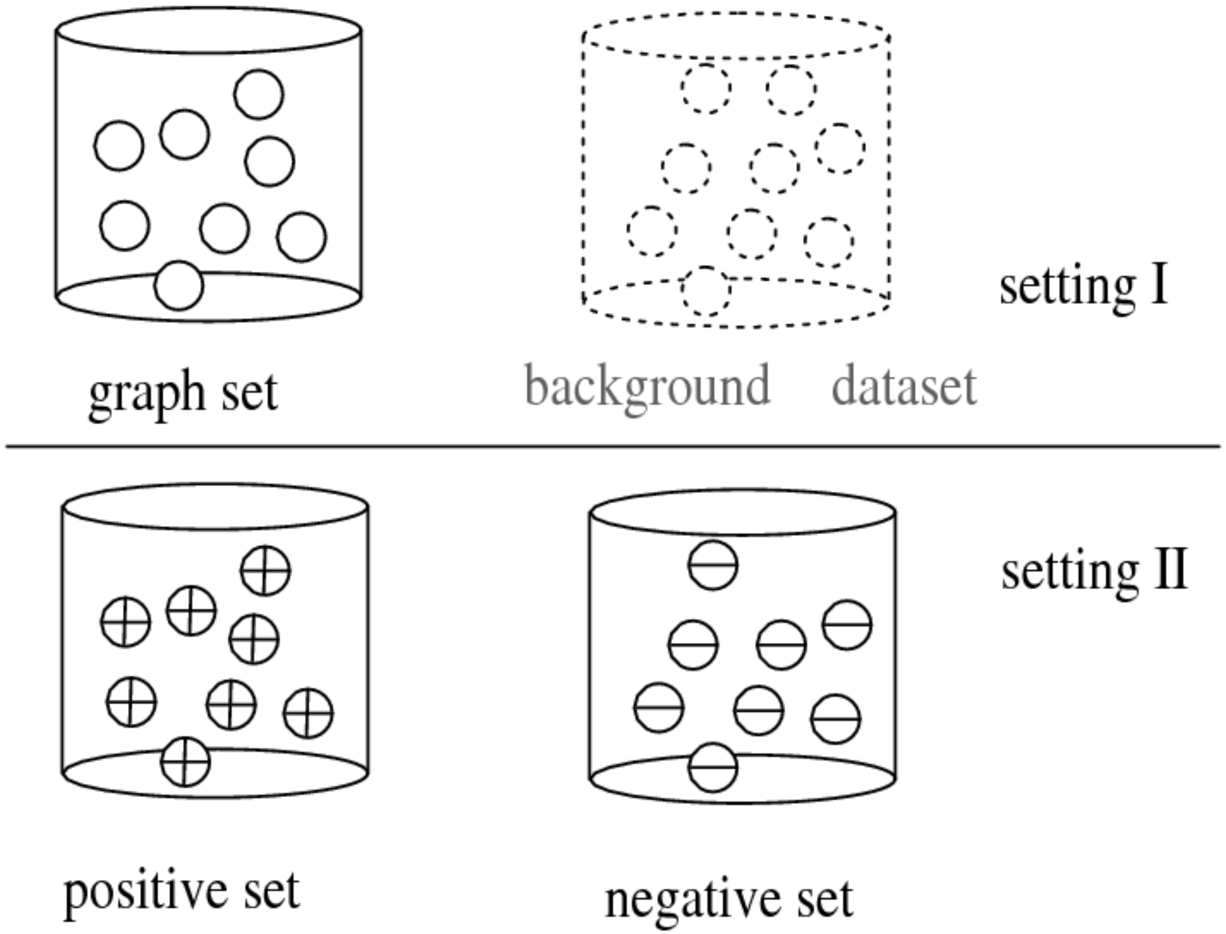
Graph Compression

Other important topics (graph model, laws, graph dynamics, social network analysis, visualization, summarization, graph clustering, link analysis, …)

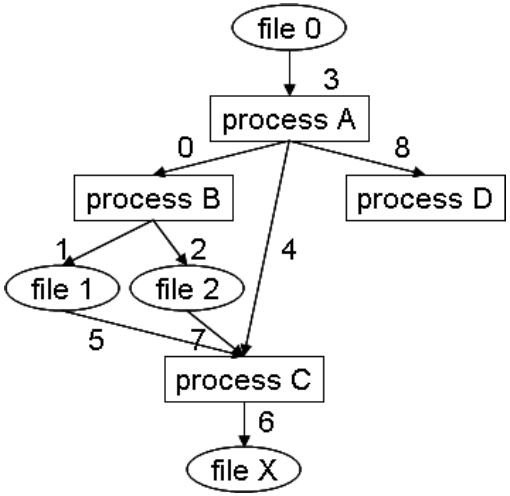
Applications of Graph Patterns

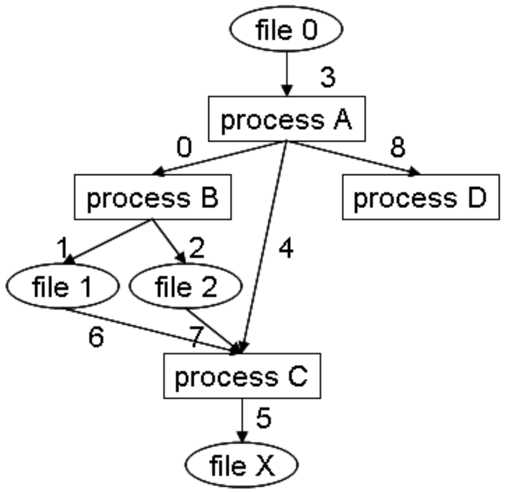
- Mining biochemical structures
- Finding biological conserved subnetworks
- Finding functional modules
- Program control flow analysis
- Intrusion network analysis
- Mining communication networks
- Anomaly detection
- Mining XML structures
- Building blocks for graph classification, clustering, compression, comparison, correlation analysis, and indexing
- …

Graph Pattern Mining



multiple graphs setting





Interestingness measures / Objective functions

- Frequency: frequent graph pattern
- Discriminative: information gain, Fisher score
- Significance: G-test
- …

Frequent Graph Pattern

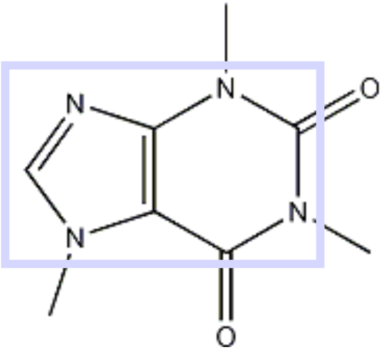
Given a graph dataset D, find subgraph g, s.t.

$$freq(g) \geq \theta$$

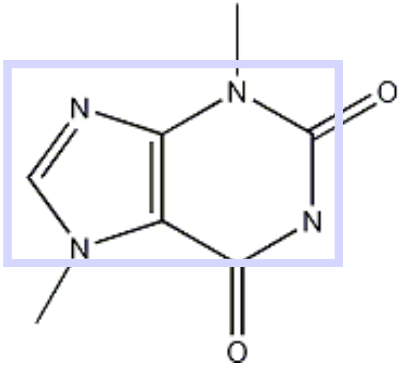
where $freq(g)$ is the percentage of graphs in D that contain g.

Example: Frequent Subgraphs

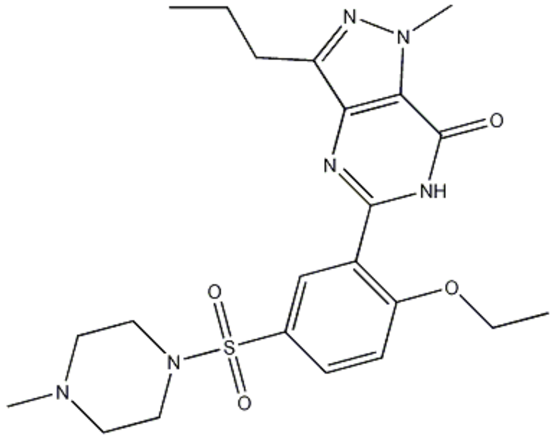
CHEMICAL COMPOUNDS



(a) caffeine



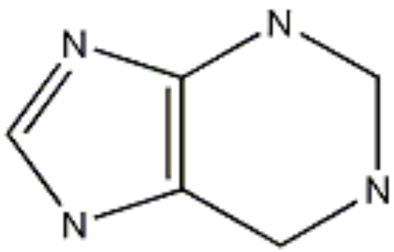
(b) diurobromine



(c) viagra

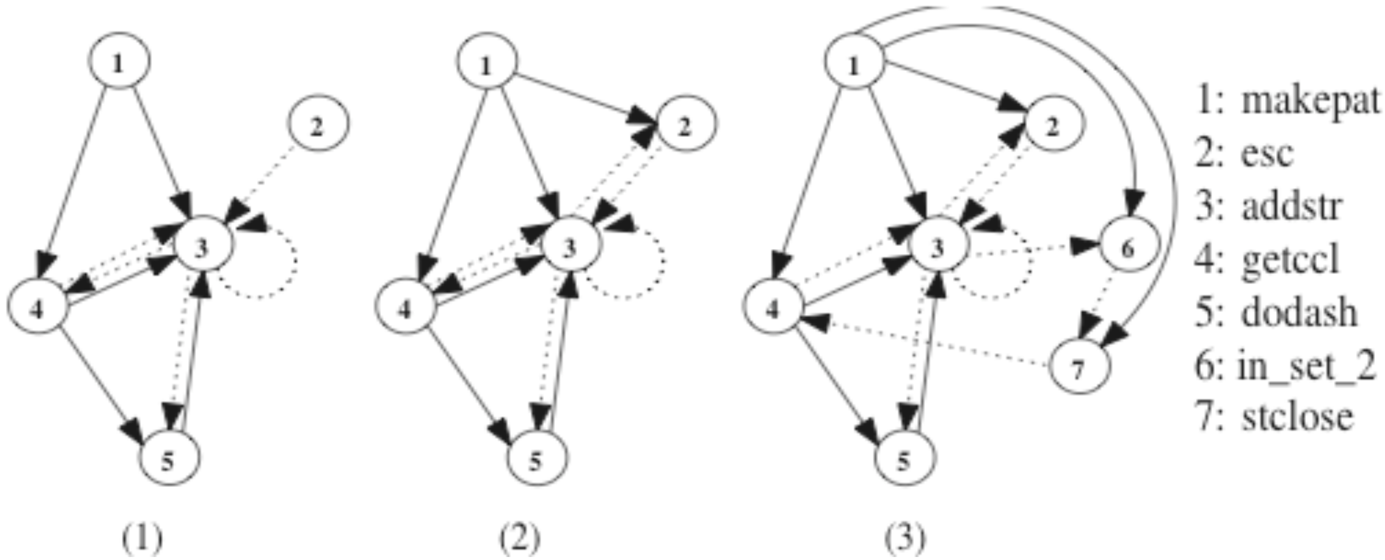
. . .

FREQUENT SUBGRAPH

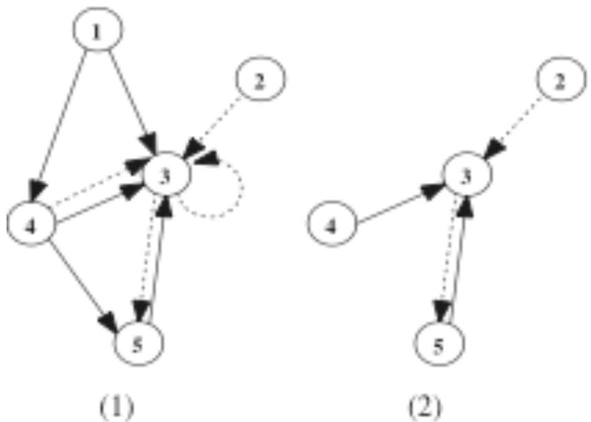


Example (cont.)

PROGRAM CALL GRAPHS



FREQUENT SUBGRAPHS
(MIN SUPPORT IS 2)



Graph Mining Algorithms

Inductive Logic Programming (WARMR, King et al. 2001)

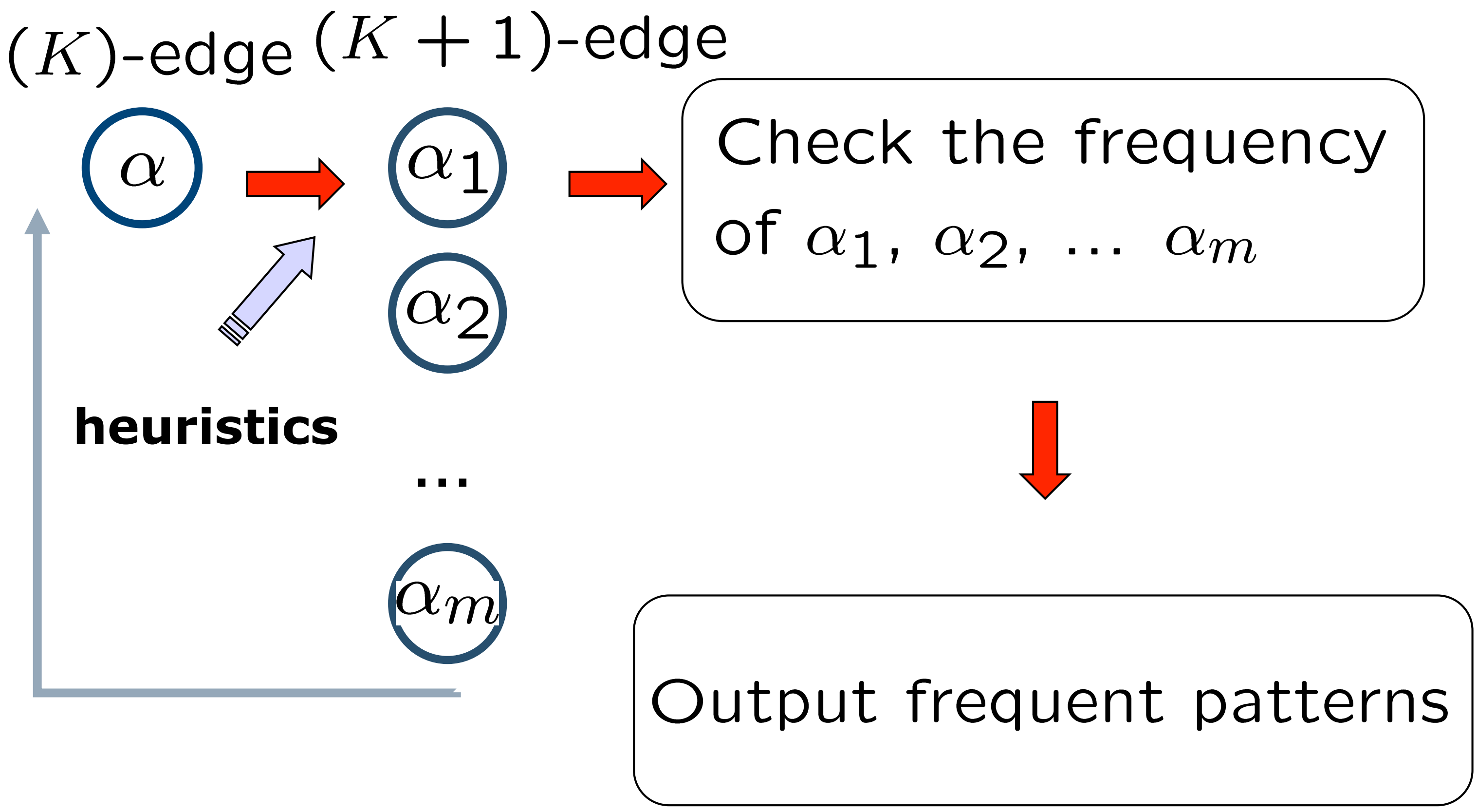
- Graphs are represented by Datalog facts

Graph Based Approaches

- Apriori-based approach
  - AGM/AcGM: Inokuchi, et al. (PKDD'00)
  - FSG: Kuramochi and Karypis (ICDM'01)
  - PATH#: Vanetik and Gudes (ICDM'02, ICDM'04)
  - FFSM: Huan, et al. (ICDM'03) and SPIN: Huan et al. (KDD'04)
  - FTOSM: Horvath et al. (KDD'06)
- Pattern growth approach
  - Subdue: Holder et al. (KDD'94)
  - MoFa: Borgelt and Berthold (ICDM'02)
  - gSpan: Yan and Han (ICDM'02)
  - Gaston: Nijssen and Kok (KDD'04)
  - CMTreeMiner: Chi et al. (TKDE'05), LEAP: Yan et al. (SIGMOD'08)

Apriori Property

If a graph is frequent, all of its subgraphs are frequent.



Cost Analysis

$$T_{total} \propto \sum_{\alpha} |D_\alpha| \times T_\alpha^{iso}$$

number of candidates

- frequent
- infrequent (X)
- duplicate (X)

data

isomorphism checking

Properties of Graph Mining Algorithms

Search Order

- breadth vs. depth
- complete vs. incomplete

Generation of Candidate Patterns

- apriori vs. pattern growth

Discovery Order of Patterns

- DFS order
- path → tree → graph

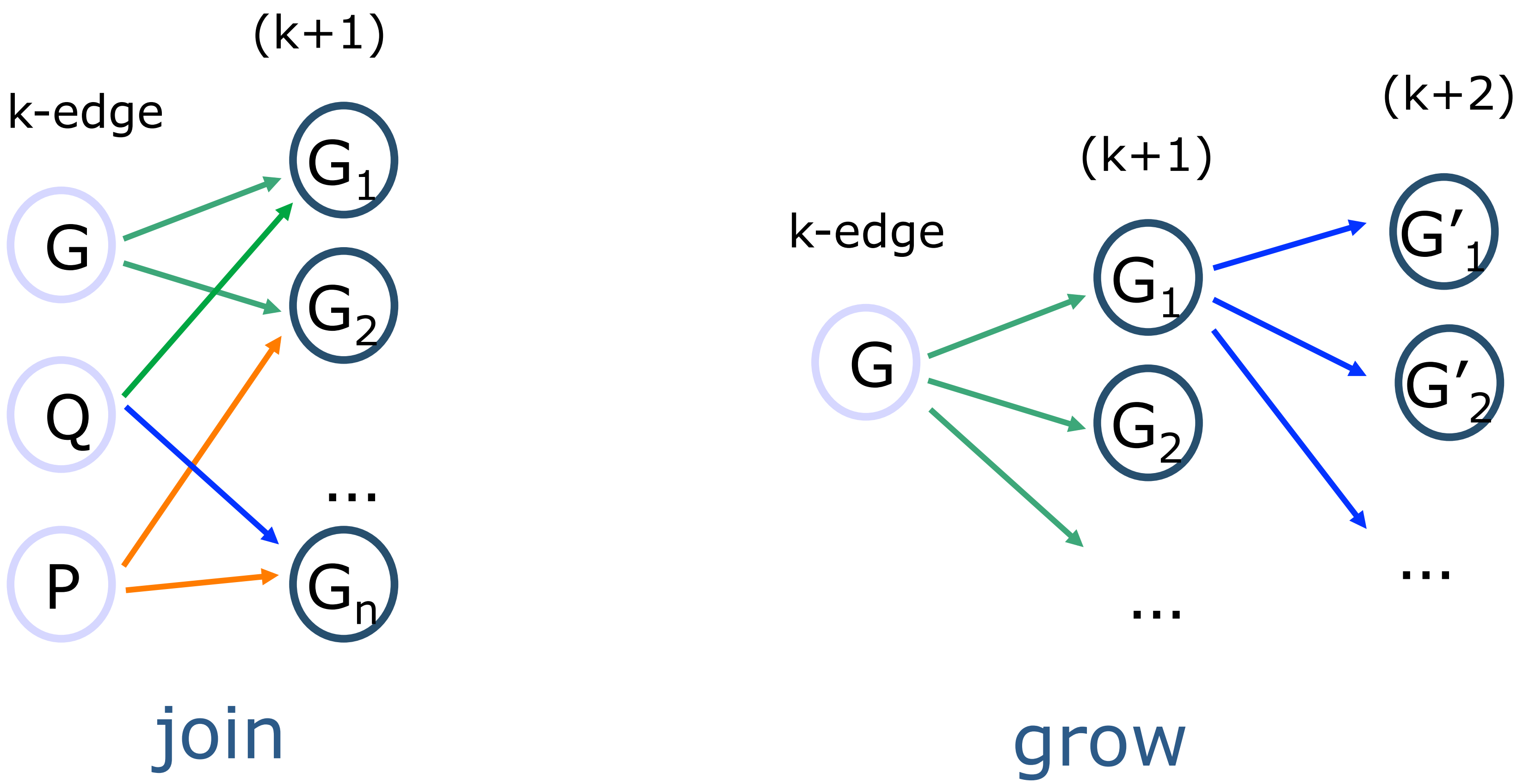
Elimination of Duplicate Subgraphs

- passive vs. active

Support Calculation

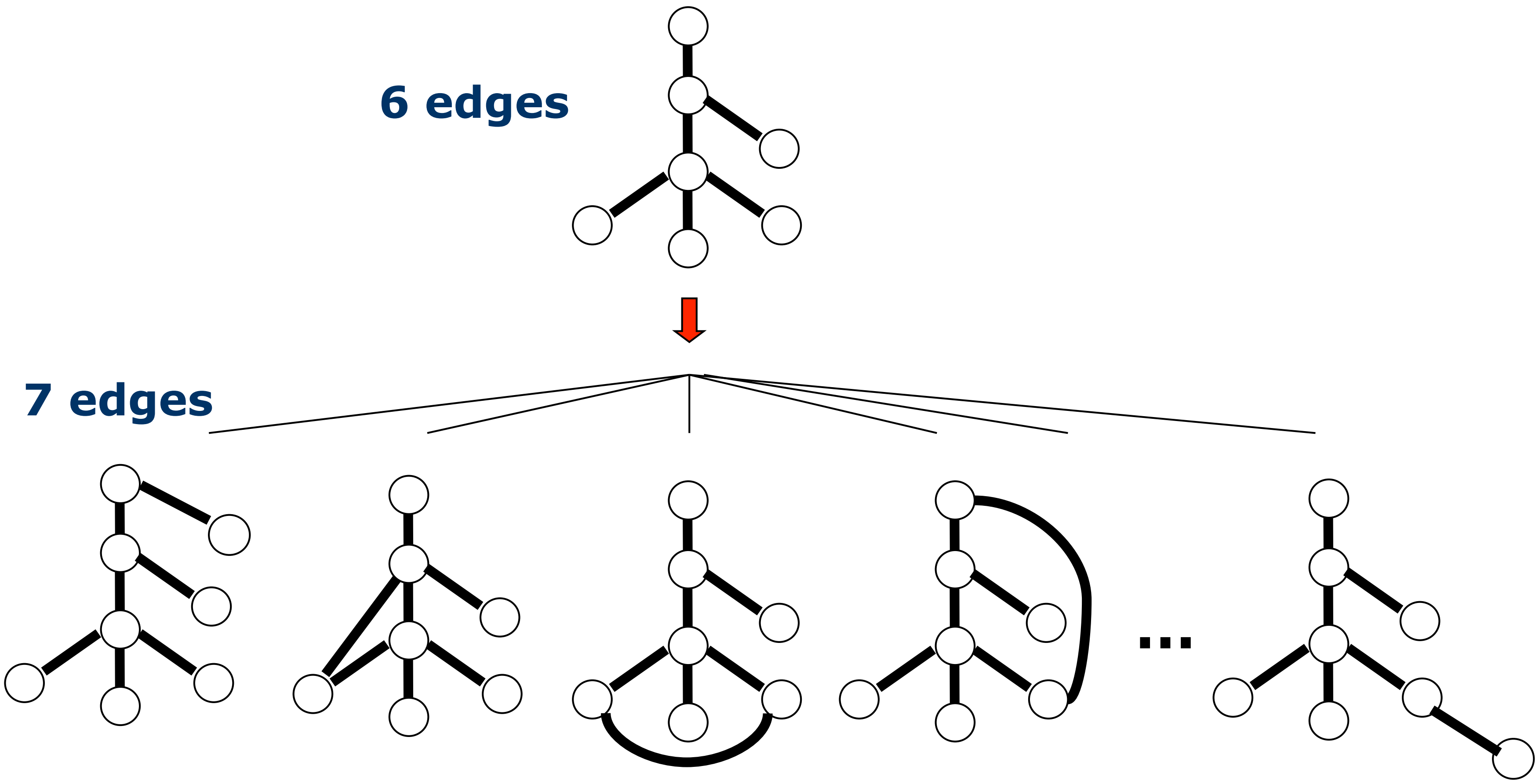
- embedding store or not

Generation of Candidate Patterns



Apriori-Based Approach VS. Pattern-Growth Approach

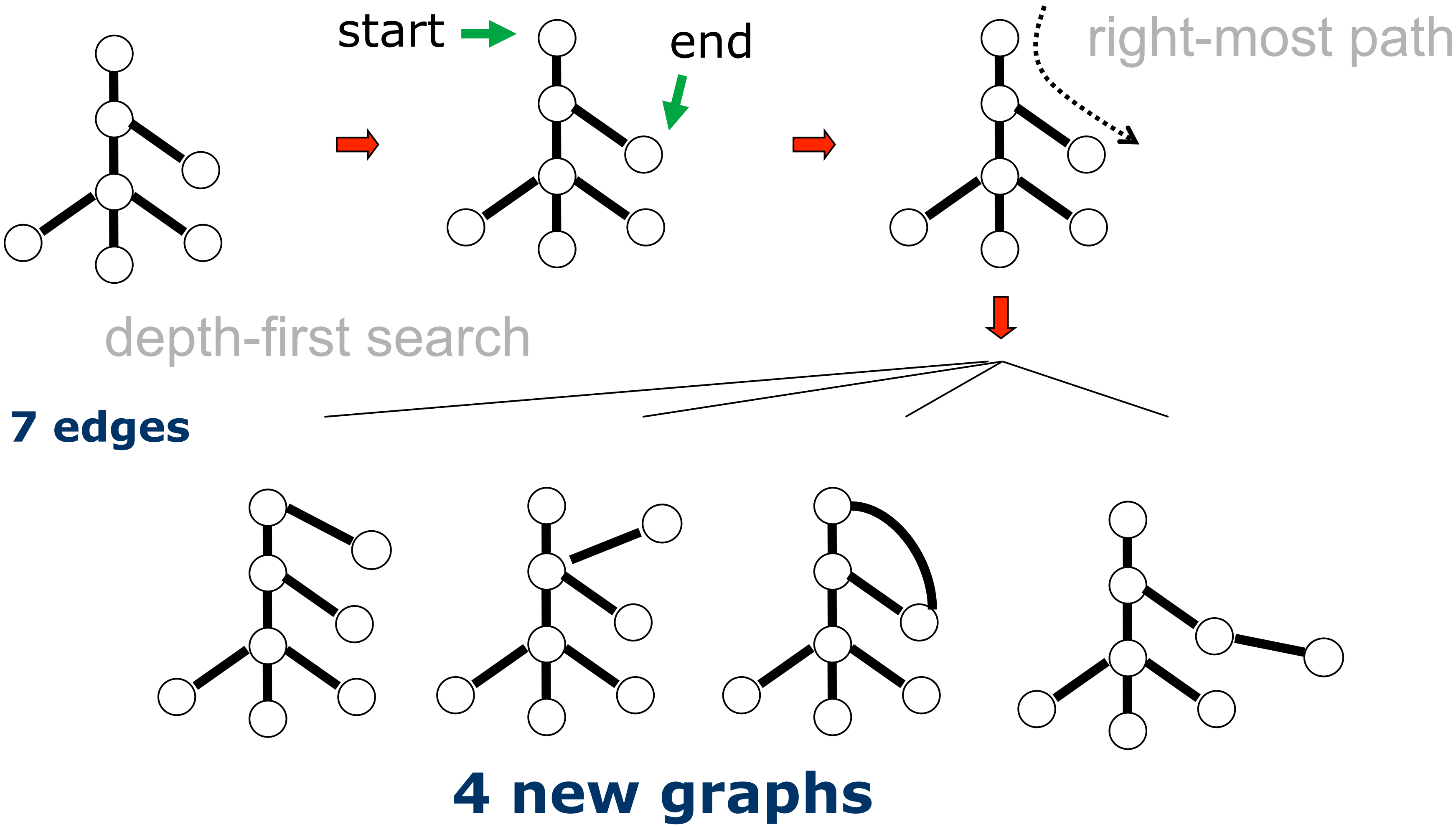
Discovery Order: Free Extension



22 new graphs

Discovery Order: Right-Most Extension

(Yan and Han ICDM'02)



Duplicates Elimination

Existing patterns $g_1, \ldots, g_N$

Newly discovered pattern g

Option 1

- Check graph isomorphism of *g* with each graph (slow)

Option 2

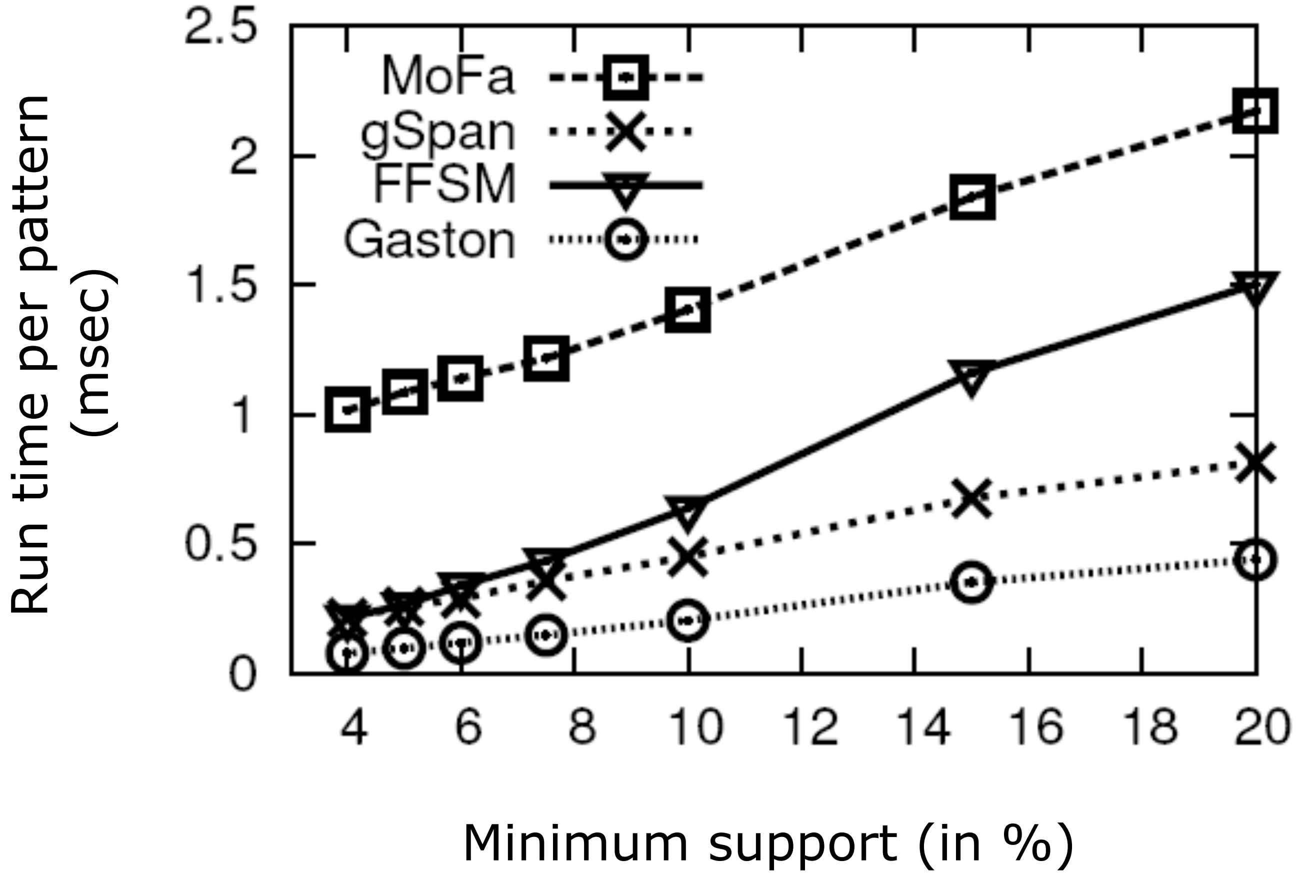
- Transform each graph to a canonical label, create a hash value for this canonical label, and check if there is a match with *g* (faster)

Option 3

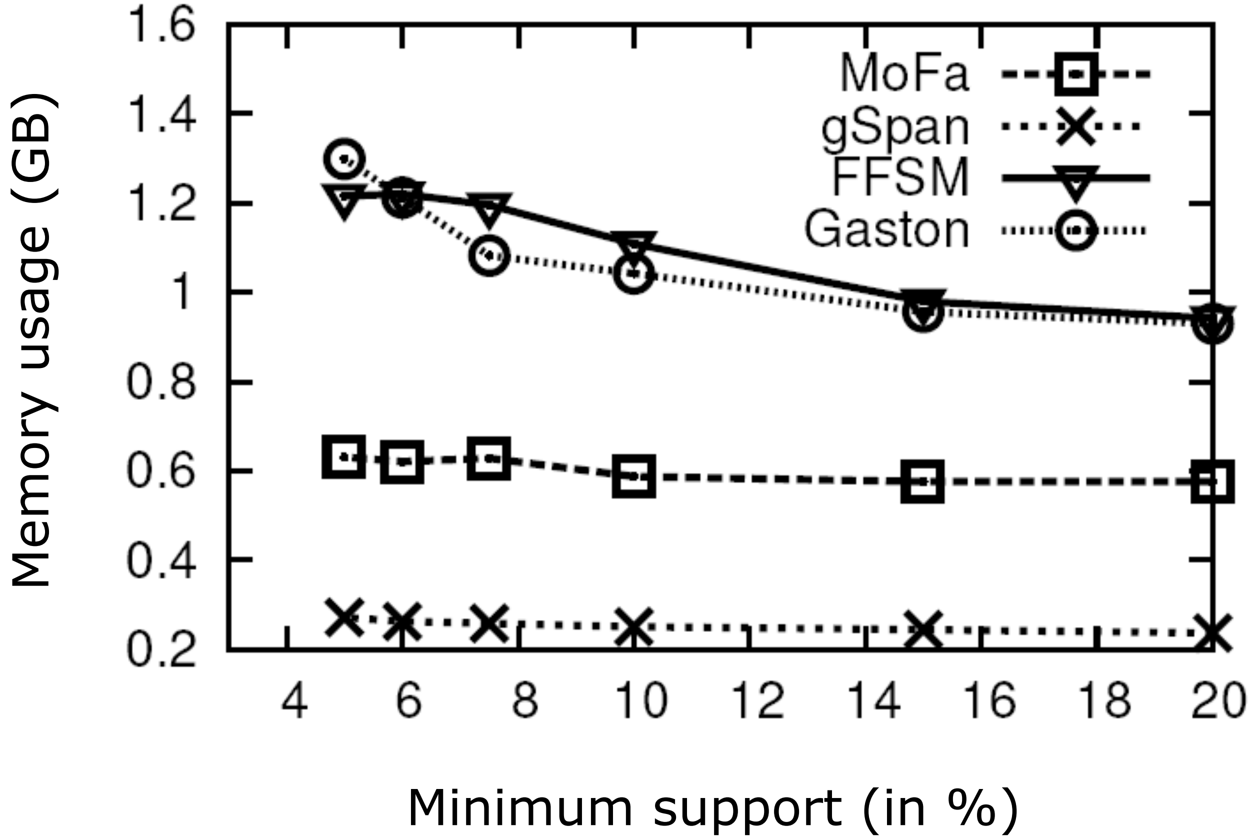
- Build a canonical order and generate graph patterns in that order (fastest)

Performance: Run Time (Wörlein et al. PKDD'05)

The AIDS antiviral screen compound dataset from NCI/NIH



Performance: Memory Usage (Wörlein et al. PKDD'05)



Graph Pattern Explosion Problem

- If a graph is frequent, all of its subgraphs are frequent ─ **the Apriori property**
- An **n**-edge frequent graph may have 2^n subgraphs!
- In the AIDS antiviral screen dataset with **400+** compounds, at the support level 5%, there are > 1M frequent graph patterns

Conclusions: Many enumeration algorithms are available

AGM, FSG, gSpan, Path-Join, MoFa, FFSM, SPIN, Gaston,

and so on, but three significant problems exist

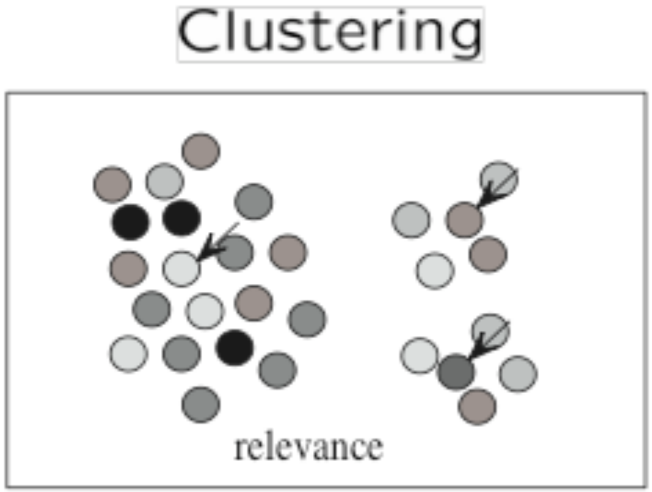
Problem 1: Interpretation Problem

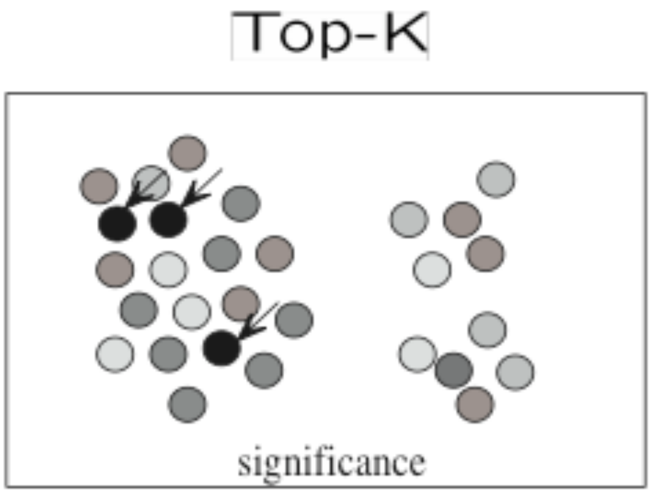
Problem 2: Exponential Pattern Set

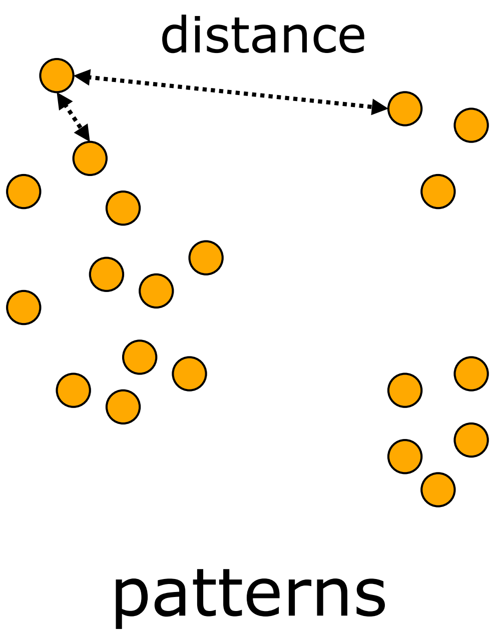
Problem 3: Threshold Setting

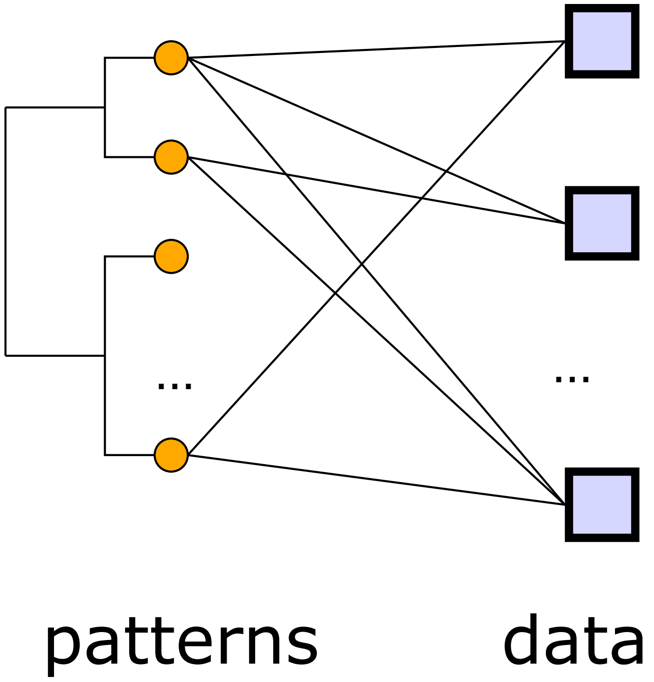
Pattern Summarization (Xin et al., KDD'06, Chen et al. CIKM'08)

- Too many patterns may not lead to more explicit knowledge
- It can confuse users as well as further discovery (e.g., clustering, classification, indexing, etc.)
- A small set of "representative" patterns that preserve most of the information









measure 1: pattern based

- pattern containment
- pattern similarity

measure 2: data based

- data similarity

Closed and Maximal Graph Pattern

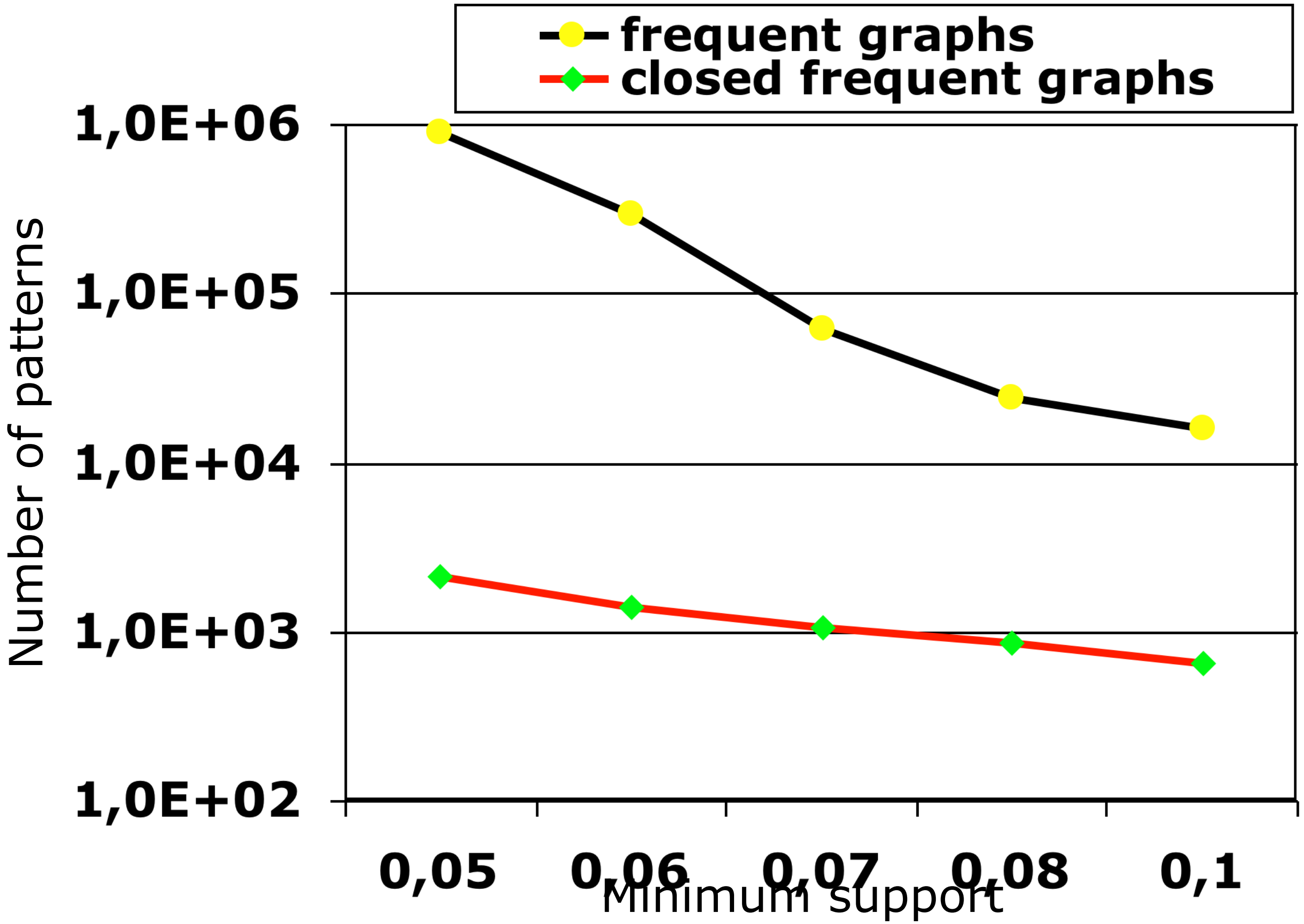
Closed Frequent Graph

- A frequent graph G is *closed* if there exists no supergraph of G that carries the same support as G
- If some of G's subgraphs have the same support, it is unnecessary to output these subgraphs (nonclosed graphs)
- *Lossless compression:* still ensures that the mining result is complete

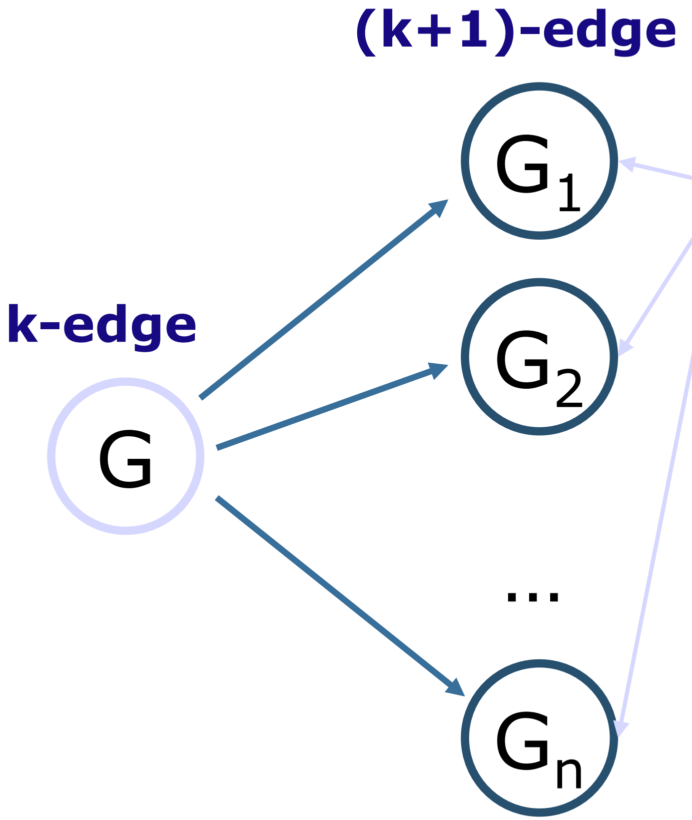
Maximal Frequent Graph

- A frequent graph G is *maximal* if there exists no supergraph of G that is frequent

Number of Patterns: Frequent vs. Closed



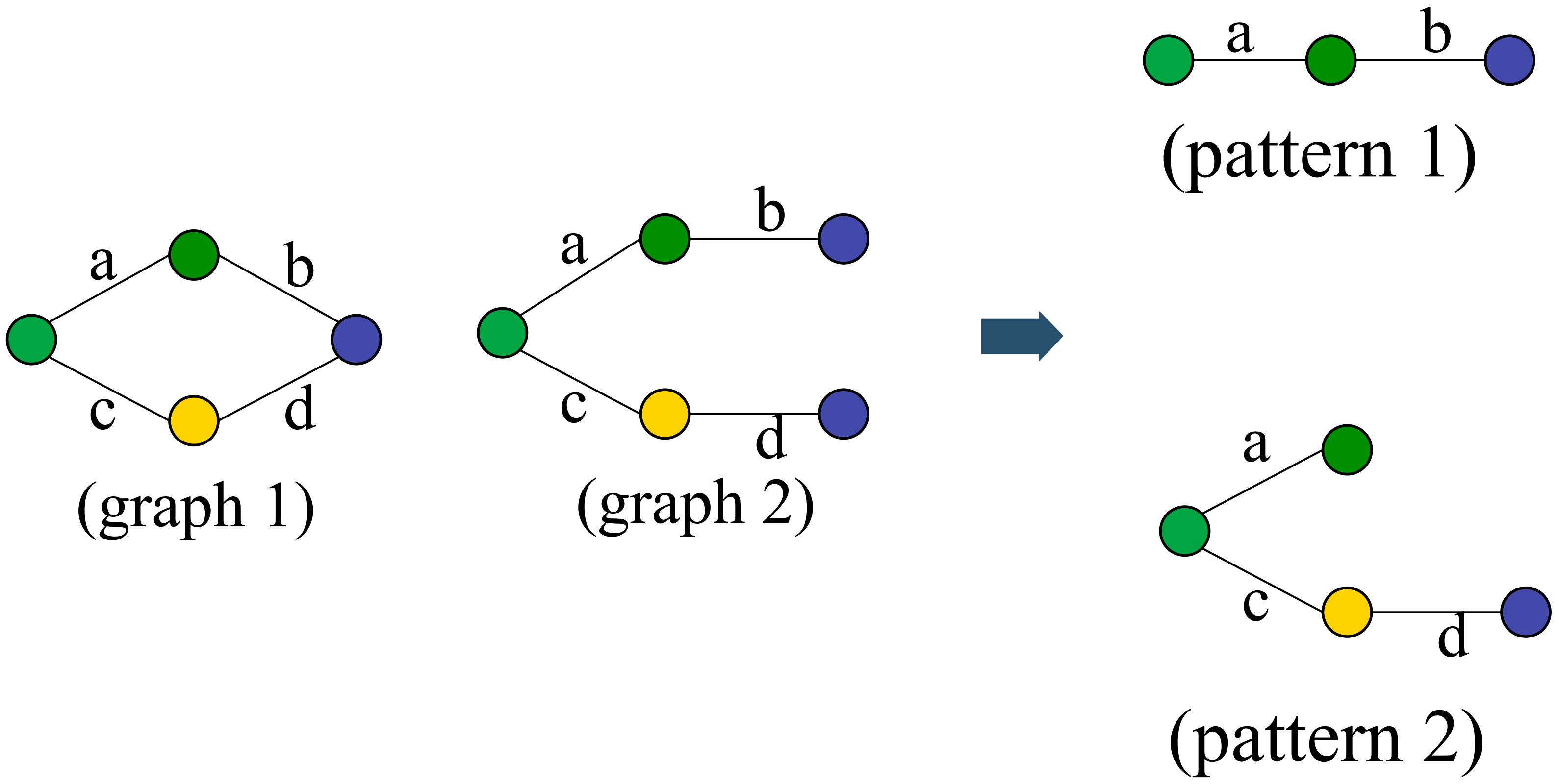
A Pattern-Growth Approach



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

If G and G′ are frequent, G is a subgraph of G′. If **in any part of graphs in the dataset where G occurs, G′ also occurs**, then we need not grow G, since none of G′s supergraphs will be closed except those of G′.

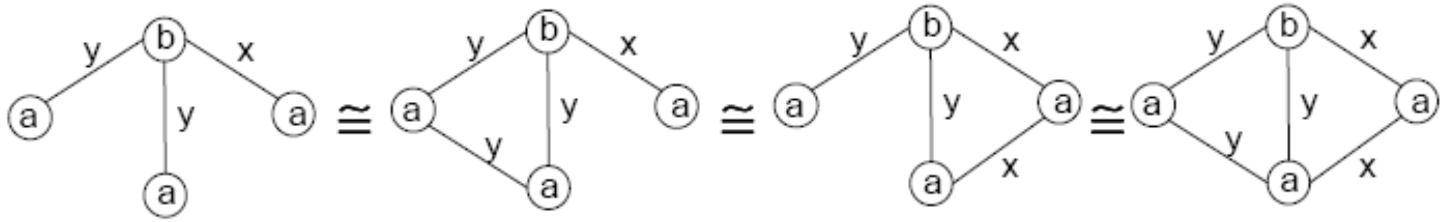
Handling Tricky Cases



Maximal Graph Pattern Mining (Huan et al. KDD'04)

Tree-based Equivalence Class

- Trees are sorted in their canonical order
- Graphs are in the same equivalence class if they have the same canonical spanning tree



Locally Maximal

- A frequent subgraph g is locally maximal if it is maximal in its equivalence class, i.e., g has no frequent supergraphs that share the same canonical spanning tree as g
- Every maximal graph pattern must be locally maximal
- Reduce enumeration of subgraphs that are not locally maximal

Graph Pattern with Other Measures

Let p and q be the frequency of g in positive and negative graph datasets,

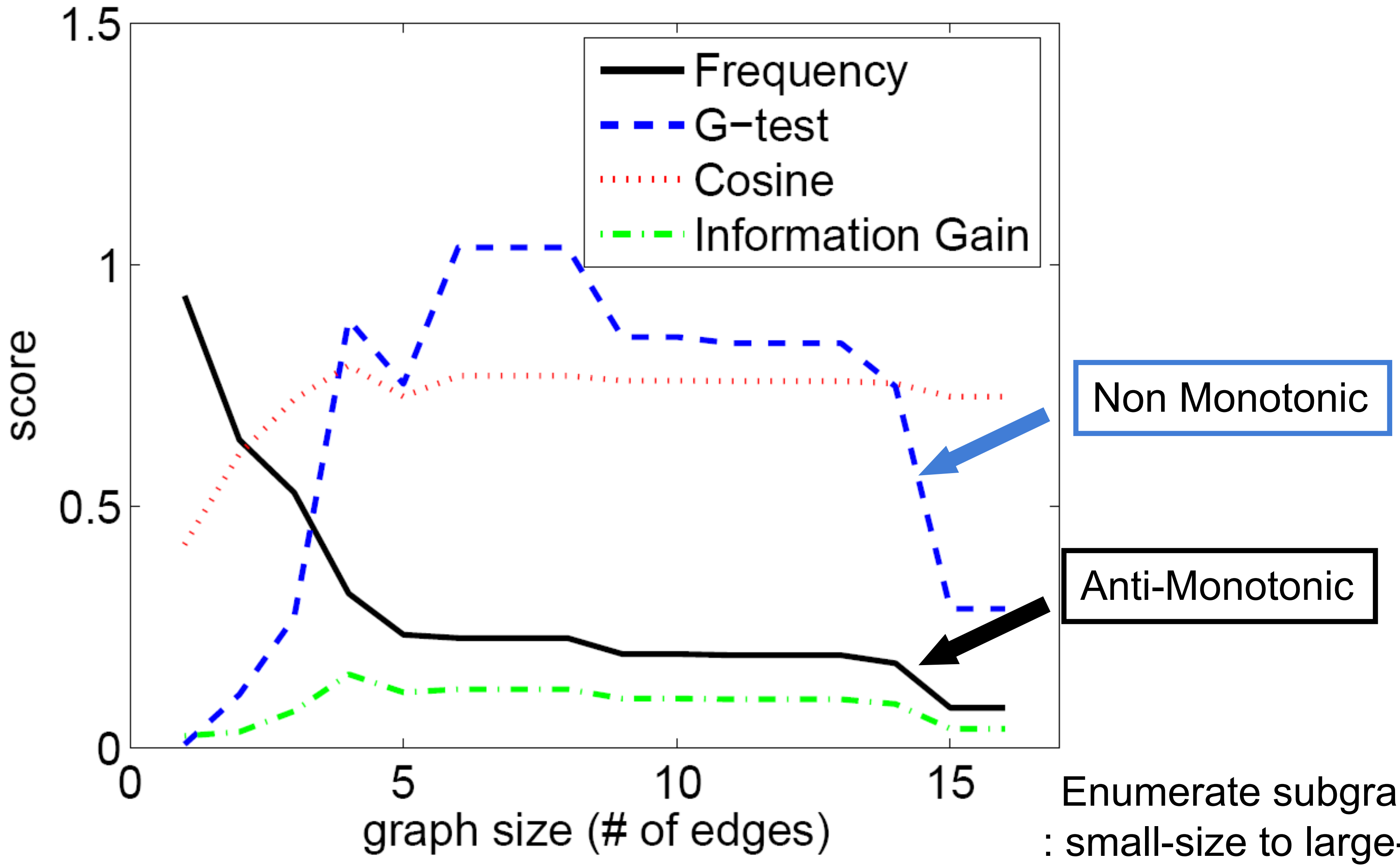
(1) Contrast: p/q,

(2) G-test: $p \cdot ln\frac{p}{q} + (1-p) \cdot ln\frac{1-p}{1-q}$,

(3) Information Gain: $H(C) - H(C|X)$

(4) Cosine

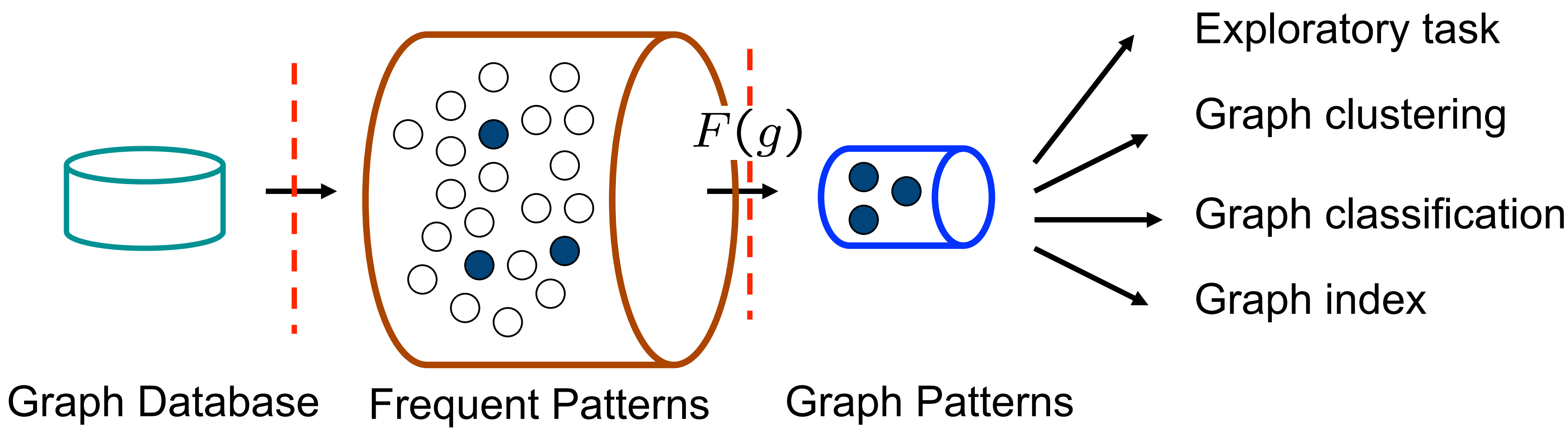
(5) many others.



Enumerate subgraphs
: small-size to large-size

Non-Monotonic: Enumerate all subgraphs, then check their score?

Frequent Pattern Based Mining Framework



1. Bottleneck : millions, even billions of patterns

2. No guarantee of quality

Optimal Graph Pattern

Given a graph dataset D and an objective function $F(g)$, find a graph pattern g^*, s.t.

$$g^* = arg\ max_g\ F(g).$$

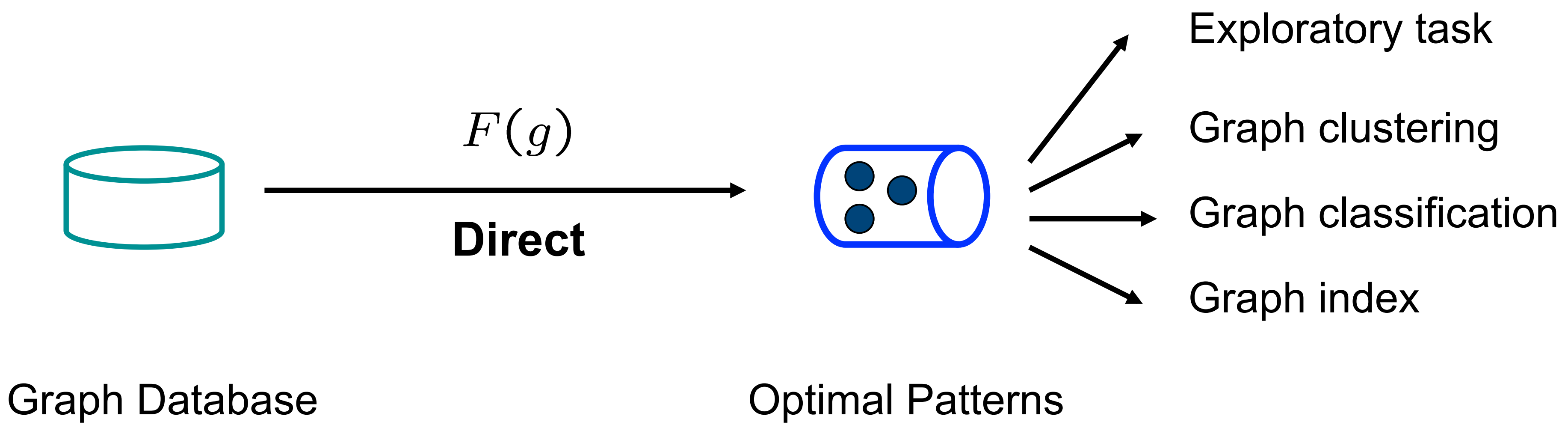
Extension:

Top-K Optimal Graph Patterns

Redundancy-aware Graph Patterns

Discriminative Patterns for Classification

Direct Pattern Mining Framework



Upper-Bound

Idea: derive an upper bound, $\widehat{F}(g)$, s.t., $\widehat{F}(g)$ is monotonic to freq(g).

$$G_t(p,q) = p \cdot ln\frac{p}{q} + (1-p) \cdot ln\frac{1-p}{1-q},$$

$$\begin{aligned}\frac{\partial G_t}{\partial q} &= \frac{q-p}{(1-q)q}, \\ \frac{\partial G_t}{\partial p} &= ln\frac{p(1-q)}{q(1-p)}.\end{aligned}$$

Since $\frac{p(1-q)}{q(1-p)} < 1$ when $p < q$, hence,

$$\text{if } p > q, \frac{\partial G_t}{\partial p} > 0, \frac{\partial G_t}{\partial q} < 0, \tag{1}$$

$$\text{if } p < q, \frac{\partial G_t}{\partial p} < 0, \frac{\partial G_t}{\partial q} > 0. \tag{2}$$

Upper-Bound: Anti-Monotonic (cont.)

$$\text{if } p > q, \frac{\partial G_t}{\partial p} > 0, \frac{\partial G_t}{\partial q} < 0, \quad (1)$$

$$\text{if } p < q, \frac{\partial G_t}{\partial p} < 0, \frac{\partial G_t}{\partial q} > 0. \quad (2)$$

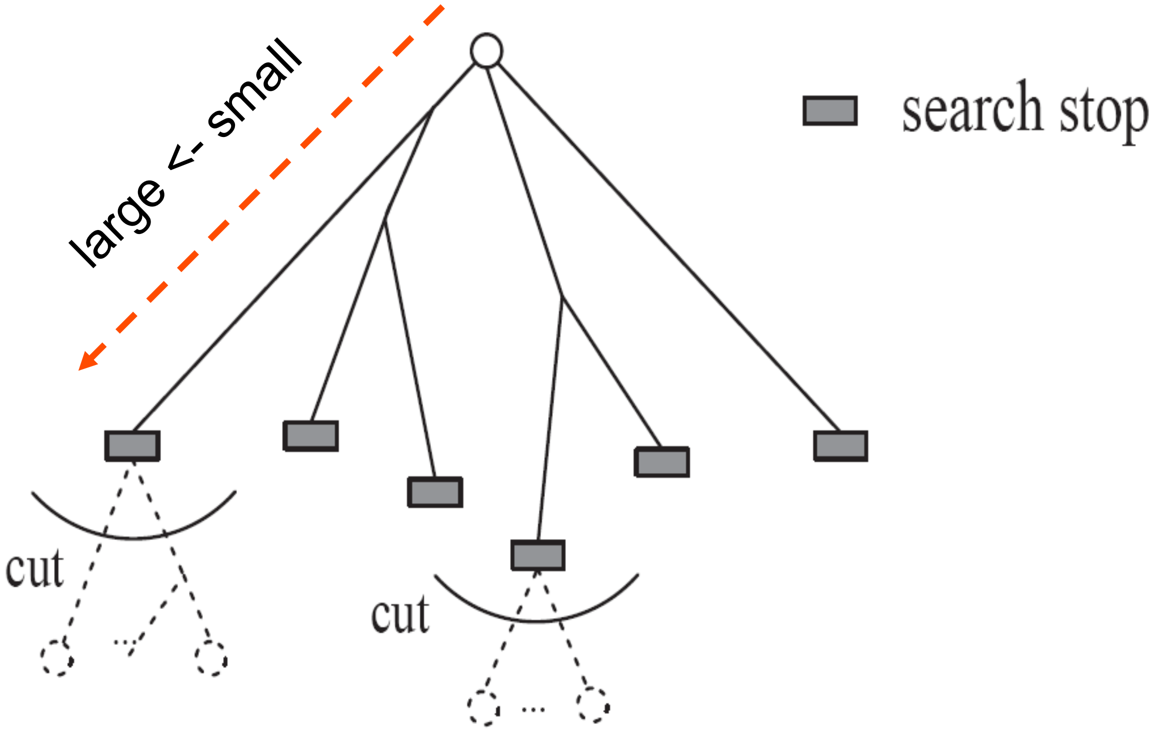
Rule of Thumb :
If the frequency difference of a graph pattern in the positive dataset and the negative dataset increases, the pattern becomes more interesting

small number

$$F(g) = F(p,q) < \max(\underbrace{F(p,\epsilon)}_{\text{Monotonic to p}}, \underbrace{F(\epsilon,q)}_{\text{Monotonic to q}}).$$

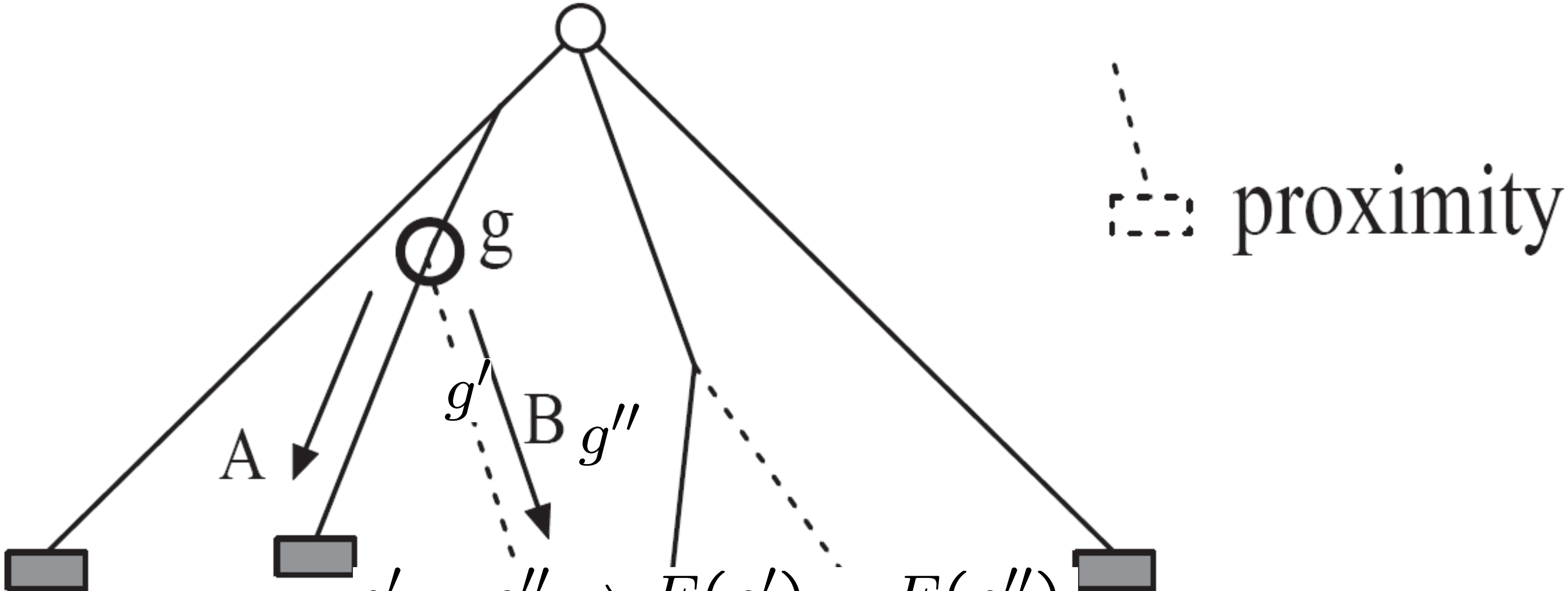
We can recycle the existing graph mining algorithms to accommodate non-monotonic functions.

Vertical Pruning



$$\max(F(p, \epsilon), F(\epsilon, q)) < F(g^*).$$

Horizontal Pruning: Structural Proximity



$$g' \sim g'' \Rightarrow F(g') \sim F(g'').$$

$$F(g') \ll F(g^*) \Rightarrow F(g'') \ll F(g^*).$$

Results: NCI Anti-Cancer Screen Datasets

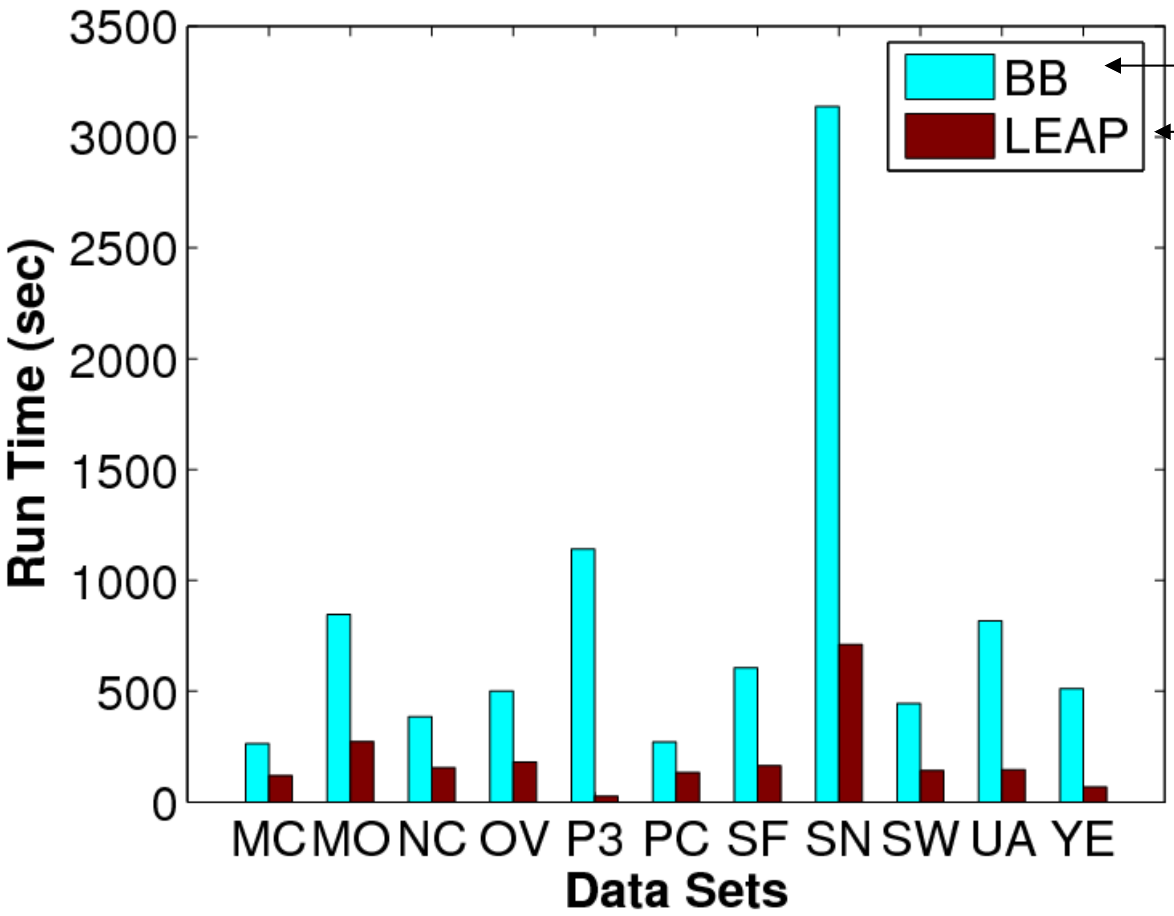
Chemical Compounds: anti-cancer or not

# of vertices: 10 ~ 200

Name	# of Compounds	Tumor Description
MCF-7	27,770	Breast
MOLT-4	39,765	Leukemia
NCI-H23	40,353	Non-Small Cell Lung
OVCAR-8	40,516	Ovarian
P388	41,472	Leukemia
PC-3	27,509	Prostate
SF-295	40,271	Central Nerve System
SN12C	40,004	Renal
SW-620	40,532	Colon
UACC257	39,988	Melanoma
YEAST	79,601	Yeast anti-cancer

Link: http://pubchem.ncbi.nlm.nih.gov

LEAP (Yan et al. SIGMOD'08)



Vertical Pruning

Vertical Pruning + Horizontal Pruning

Graph Pattern with Topological Constraints

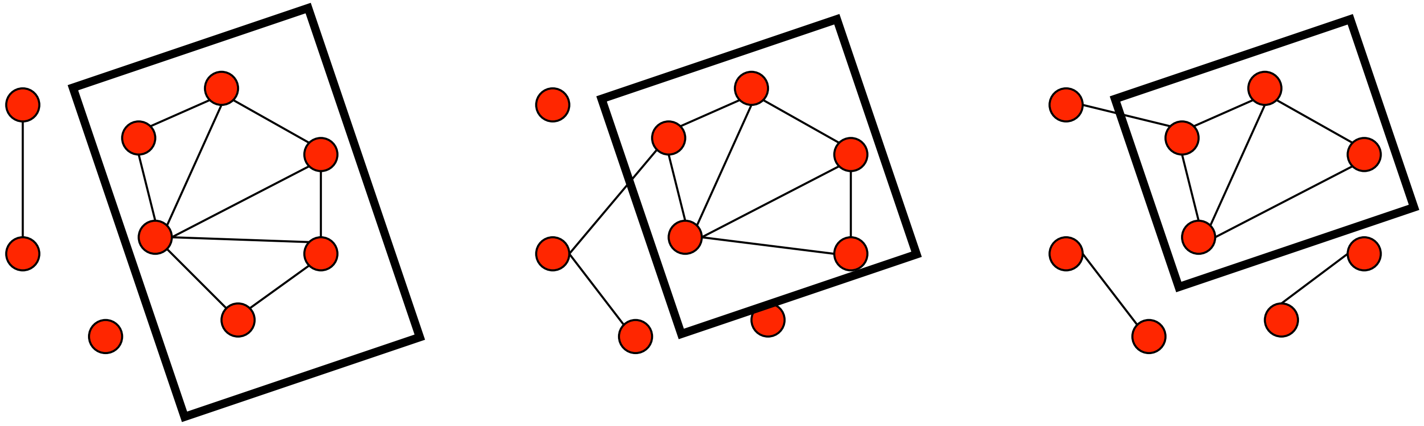
A constraint C is a boolean predicate, $C : P \rightarrow \{0, 1\}$, which maps a pattern α to a Boolean value. A pattern α satisfies constraint C if $C(\alpha) = 1$.

graph constraints

- Degree
- Size
- Density
- Density ratio
- Diameter
- Edge connectivity
- Vertex connectivity
- Aggregation (min, max, avg)

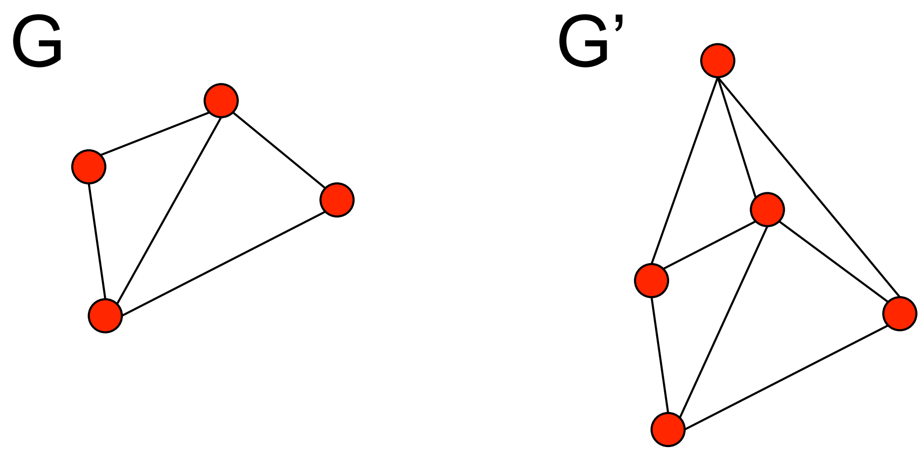
Constraint-Based Graph Pattern Mining

- Highly connected subgraphs in a large graph usually are not artifacts (group, functionality)

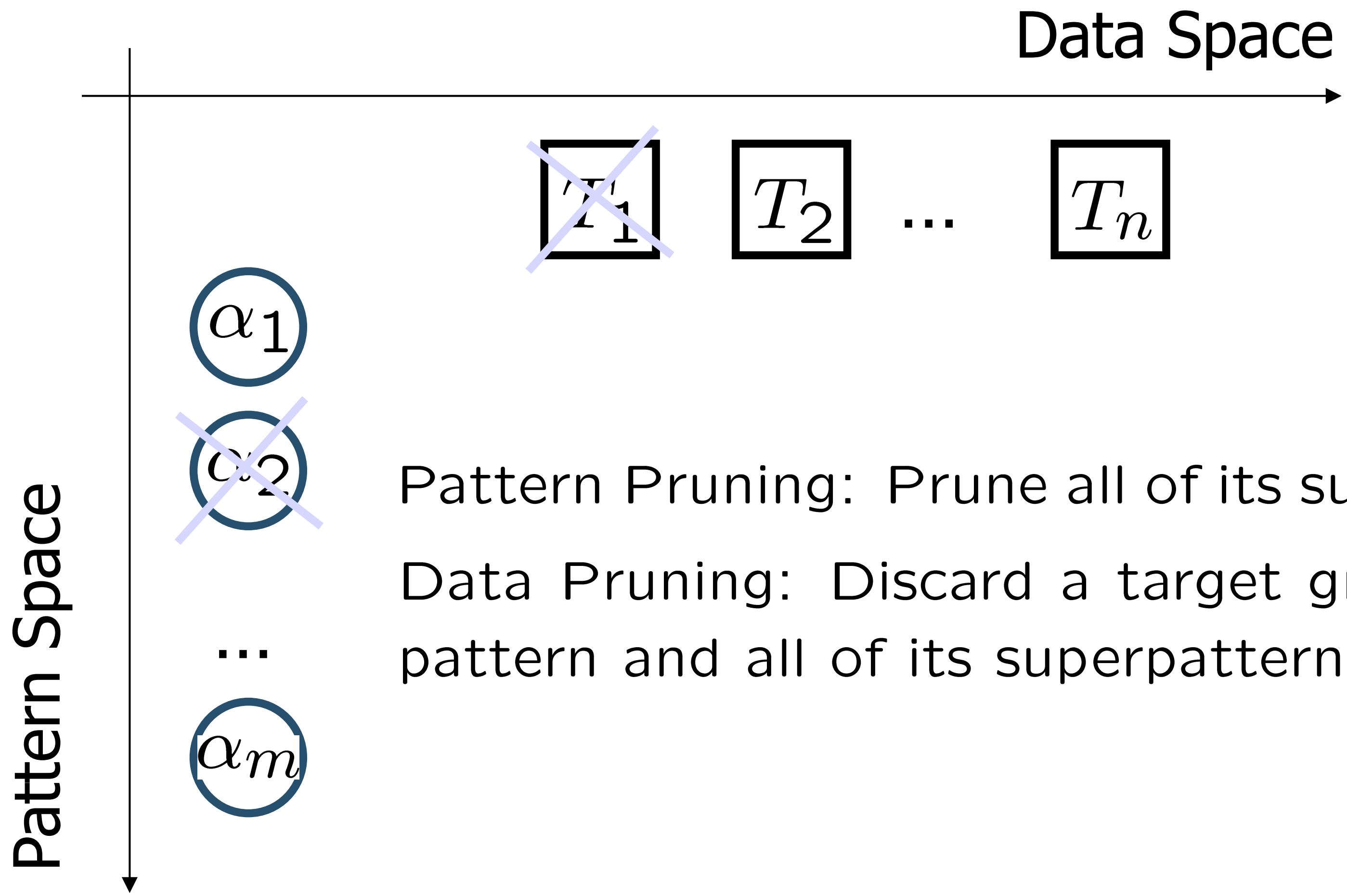


- Recurrent patterns discovered in multiple graphs are more robust than the patterns mined from a single graph

Given two graphs G and G', if G is a subgraph of G', it does not imply that the connectivity of G' is less than that of G, and vice versa.



Pruning Patterns vs. Data (Zhu et al. PAKDD'07)

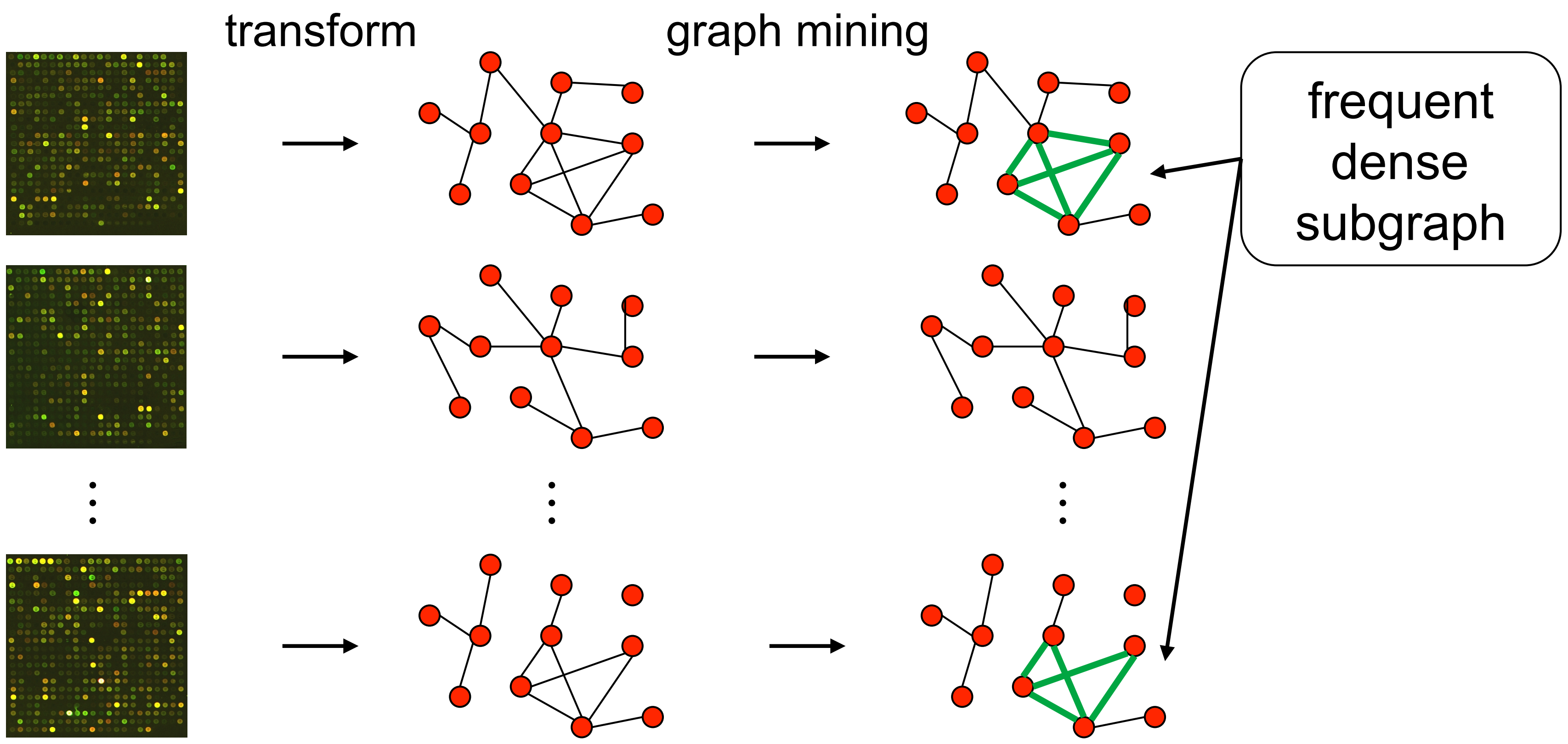


Pattern Pruning: Prune all of its superpatterns

Data Pruning: Discard a target graph for the pattern and all of its superpatterns

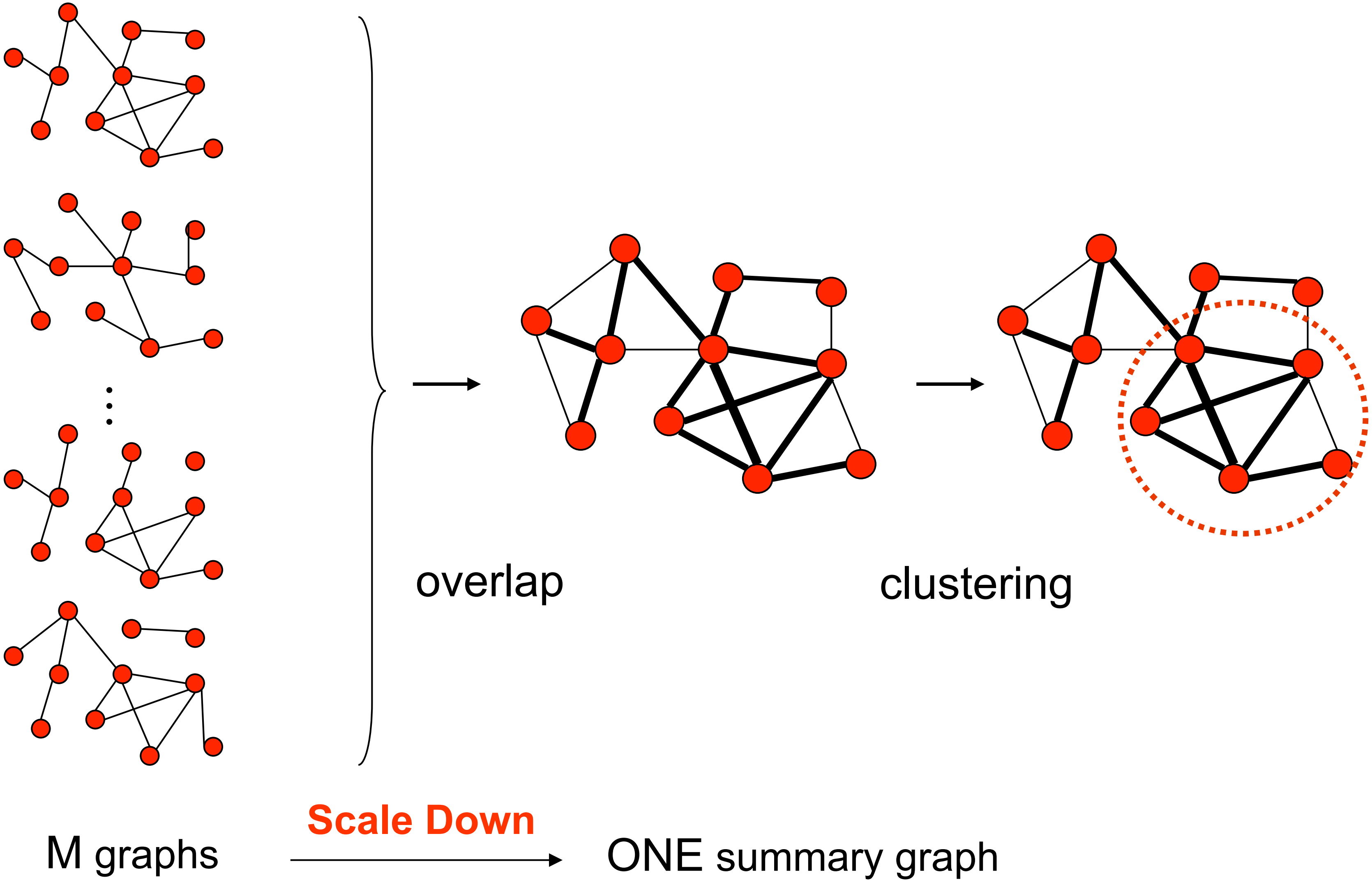
Mining Gene Co-expression Networks

Patterns discovered in multiple graphs are more reliable and significant



~9000 genes   150 x ~(9000 x 9000) = 12 billion edges

Summary Graph



Vertexlet (Yan et al. ISMB'07)

Vertexlet: a small subset of vertices.

Let π_u be the set of frequent dense $(k-1)$-vertexlets that contain vertex u and $\pi_{u,v}$ be the set of frequent dense k-vertexlets that contain vertices u and v.

$$score(u,v) = \frac{\pi_{u,v}}{\pi_u}$$

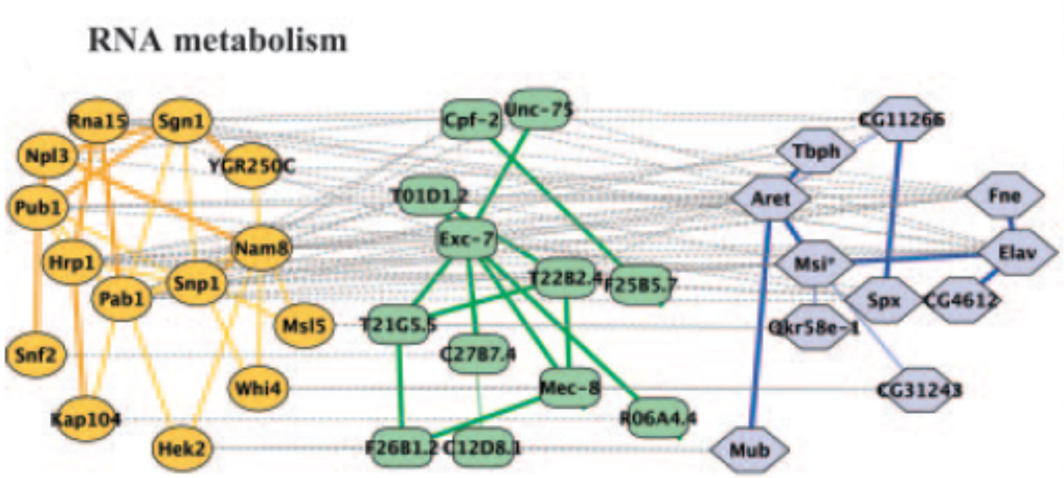
reweight the edge between u and v

PathBlast

- Exhaustive search: the highest-scoring paths with four nodes are identified

NetworkBlast

- Local search: start from high-scoring seeds, refine them, and expand them
- Filter overlapping graph patterns



Conserved clusters within the protein interaction networks of yeast, worm, and fly

Graph Classification

Structure-based Approach

- Local structures in a graph, e.g., neighbors surrounding a vertex, paths with fixed length

Pattern-based Approach

- Subgraph patterns from domain knowledge or from graph mining
- Decision Tree (Fan et al. KDD'08)
- Boosting (Kudo et al. NIPS'04)
- LAR-LASSO (Tsuda, ICML'07)

Kernel-based Approach

- Random walk (Gärtner '02, Kashima et al. '02, ICML'03, Mahé et al. ICML'04)

Structure/Pattern-based Classification

Basic Idea

- Transform each graph in the dataset into a feature vector,

$$G \rightarrow \mathbf{x} = \{x_1, x_2, \ldots, x_n\}$$

where x_i is the frequency of the i-th structure/pattern in G_i. Each vector is associated with a class label. Classify these vectors in a vector space

Structure Features

- Local structures in a graph, e.g., neighbors surrounding a vertex, paths with fixed length
- Subgraph patterns from domain knowledge
  - Molecular descriptors
- Subgraph patterns from data mining

Enumerate all of the subgraphs and select the best features?

Graph Patterns from Data Mining

- Sequence patterns (De Raedt and Kramer IJCAI'01)
- Frequent subgraphs (Deshpande et al, ICDM'03)
- Coherent frequent subgraphs (Huan et al. RECOMB'04)
  - A graph G is *coherent* if the mutual information between G and each of its own subgraphs is above some threshold

$$p(X_G = 1) = \text{frequency of } G$$

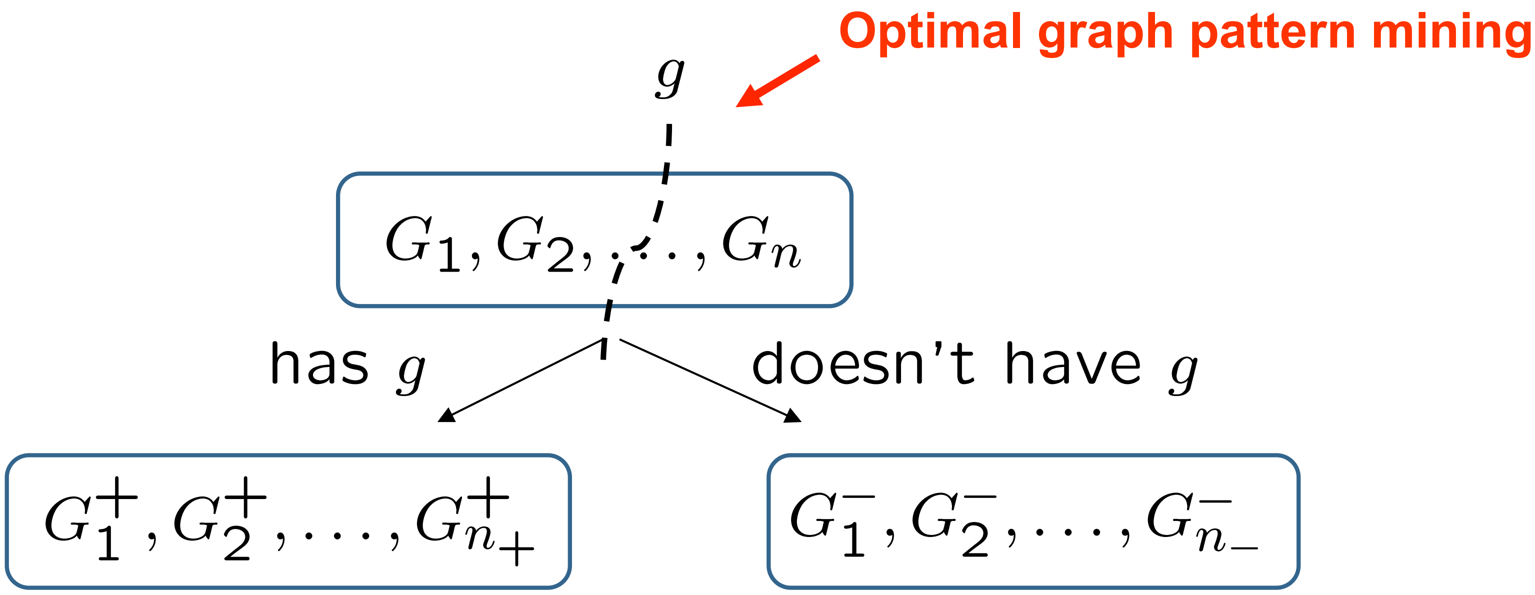
$$I(G, G') = \sum_{X_G, X_{G'}} p(X_G, X_{G'}) log \frac{p(X_G, X_{G'})}{p(X_G) p(X_{G'})}$$

- Closed frequent subgraphs (Liu et al. SDM'05)
- Acyclic Subgraphs (Wale and Karypis, technical report '06)

Decision-Tree (Fan et al. KDD'08)

Basic Idea

- Partition the data in a top-down manner and construct the tree using the best feature at each step according to some criterion
- Partition the data set into two subsets, one containing this feature and the other does not



Boosting in Graph Classification (Kudo et al. NIPS'04)

Simple classifiers: A rule is a tuple $<t,y>$.

If a molecule contains substructure t, it is classified as y.

$$h_{<t,y>}(\mathbf{x}) = \begin{cases} y & \text{if } t \subseteq \mathbf{x}, \\ -y & otherwise. \end{cases}$$

- Gain

$$gain(< t, y >) = \sum_{i=1}^{n} y_i h_{<t,y>}(\mathbf{x}_i)$$

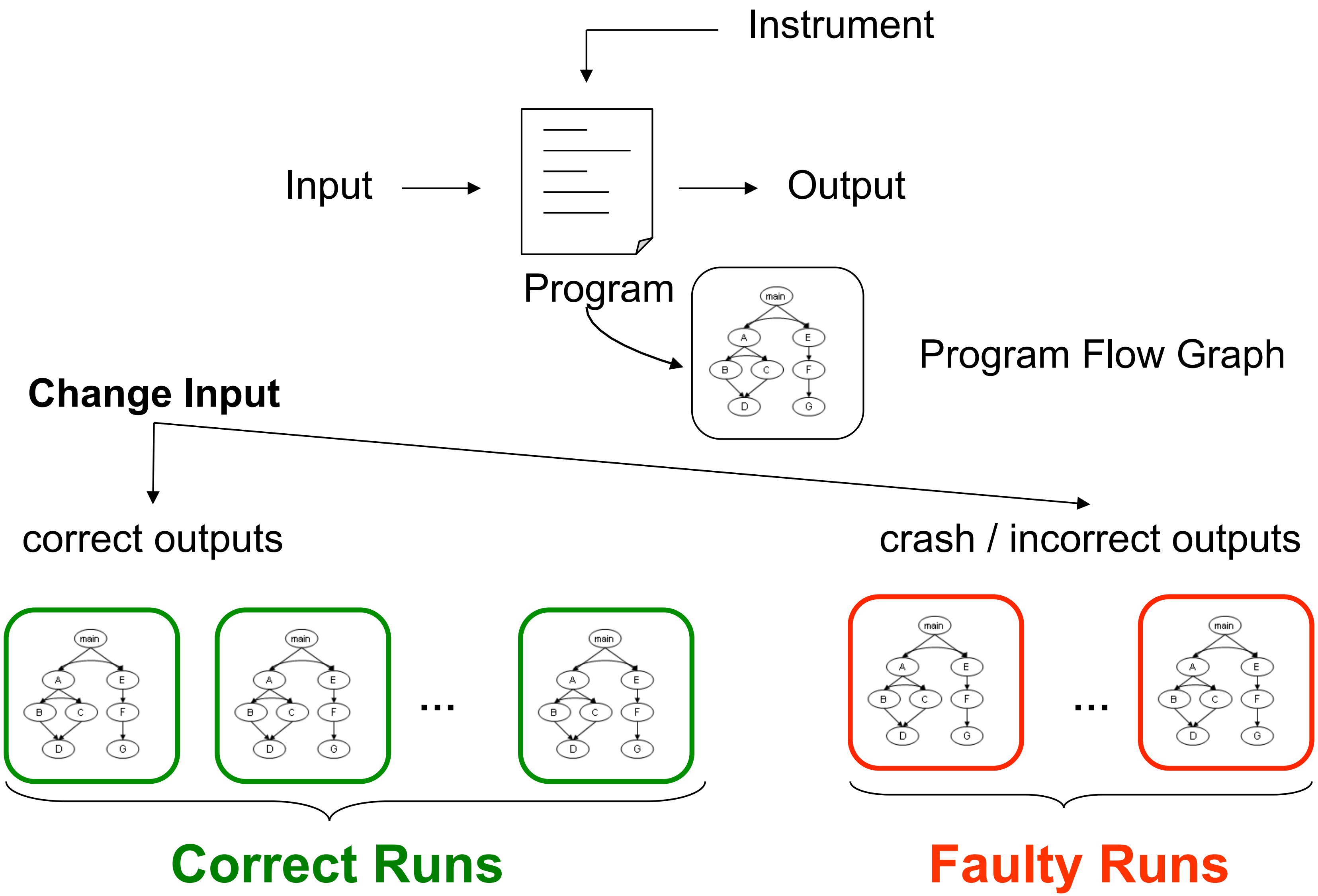
Optimal graph pattern mining

$$gain(< t, y >) = \sum_{i=1}^{n} y_i \boxed{d_i} h_{<t,y>}(\mathbf{x}_i)$$

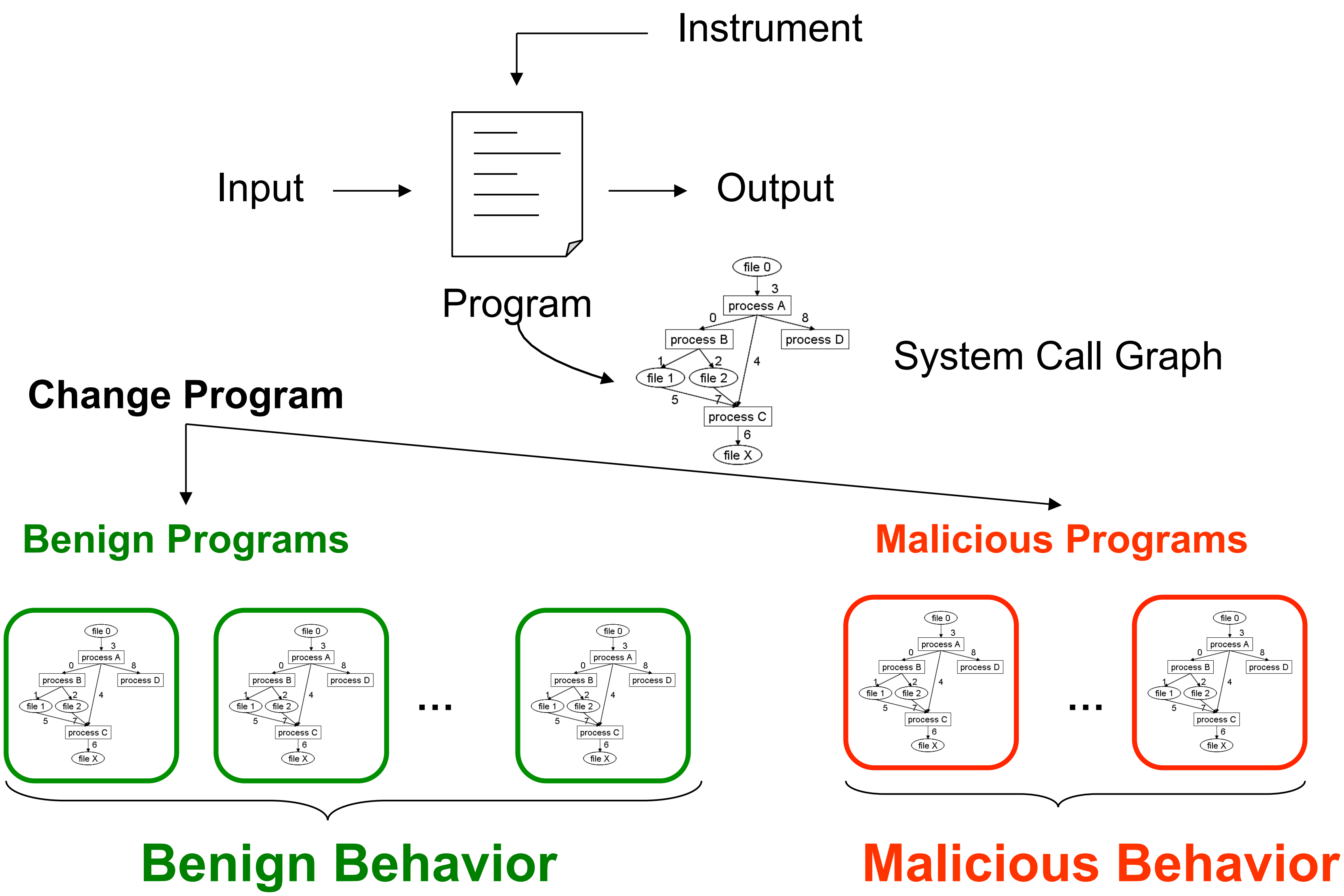
- Applying boosting

New Development: Graph in LAR-LASSO (Tsuda, ICML'07)

Graph Classification for Bug Isolation
(Chao et al. FSE'05, SDM'06)

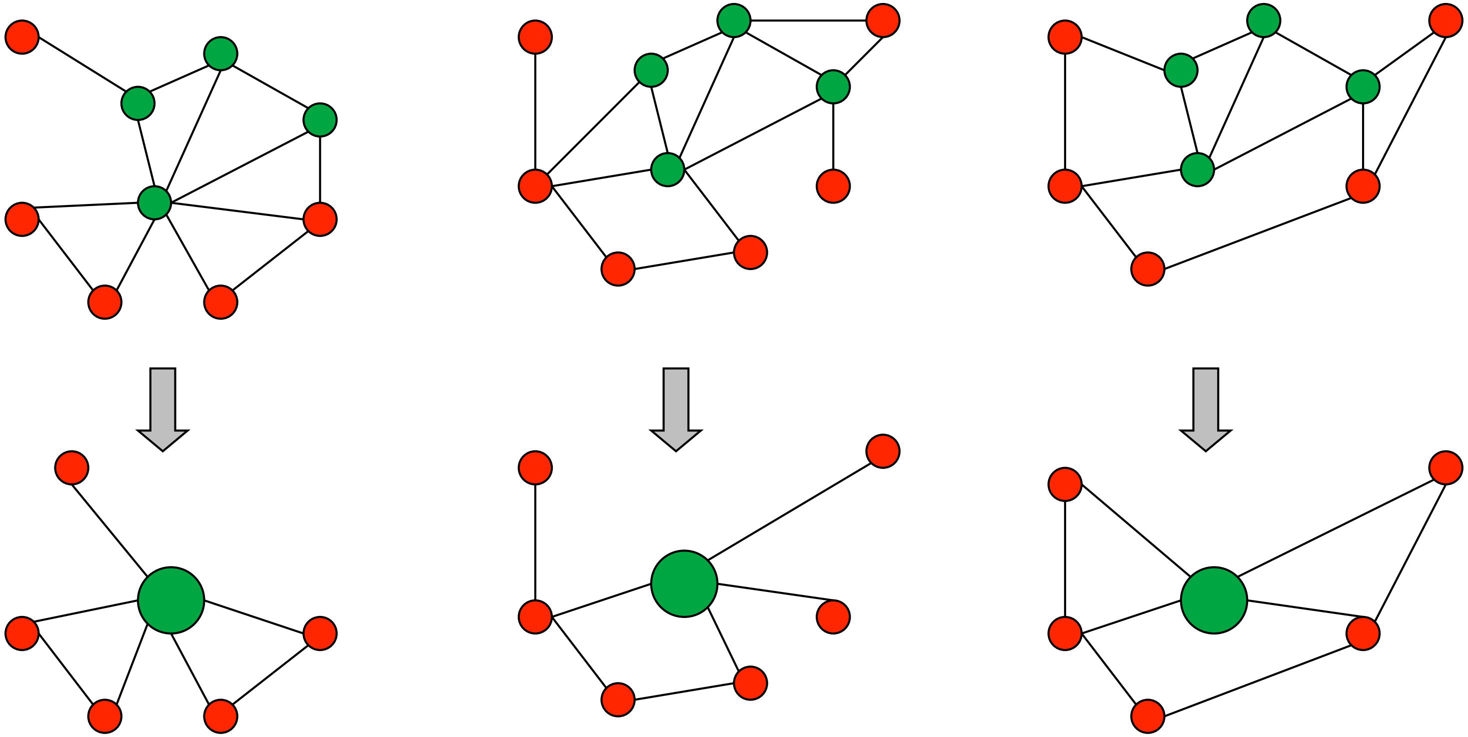


Graph Classification for Malware Detection



Graph Compression (Holder et al., KDD'94)

Extract common subgraphs and simplify graphs by condensing these subgraphs into nodes



Conclusions

Graph mining from a pattern discovery perspective

- Graph Pattern Mining
- Graph Classification
- Graph Compression

Other Interesting Topics

- Graph Model, Laws, and Generators
- Graph Dynamics
- Social Network Analysis
- Graph Summarization
- Graph Visualization
- Graph Clustering
- Link Analysis

References (1)

- T. Asai, et al. “Efficient substructure discovery from large semi-structured data”, SDM'02
- F. Afrati, A. Gionis,and H. Mannila, “Approximating a collection of frequent sets”, KDD’04
- C. Borgelt and M. R. Berthold, “Mining molecular fragments: Finding relevant substructures of molecules”, ICDM'02
- Y. Chi, Y. Xia, Y. Yang, R. Muntz, “Mining closed and maximal frequent subtrees from databases of labeled rooted trees,” TKDE 2005
- M. Deshpande, M. Kuramochi, and G. Karypis, “Frequent substructure based approaches for classifying chemical compounds”, ICDM’03
- M. Deshpande, M. Kuramochi, and G. Karypis. “Automated approaches for classifying structures”, BIOKDD'02
- L. Dehaspe, H. Toivonen, and R. King. “Finding frequent substructures in chemical compounds,” KDD'98
- C. Faloutsos, K. McCurley, and A. Tomkins, “Fast discovery of connection subgraphs”, KDD'04
- W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. S. Yu, O. Verscheure, “Direct mining of discriminative and essential graphical and itemset features via model-based search tree,” KDD'08
- H. Fröhlich, J. Wegner, F. Sieker, and A. Zell, “Optimal assignment kernels for attributed molecular graphs”, ICML’05
- T. Gärtner, P. Flach, and S. Wrobel, “On graph kernels: Hardness results and efficient alternatives”, COLT/Kernel’03

References (2)

- L. Holder, D. Cook, and S. Djoko, “Substructure discovery in the subdue system”, KDD'94
- T. Horváth, J. Ramon, and S. Wrobel, “Frequent subgraph mining in outerplanar graphs,” KDD’06
- J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, and A. Tropsha. “Mining spatial motifs from protein structure graphs”, RECOMB’04
- J. Huan, W. Wang, and J. Prins, “Efficient mining of frequent subgraph in the presence of isomorphism”, ICDM'03
- J. Huan, W. Wang, and J. Prins, and J. Yang, “SPIN: Mining maximal frequent subgraphs from graph databases”, KDD’04
- A. Inokuchi, T. Washio, and H. Motoda. “An apriori-based algorithm for mining frequent substructures from graph data”, PKDD'00
- H. Kashima, K. Tsuda, and A. Inokuchi, “Marginalized kernels between labeled graphs”, ICML’03
- B. Kelley, R. Sharan, R. Karp, E. Sittler, D. Root, B. Stockwell, and T. Ideker, “Conserved pathways within bacteria and yeast as revealed by global protein network alignment,” PNAS, 2003
- R. King, A Srinivasan, and L Dehaspe, "Warmr: a data mining tool for chemical data," J Comput Aided Mol Des 2001

References (3)

- M. Koyuturk, A. Grama, and W. Szpankowski. “An efficient algorithm for detecting frequent subgraphs in biological networks”, Bioinformatics, 20:I200--I207, 2004
- C. Liu, X. Yan, H. Yu, J. Han, and P. S. Yu, “Mining behavior graphs for ‘backtrace" of noncrashing bugs,'‘ SDM'05
- T. Kudo, E. Maeda, and Y. Matsumoto, “An application of boosting to graph classification”, NIPS’04
- M. Kuramochi and G. Karypis. “Frequent subgraph discovery”, ICDM'01
- M. Kuramochi and G. Karypis, “GREW: A scalable frequent subgraph discovery algorithm”, ICDM’04
- P. Mahé, N. Ueda, T. Akutsu, J. Perret, and J. Vert, “Extensions of garginalized graph kernels”, ICML’04
- B. McKay. Practical graph isomorphism. Congressus Numerantium, 30:45--87, 1981.
- S. Nijssen and J. Kok, “A quickstart in frequent structure mining can make a difference,” KDD'04
- R. Sharan, S. Suthram, R. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. Karp, and T. Ideker, “Conserved patterns of protein interaction in multiple species,” PNAS, 2005
- J. R. Ullmann. “An algorithm for subgraph isomorphism”, J. ACM, 23:31--42, 1976.
- N. Vanetik, E. Gudes, and S. E. Shimony. “Computing frequent graph patterns from semistructured data”, ICDM'02
- K. Tsuda, “Entire regularization paths for graph data,” ICML’07

References (4)

- N. Wale and G. Karypis, "Acyclic subgraph based descriptor spaces for chemical compound retrieval and classification", Univ. of Minnesota, Technical Report: #06–008
- C. Wang, W. Wang, J. Pei, Y. Zhu, and B. Shi. "Scalable mining of large disk-base graph databases", KDD'04
- T. Washio and H. Motoda, "State of the art of graph-based data mining," SIGKDD Explorations, 5:59-68, 2003
- M. Wörlein, T. Meinl, I. Fischer, M. Philippsen, "A quantitative comparison of the subgraph miners MoFa, gSpan, FFSM, and Gaston," PKDD'05
- X. Yan, H. Cheng, J. Han, and P. S. Yu, "Mining significant graph patterns by leap search," SIGMOD'08
- 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, X. Zhou, and J. Han, "Mining closed relational graphs with connectivity constraints", KDD'05
- X. Yan et al. "A graph-based approach to systematically reconstruct human transcriptional regulatory modules," ISMB'07
- M. Zaki. "Efficiently mining frequent trees in a forest", KDD'02
- Z. Zeng, J. Wang, L. Zhou, G. Karypis, "Coherent closed quasi-clique discovery from large dense graph databases," KDD'06