

頻出部分グラフマイニング の原理と応用

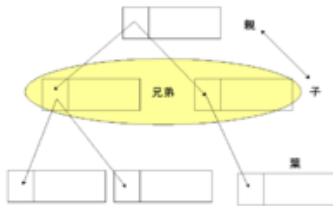
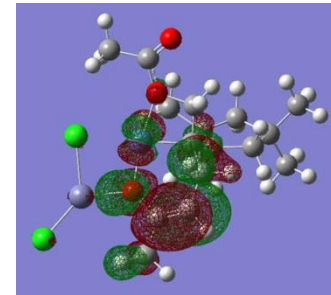
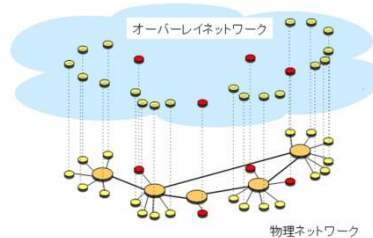
2023年10月-11月

鷺尾隆

大阪大学産業科学研究所

背景

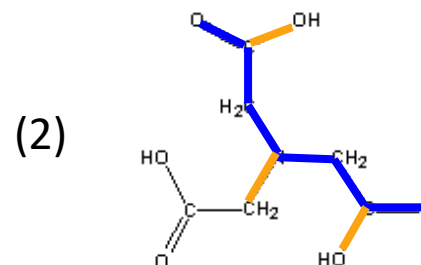
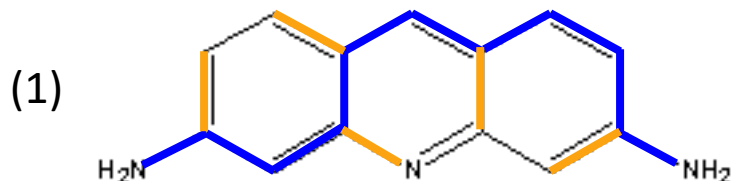
- 世の中にネットワークや木、系列など構造を持つデータが沢山蓄積されている。



- 蓄積されたデータから特定の構造を持つものを検索したり、特徴的部分構造を見つけたい。

列挙による頻出構造マイニング以前の グラフ構造データ(化学分子)

- CASE, MultiCASE[Klopman,84,92]
 - 化学分子中で枝分かれのない1本の共通原子鎖をマイニング



- 分類決定木C4.5, 回帰分類木M5[Kramer, 97],
ILP[King,96]
化学分子中の特徴的部分構造を予め人間が指定し,
記述子(属性)表・トランザクションとして付与

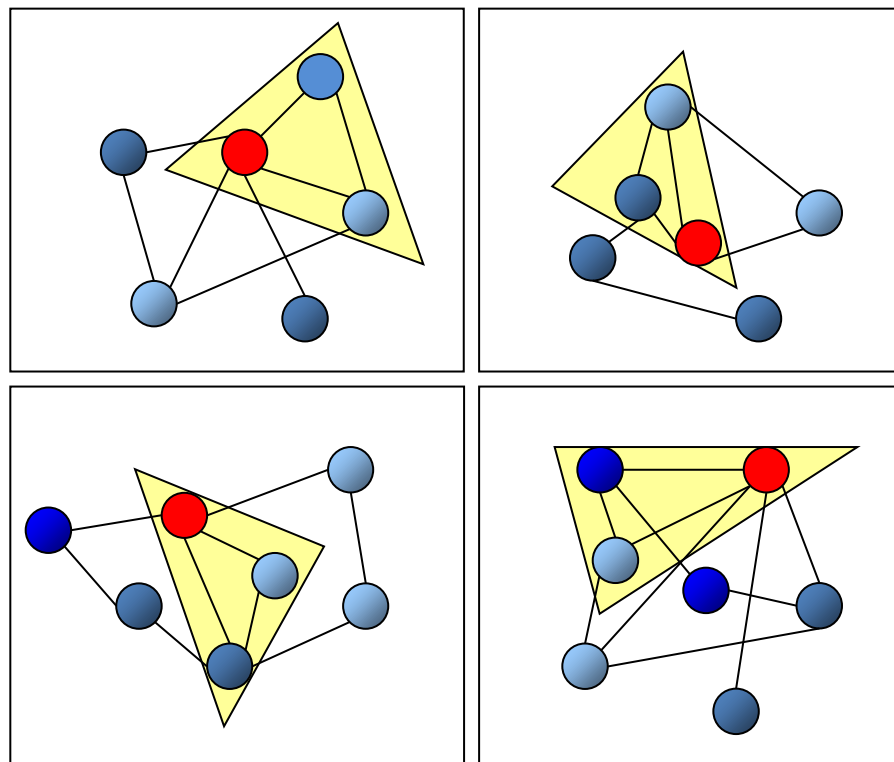
(1)ベンゼン環:2, NH₂:2,... (2)OH:3, O=:3,...

列挙による頻出グラフ構造マイニング 時代の幕開け

- Geedyな探索・列挙：不完だが全高速
 - SUBDUE [Cook 94], GBI[Yoshida&Motoda,95]
 - 頂点ペアチャンク反復によるGreedy探索
- 論理による探索・列挙：完全だが低速
 - WARMAR (Apriori+ILP)[Dehaspe,98]
 - Progol記述, 幅優先完全探索(2, 3ノードまで)
- **パターン列挙と計数による探索：完全で比較的高速**
 - AGM[Inokuchi et al., 99,00]
 - Apriori+グラフ接続行列の幅優先完全探索
(高速, 10から20ノードまで)

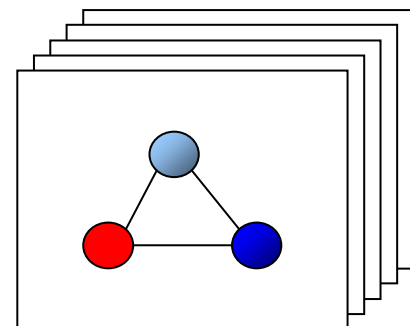
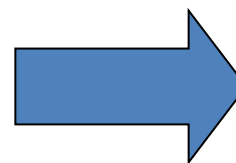
グラフカーネルなど数理的方法については今回割愛する。

パターン列挙と計数による 頻出部分グラフデータマイニング



入力

最小支持度(閾値)=2



出力(多頻度部分グラフ)

- 多頻度グラフの候補の数 (頂点, 辺, ラベルの組合せ)
- 部分グラフ同型問題 (NP-Complete)

定義

- 支持度 (support)

$$sup(G) = \frac{G \text{を部分グラフとして含むグラフデータの数}}{\text{データベースに含まれるグラフデータの数}}$$

- 多頻度部分グラフ (frequent subgraph)
 - ユーザが指定した最小支持度 (minsup: minimum support)を超える支持度を持つグラフ
- 多頻度部分グラフ抽出問題
 - グラフ構造データの集合と最小支持度が与えられたときに、部分グラフとして含まれる多頻度部分グラフを全て抽出する問題

パターン列挙と計数による グラフマイニング手法の歴史

Table of Enumeration Based Graph Miners

Name	Year	Function	History
AGM	2000	Enumerate all frequent induced subgraphs in a set of graphs.	
FSG(Pafi)	2001–2002	Enumerate all connected frequent subgraphs in a set of graphs.	
AcGM	2002	Enumerate all connected frequent subgraphs in a set of graphs, optionally restricted to induced subgraphs.	Extension of AGM
gFSG	2002	Enumerate all geometric frequent subgraphs in a set of graphs.	
gSpan	2002–2003	Enumerate all connected frequent subgraphs in a set of graphs.	
Frequent Path based GraphMiner	2002–2004	Enumerate all frequent induced subgraphs (labeled or unlabeled) in a set of graphs, where an embedding is counted only if it is 'edge disjoint' with other embeddings.	
CloseGraph	2003	Enumerate all connected frequent closed subgraphs in a set of graphs.	Extension of gSpan
FFSM	2003	Enumerate all connected frequent subgraphs in a set of graphs.	
FreeTreeMiner	2003–2004	Enumerate all free subtrees in a set of graphs that satisfy constraints specified by the user, optionally using version spaces if other constraints than minimum frequency constraints are used.	
ADI-Mine	2004	Enumerate all frequent induced subgraphs in a set of graphs.	Extension of gSpan for small graphs and large numbers of labels
DSPM	2004	Enumerate all connected frequent closed subgraphs in a set of graphs.	Extension of gSpan
SiGraM(Pafi)	2004	Enumerate all non-overlapping connected frequent subgraphs in one large graph	
Spin	2004	Enumerate all maximal connected frequent subgraphs in a set of graphs.	Extension of FFSM
Gaston	2004	Enumerate all connected frequent subgraphs in a set of graphs.	
Generalized (Biased) AGM	2004	Enumerate all connected frequent subgraphs in a set of graphs, where a hierarchy is available for the labels.	Extension of AcGM
Reference: http://hms.liacs.nl/graphs.html			

パターン列挙と計数による 頻出部分グラフデータマイニング

- 共通する原理

1. グラフデータ $D = \{G_i \mid i=1, \dots, n\}$ に頻出する大きさ (頂点数や辺数) $k=1$ の部分グラフを列挙

$$fg(k) = \{g(k) \mid \text{sup}(g(k)) \geq \text{minsup}\}$$

2. $fg(k)$ の各部分グラフ $g(k)$ の頻度 $\text{sup}(g(k))$ より、それを含む1つ大きな部分グラフ $(g(k) \subset) g(k+1)$ の頻度 $\text{sup}(g(k+1))$ は、小さいか等しい; $\text{sup}(g(k+1)) \leq \text{sup}(g(k))$ という頻度の逆単調性から、 $g(k) \subset g(k+1)$ を満たす大きさ $k+1$ の頻出部分グラフ候補 $g(k+1)$ を列挙
3. 各候補 $g(k+1)$ の D における $\text{sup}(g(k+1))$ の頻度を数え、実際に頻出な $g(k+1)$ の集合を得る。

$$fg(k+1) = \{g(k+1) \mid \text{sup}(g(k+1)) \geq \text{minsup}\}$$

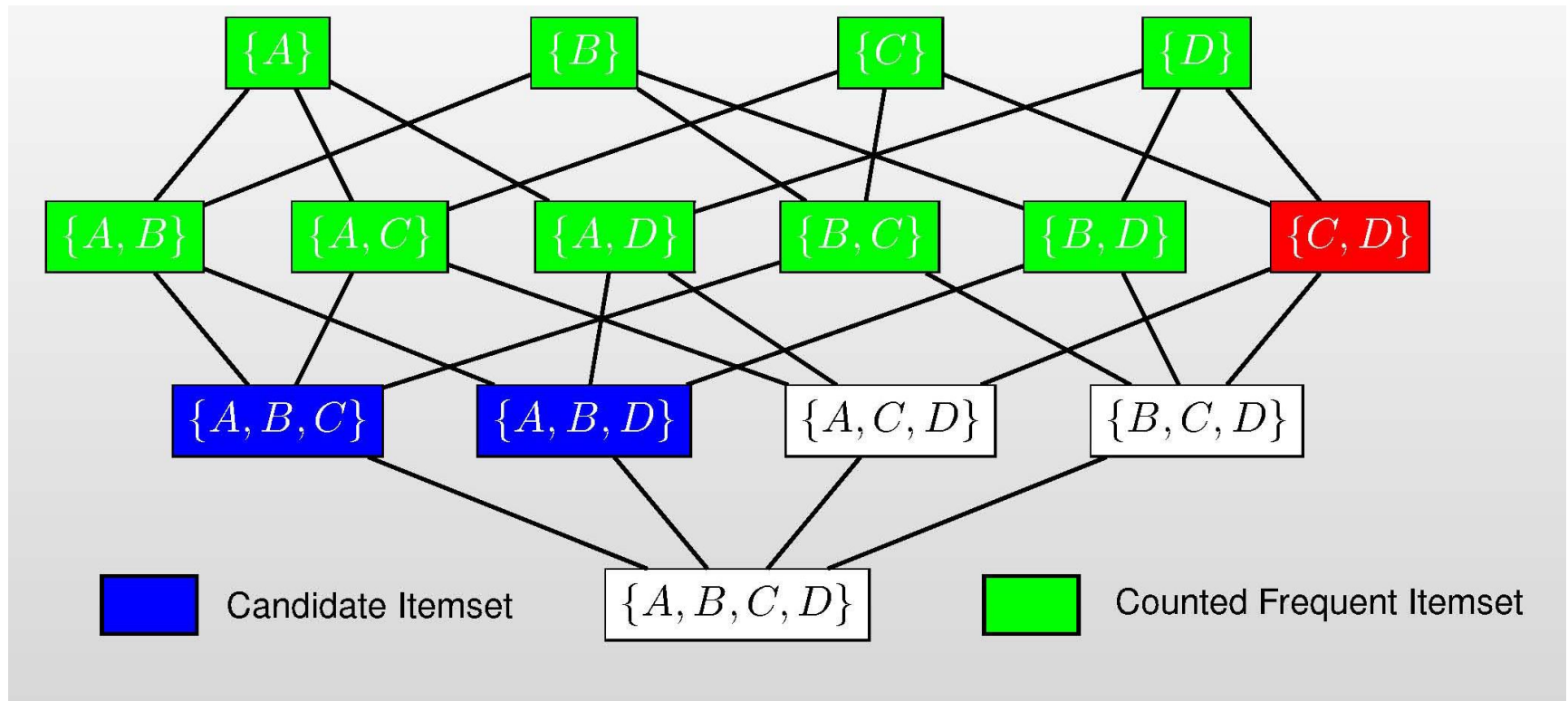
4. $k=k+1$, $fg(k)$ が空でないなら, ステップ2へ。
5. 頻出部分グラフ集合 $FG = \bigcup_{i=1, \dots, k} fg(i)$ を得る。

パターン列挙と計数による 頻出部分グラフデータマイニング

- 幅優先探索(BFS) V.S. 深さ優先探索(DFS)
 - 大きさ k の頻出部分グラフ $g(k)$ を含む1つ大きな頻出部分グラフ候補($g(k) \subset g(k+1)$)を列挙する際、幅優先探索(BFS)を行うものと深さ優先探索(DFS)を行うものがある。
 - メモリ使用効率や速さの点でDFSを使うものが多い。
- 部分グラフ列挙 V.S. スパニングツリー列挙と枝付加
 - 頻出部分グラフ $g(k)$ から頻出部分グラフ候補 $g(k+1)$ を生成する際、 $g(k)$ から直接 $g(k+1)$ を列挙するものと $g(k)$ のスパニングツリーから $g(k+1)$ のスパニングツリーを列挙し枝を追加するものがある。
- 複数グラフ V.S. 1枚グラフ
 - 複数グラフに亘るマイニングを行うもの(殆どのもの)と、1枚の大きなグラフのマイニングを行うもの(SiGraM(Pafi)など)

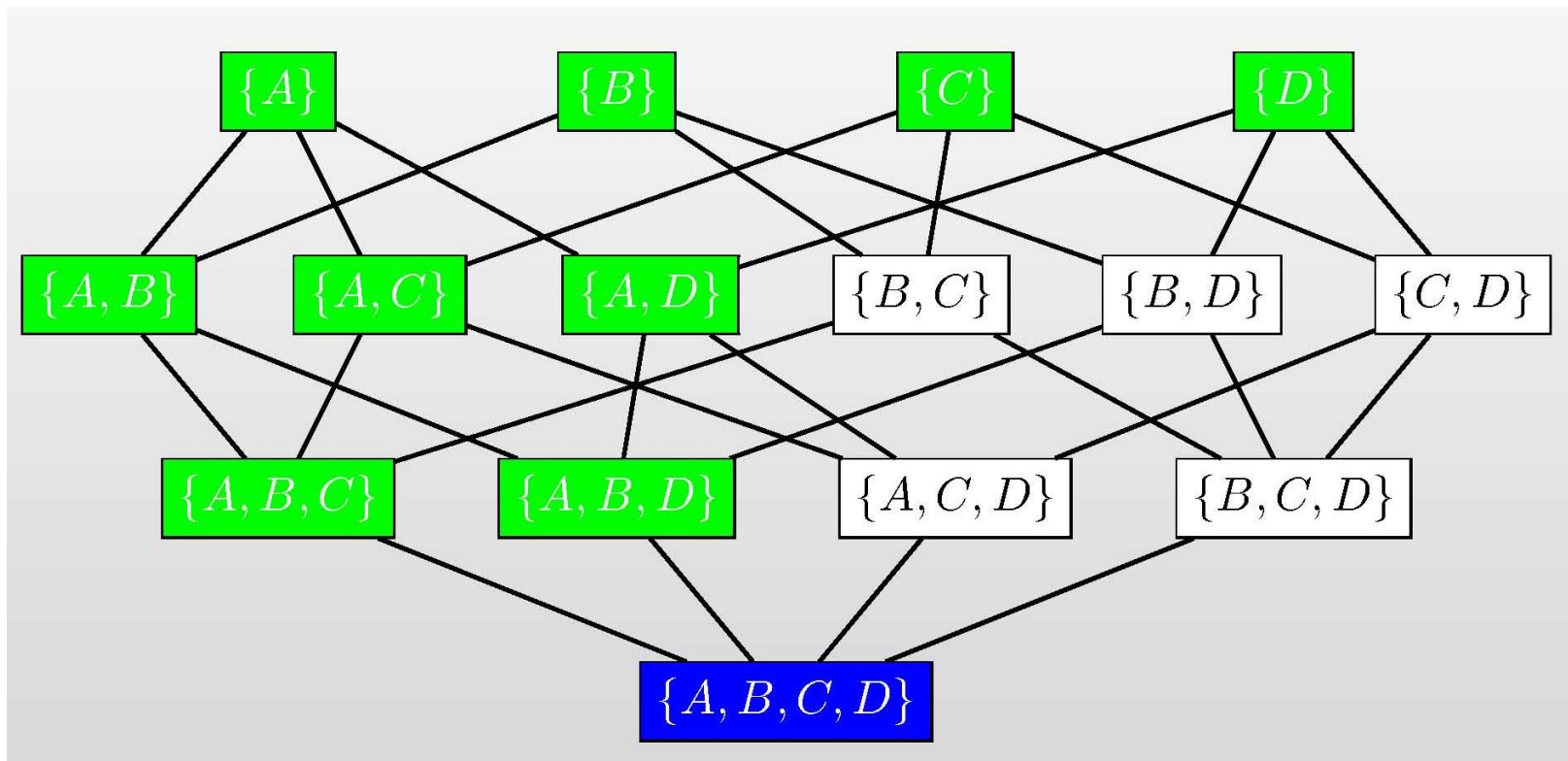
パターン列挙と計数による 頻出部分グラフデータマイニング

- 幅優先探索(BFS)方式: AGM, AcGMなど



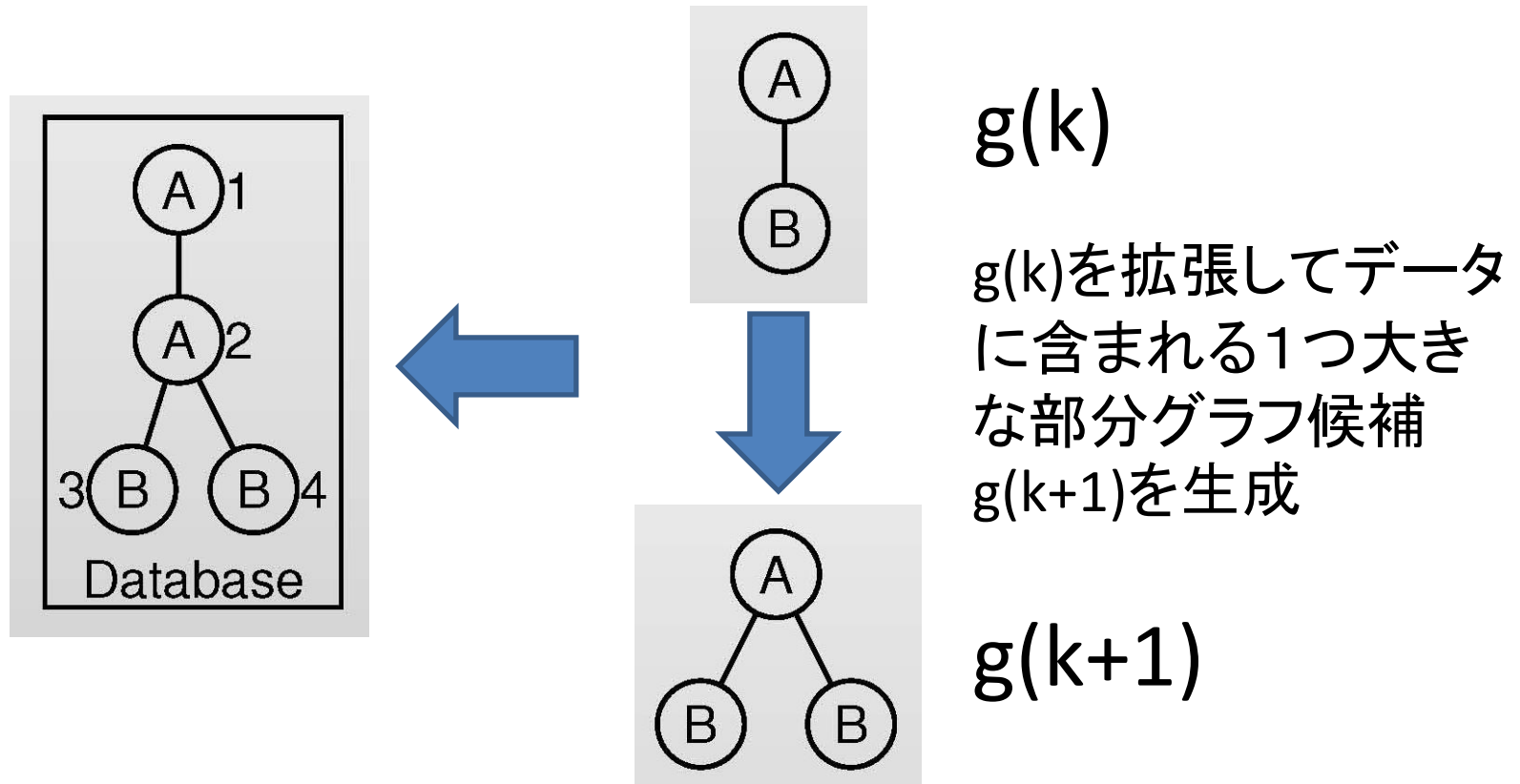
パターン列挙と計数による 頻出部分グラフデータマイニング

- 深さ優先探索(DFS)方式:gSpan,Gastonなど



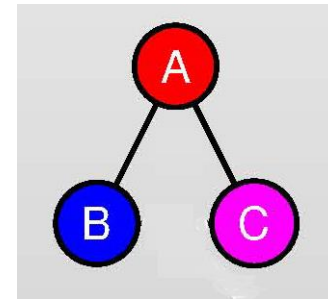
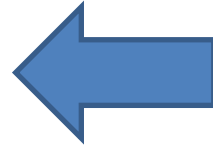
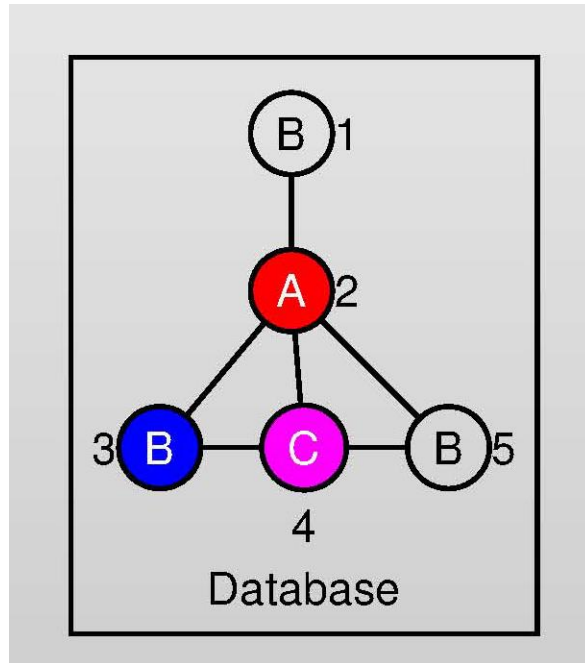
パターン列挙と計数による 頻出部分グラフデータマイニング

- 部分グラフ列挙方式: AGM, AcGM, gSpanなど



パターン列挙と計数による 頻出部分グラフデータマイニング

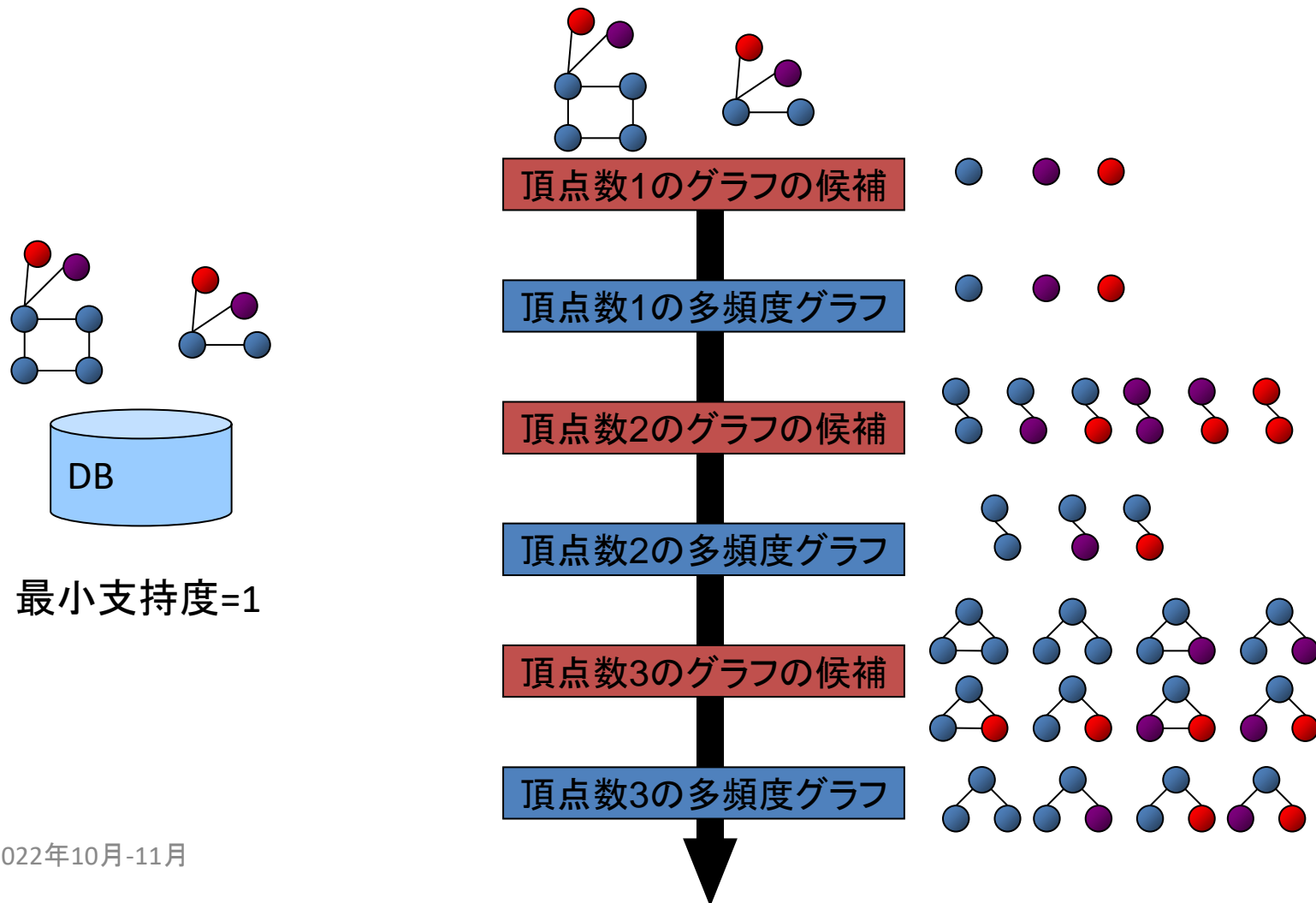
- スパニングツリー列挙と枝付加方式: GASTONなど



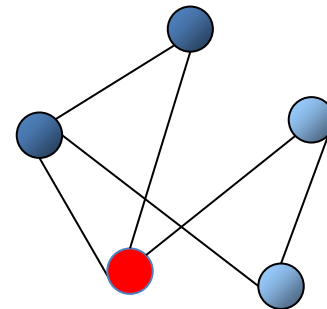
$g(k)$ の小さいスパニングツリーに
頂点1個とその間の辺を追加し
て、データに含まれる1つ大きな
スパニングツリーとそれを含む部
分グラフ候補 $g(k+1)$ を生成

AGM (Apriori-based Graph Mining) アルゴリズム

世界初のパターン列挙と計数による
頻出部分グラフ完全探索アルゴリズム
(幅優先探索(BFS)方式, 部分グラフ列挙方式)



定義 - カノニカルラベル -



- code

$$X_5 = \begin{matrix} & 1 & 1 & 2 & 3 & 3 \\ \begin{matrix} 1 \\ 1 \\ 2 \\ 3 \\ 3 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

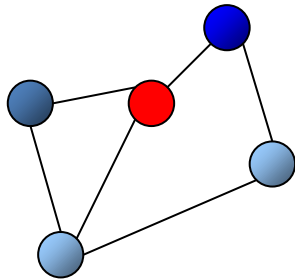
$$code(X_5) = \underline{1011100011}$$

- CODE

$$X_5 = \begin{matrix} & \underline{1} & \underline{1} & \underline{2} & \underline{3} & \underline{3} \\ \begin{matrix} 1 \\ 1 \\ 2 \\ 3 \\ 3 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

$$CODE(X_5) = \underline{11233} \underbrace{1011100011}_{code(X_5)}$$

定義 - 正準形 -



$$\begin{matrix} & G & G & R & B & B \\ \begin{matrix} G \\ G \\ R \\ B \\ B \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

$$\begin{matrix} & B & B & G & G & R \\ \begin{matrix} B \\ B \\ G \\ G \\ R \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

$$\begin{matrix} & B & R & G & G & B \\ \begin{matrix} B \\ R \\ G \\ G \\ B \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$\begin{matrix} & G & R & G & B & B \\ \begin{matrix} G \\ R \\ G \\ B \\ B \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

- 正準形 (Canonical Form)
 - 同型のグラフを表す隣接行列のなかでラベルコードが最大となる隣接行列

多頻度グラフの候補の生成

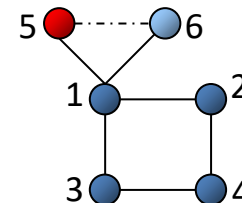
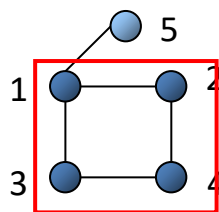
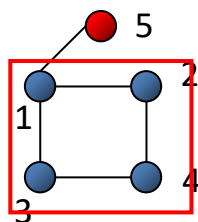
- 条件 1

G	G	G	G	R
0	1	1	0	1
1	0	0	1	0
1	0	0	1	0
0	1	1	0	0
1	0	0	0	0

G	G	G	G	B
0	1	1	0	1
1	0	0	1	0
1	0	0	1	0
0	1	1	0	0
1	0	0	0	0



G	G	G	G	R	B
0	1	1	0	1	1
1	0	0	1	0	0
1	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	*
1	0	0	0	*	0



- 条件 2

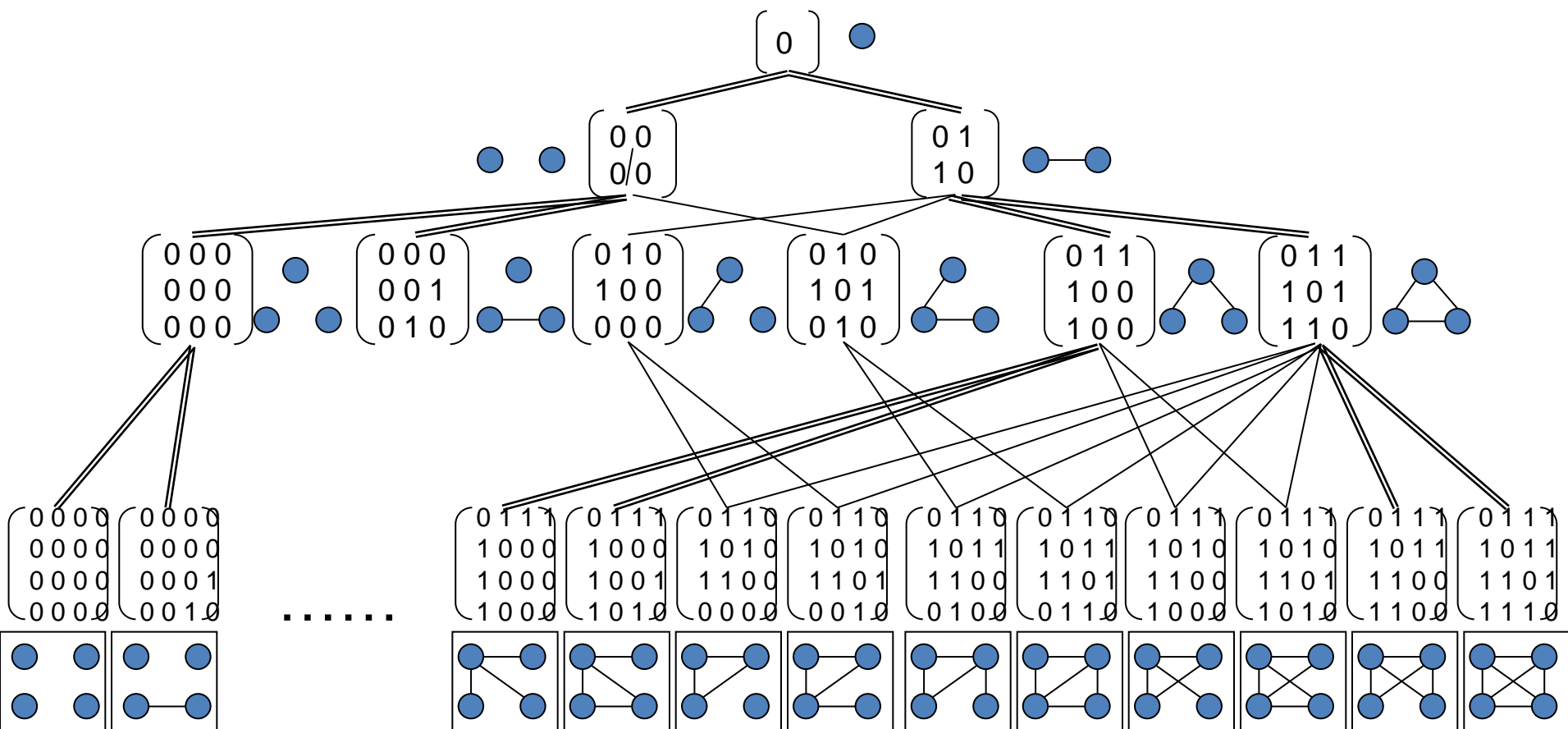
- 第1生成行列が正準形である

- 条件 3

- $\text{CODE}(X_k) \geq \text{CODE}(Y_k)$



AGMアルゴリズム(候補列挙)

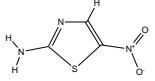
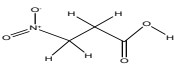


創薬における副作用候補分析支援システムへの応用

我々と化学者(関西学院大学:岡田教授)の共同研究成果

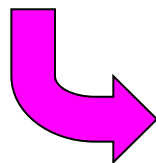
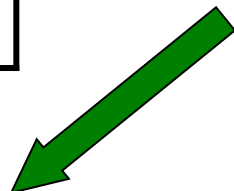
[Inokuchi et al. 2001]

治験薬構造活性データベース

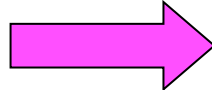
構造	生理活性
	活性
	不活性



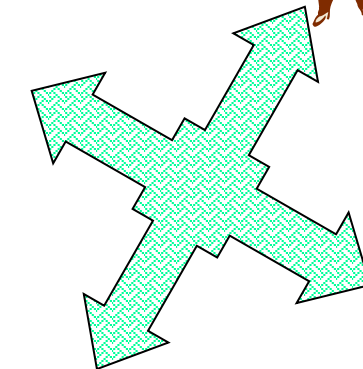
前処理



マイニング



活性プロファイル
知識ベース



予想外
副作用?

リスク警告



類似薬品
検索



データ
ベース

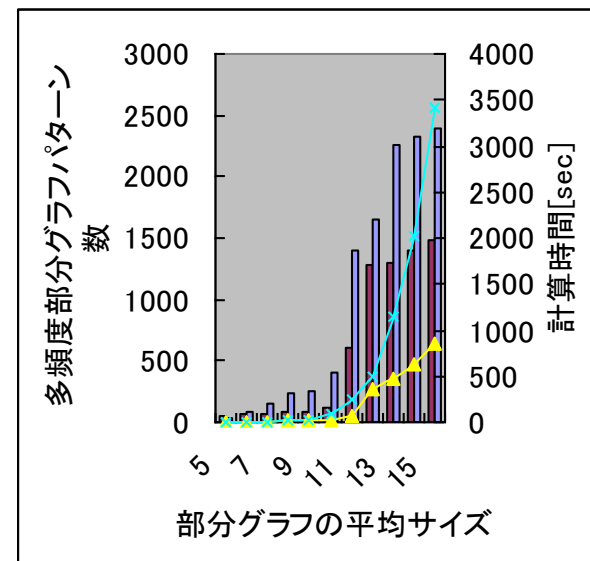
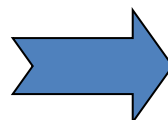
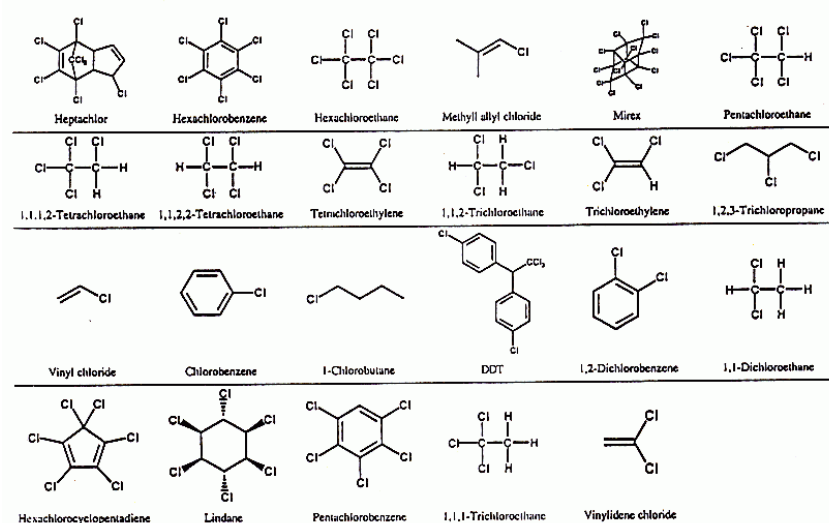


新薬品



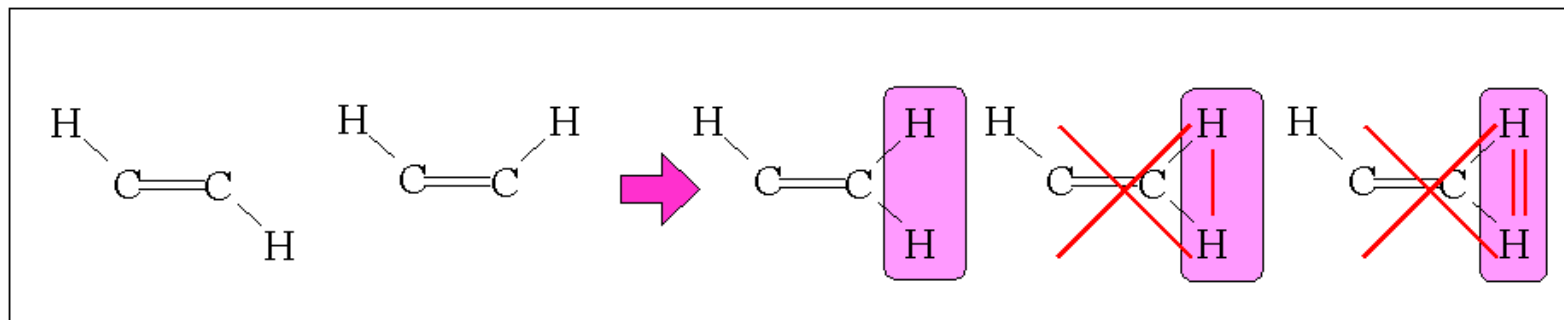
頻出部分グラフマイニング手法 AGM を適用

変異原性を有する化学物質構造の例
各分子構造を1枚のグラフとして扱い、変異原性の強い分子に頻出する部分分子構造を洗い出す。



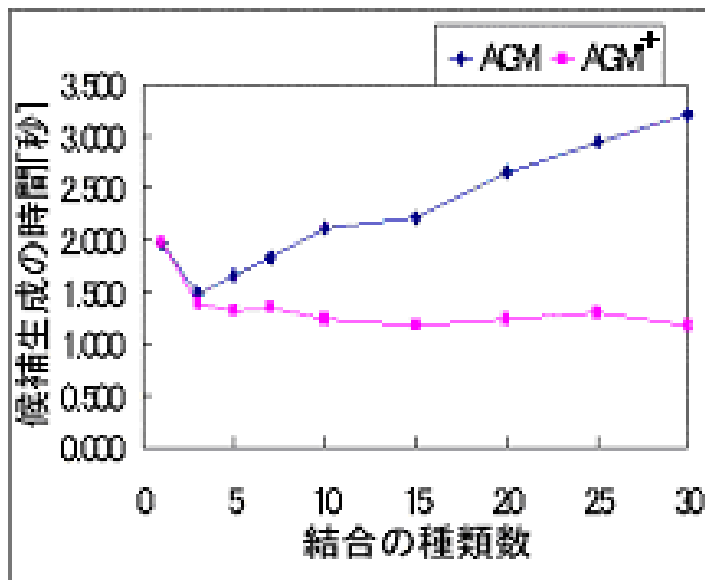
データの性質に合わせたAGMの高速化

化学分子固有のグラフ構造の性質を構造列挙の制約に用いる。

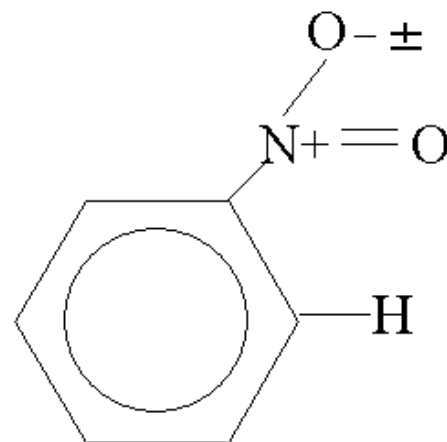


成果

結合種類と計算時間：AGM⁺は改良手法

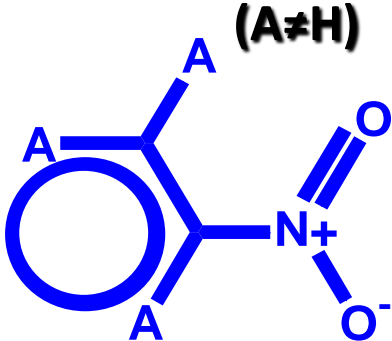
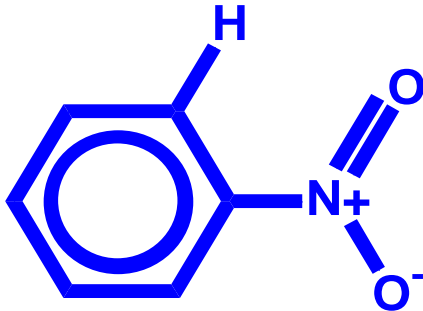


NO₂基とHが引き合いNO₂基がベンゼン環平面と同じになる。
RNA2重螺旋に入りこみやすくなり、遺伝子を傷つける要因となる。

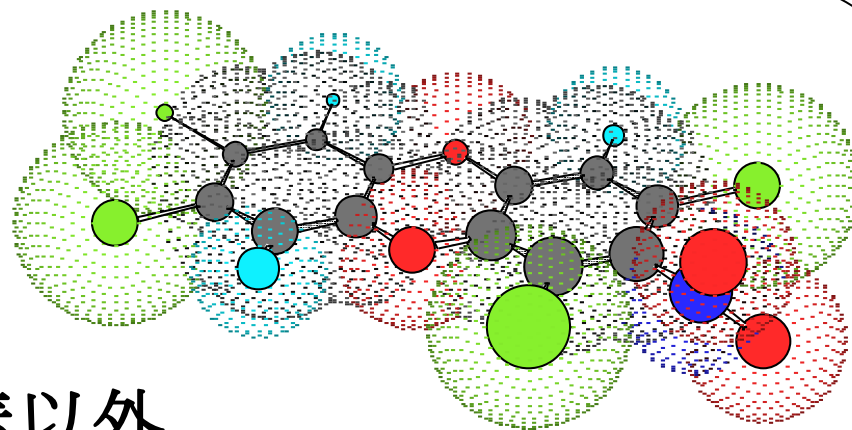
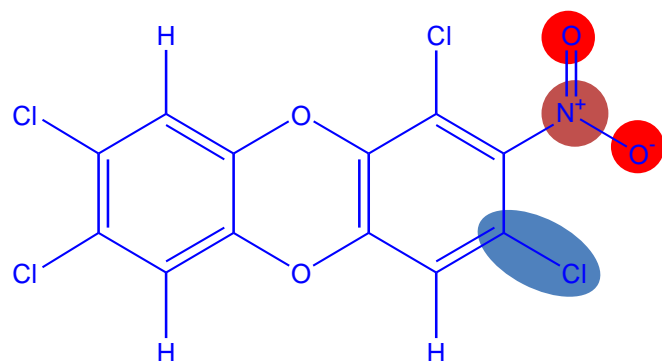


ニトロ芳香族化合物の変異原性

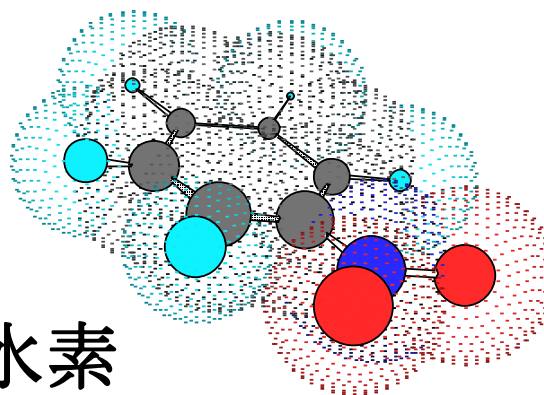
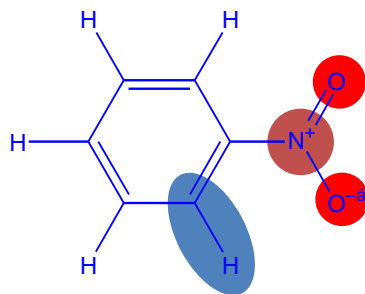
発見された部分構造

マイニング法	カスケードモデル	AGM
条件	高いLUMO AND $(A \neq H)$ 	高いLogP AND 
変異原性	低い	高い

ニトロ芳香族化合物の変異原性



水素以外



水素

**NO₂基とHが引き合いNO₂基がベンゼン環平面と同じになる。
RNA2重螺旋に入りこみやすくなり、遺伝子を傷つける要因となる。**

現状最速のパターン列挙・計数型の頻出 グラフマイニングアルゴリズムの1つ

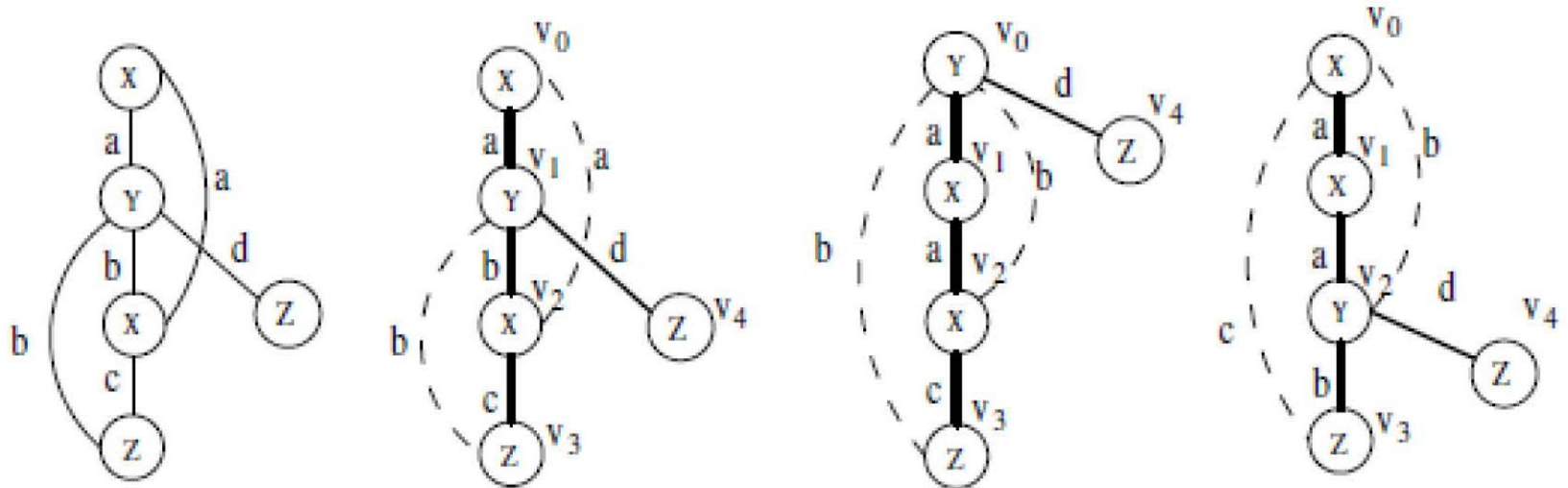
- gSpan [Yan and Han 2002]
 - グラフを直接探して同型性を確認するのではなく、カノニカルDFSコードを生成して比較する。
 - 各グラフは一意的なカノニカルDFSコードを持つので、コードの照合でグラフが同型か否か判る。
 - カノニカルDFSコードは深さ優先探索(DFS)木を基に定義される。
 - 探索木と一体なコードを定義することによって、アルゴリズム上効率的に照合して同型性を判定すると共に、無駄な探索を避ける。

gSpan

- DFS 木
 - 木のノードが探索中の部分グラフの頂点に対応
 - 探索途中で訪問順にグラフ頂点番号を振る
 - $v_i < v_j$ if v_i is traversed before v_j
- DFS木のルートからノードまでのパスが、部分グラフ上の順序づけられたパスに対応する。
- DFS木は部分グラフの各辺を2つのグループに分ける。
 - 前向き辺集合: (v_i, v_j) where $v_i < v_j$
 - 後ろ向き辺集合: (v_i, v_j) where $v_i > v_j$
- 部分グラフ上の辺のなぞり方は多数あるので、このままではDFS木も膨大にあり得る。

gSpan

- あるグラフの異なるなぞり方

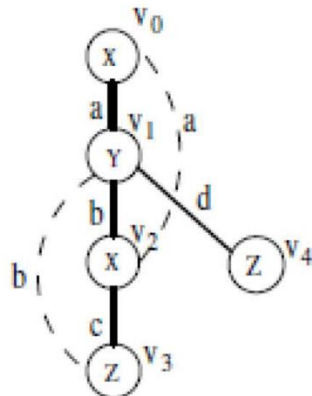


実線: 前向き辺集合

破線: 後ろ向き辺集合

gSpan

- なぞり方を一意にするため辺に全順序を導入
 - 1. $(u, v) <_T (u, w)$ if $v < w$
 - 2. $(u, v) <_T (v, w)$ if $u < v$
 - 3. $e_1 <_T e_2$ and $e_2 <_T e_3$ implies $e_1 <_T e_3$
- この規則に従う辺の全順序付けをDFSコードという。これによってなぞり方が減る。



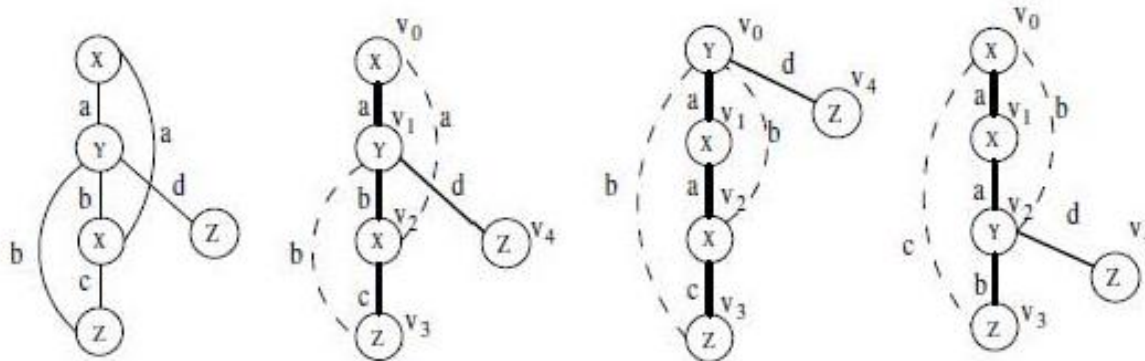
$$\{ (v_0, v_1), (v_1, v_2), (v_2, v_0), (v_2, v_3), (v_3, v_1), (v_1, v_4) \}$$

gSpan

- 1つの辺を表すDFSコードは4つのタプルからなる。
 - 辺(起点と終点の頂点番号の組み)
 - 起点頂点のラベル
 - 辺のラベル
 - 終点頂点のラベル
- 頂点や辺のラベルに辞書順を導入する。
- 以上により、4つのタプルからなるDFSコードにも辞書順を定義する。

gSpan

- DFSコードの例



edge	α	β	γ
0	$(0, 1, X, a, Y)$	$(0, 1, Y, a, X)$	$(0, 1, X, a, X)$
1	$(1, 2, Y, b, X)$	$(1, 2, X, a, X)$	$(1, 2, X, a, Y)$
2	$(2, 0, X, a, X)$	$(2, 0, X, b, Y)$	$(2, 0, Y, b, X)$
3	$(2, 3, X, c, Z)$	$(2, 3, X, c, Z)$	$(2, 3, Y, b, Z)$
4	$(3, 1, Z, b, Y)$	$(3, 0, Z, b, Y)$	$(3, 0, Z, c, X)$
5	$(1, 4, Y, d, Z)$	$(0, 4, Y, d, Z)$	$(2, 4, Y, d, Z)$

gSpan

- あるグラフ G の辞書順で最小のDFSコードを最小DFSコード $\min(G)$ という。
- 定理 任意2つのグラフ G と H は、 $\min(G)=\min(H)$ の時、かつその時に限り同型である。
(最小DFSコードはカノニカルである)
- 頻出部分グラフマイニング問題
 - 頻出部分グラフ問題はそれらに対応する最小DFSコードをマイニングすることと等価である。
 - これはパターン列挙・探索アルゴリズムで逐次処理可能である。

gSpan

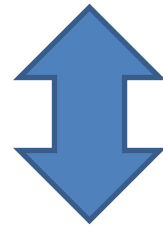
- DFSコード木を定義して用いる。
 - DFS コードの親と子
$$\alpha = (a_0, a_1, \dots, a_m)$$
$$\beta = (a_0, a_1, \dots, a_m, \mathbf{b})$$
 - code's parent and child
 - α は β の親であり、 β は α の子である。
- DFS コード木では
 - 各ノードがルートからのDFSコードを表す。
 - 親子間の関係は上記で定義した関係を表す。
 - 兄弟はDFSの辞書順の関係に従う。

gSpan

- DFSコード木の性質
 - グラフの頂点、辺ラベル集合 L が与えられたとき、DFSコード木はその L についてすべての可能なグラフを表すことができる。
 - DFSコード木において、第 n レベルのノードに対応する各部分グラフは $n-1$ 個の辺からなる。
 - DFSコード木はそのノードが表すすべての部分グラフの最小DFSコードを含む。

gSpan

- 頻度の逆単調性より、ある部分グラフ g_1 が多頻度ならば、その部分グラフ $g_2(\subset g_1)$ は多頻度である。



- もしDFSコード α が多頻度なら、 α のすべての祖先も多頻度である。
- もし α が多頻度でなければ、 α のすべての子孫も多頻度ではない。

gSpan

- DSFコード木の優れた性質
 - DFSコード木の中で、いくつかの部分グラフは複数のDFSコードノードに対応する。
 - DFSコード木を辞書順になぞった時、ある部分グラフについて最初に現れたコードが最小DFSコードである。
- 定理 DFSコード木において、もしノードのDFSコードが最小でない場合、そのノードより下を枝刈りしても、取りこぼしなくすべての部分グラフが保持される。
 - DFSコード木を単純に辞書順に探索することで、すべての頻出部分グラフ候補を列挙できる。

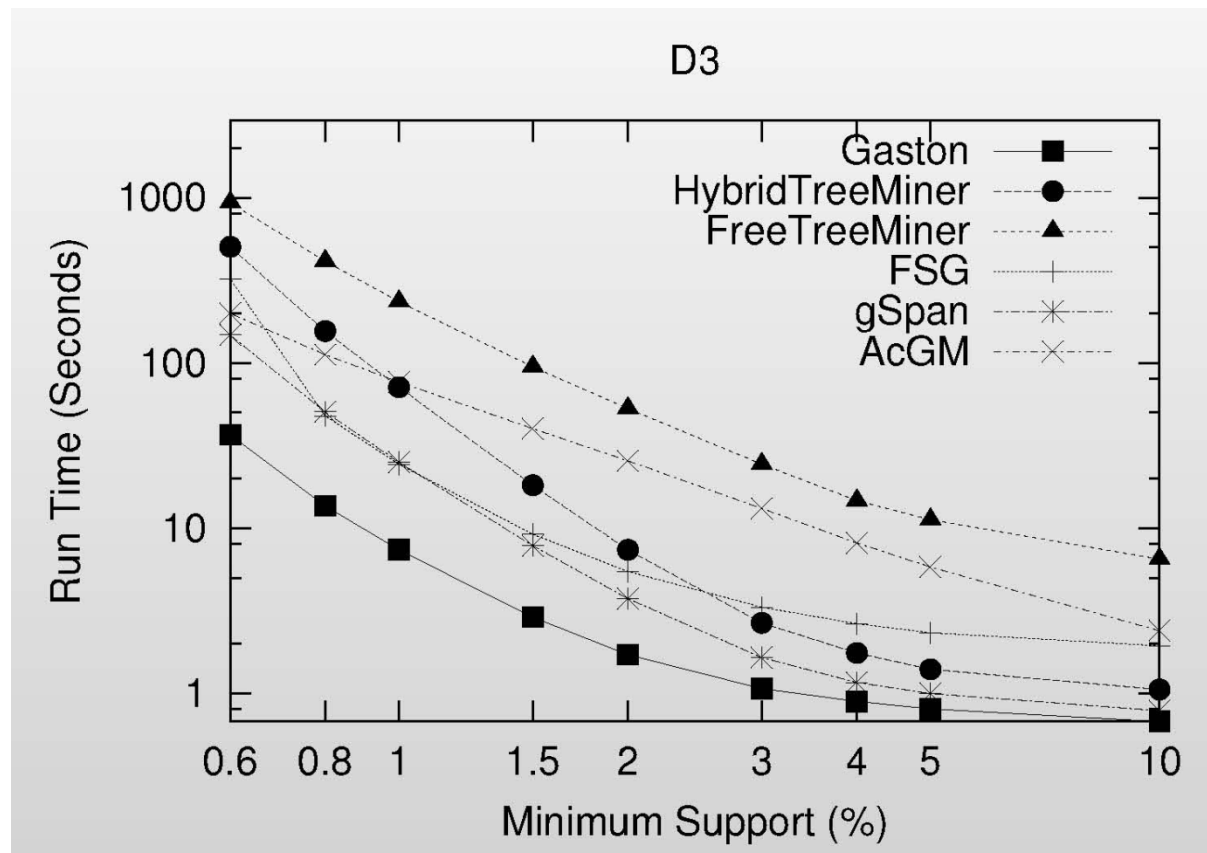
gSpan

- 前述のDFSコード木の性質を使うことで
 - $(k+1)$ -辺を持つ多頻出部分グラフを、拡張により既知の k -辺の頻出部分グラフから生成できる。
 - この時、DFSコードに性質により、深さ優先探索で拡張しても取りこぼしは起きない。
- 無駄なバックトラックが発生せず、メモリー空間も節約でき、非常に効率的である。

現状最速のパターン列挙・計数型の 頻出グラフマイニングアルゴリズムは？

- ベンチマーク比較 [Nijssen 2004]

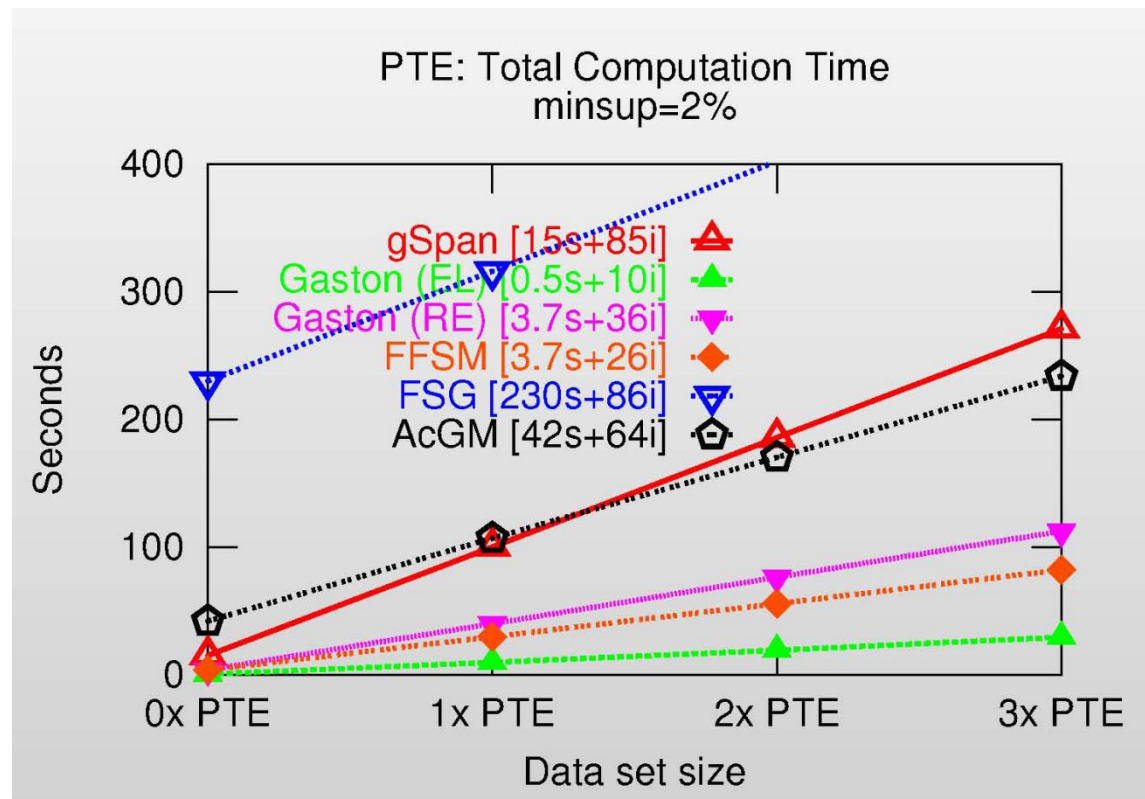
自由木(根なし木)データセットのマイニング結果



現状最速のパターン列挙・計数型の 頻出グラフマイニングアルゴリズムは？

- ベンチマーク比較 [Nijssen 2004]

分子構造データセットのマイニング結果

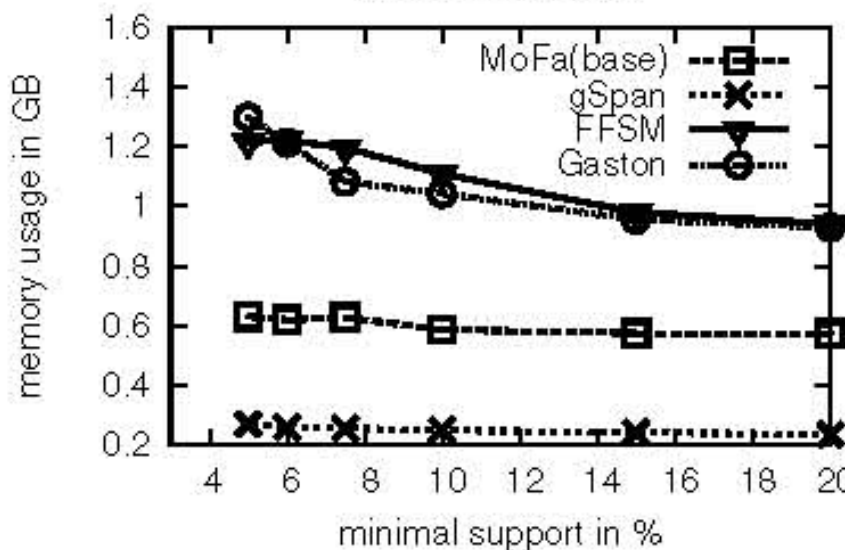


現状最速のパターン列挙・計数型の 頻出グラフマイニングアルゴリズムは？

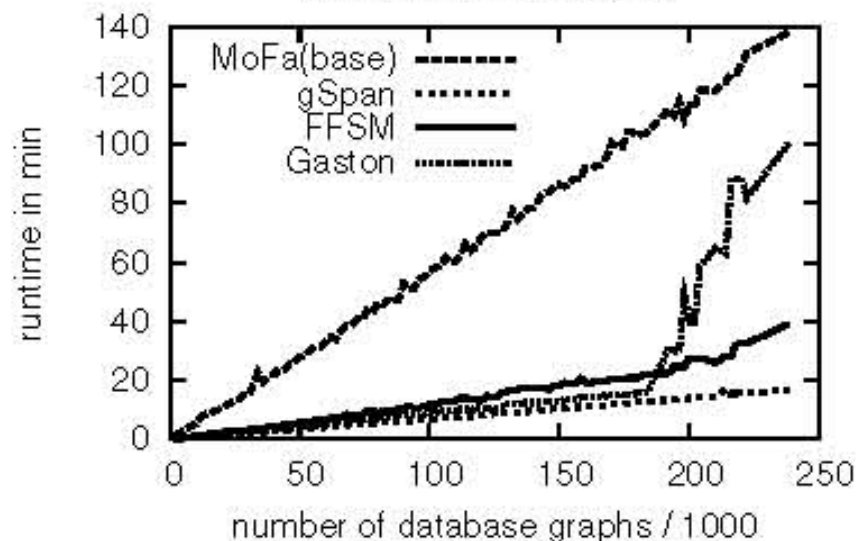
- ベンチマーク比較 [Worlein et al. 2005]

HIVデータ 分子構造グラフ42639個,
最大頂点数234, 平均頂点数27, 頂点ラベル数58

メモリ消費量
HIV(42689 graphs)



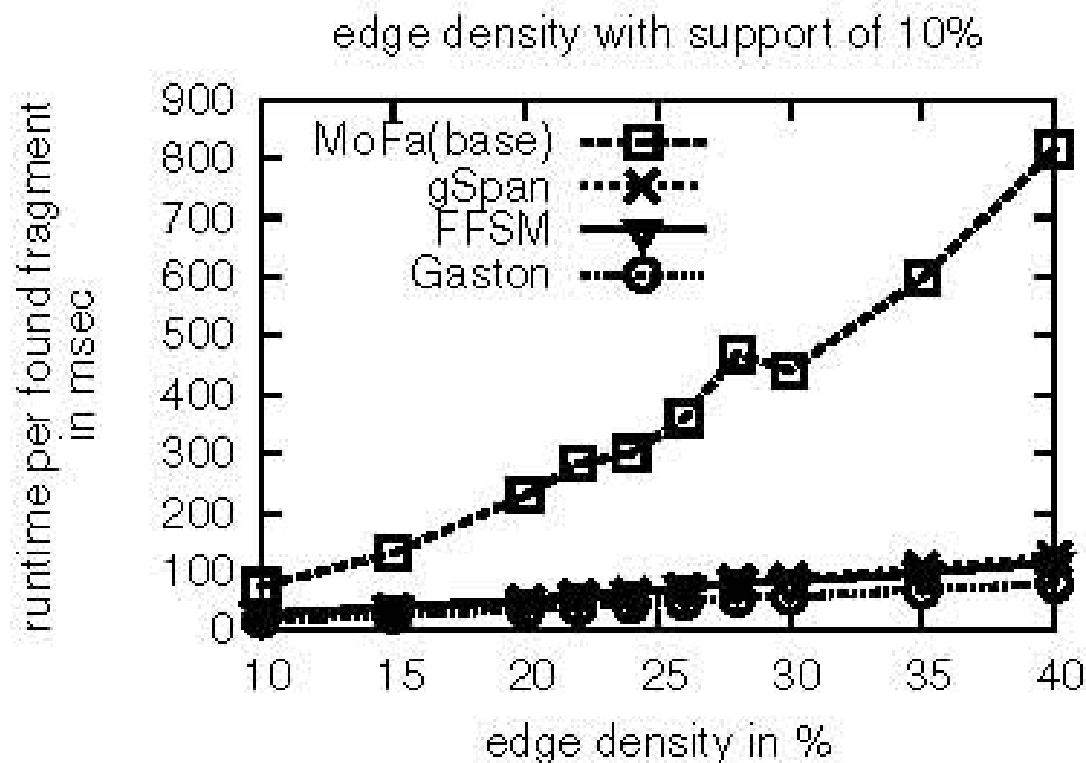
計算時間
NCI(partial; 5% support)



現状最速のパターン列挙・計数型の 頻出グラフマイニングアルゴリズムは？(つづき)

- ベンチマーク比較 [Worlein et al. 2005]

各種辺密度の人工データに関する計算時間



まとめ

- グラフデータに頻出する部分グラフを探索するグラフマイニングの枠組みと原理、比較、応用例について話をした。
- 多数グラフからなるデータセットから頻出部分グラフを完全探索する最速アルゴリズムは、現状では深さ優先探索 (DFS) を用いるgSpanかGASTONである。
- 化学物質分子の部分構造と物性 (毒性) の相関解析への適用例を示した。