DEIM Forum 2023 2b-4-4

グラフデータベースに対する高速高精度な相関問合せ

直井　悠馬†　　真次　彰平†　　塩川　浩昭††

† 筑波大学 理工情報生命学術院　〒 305-8577 茨城県つくば市天王台 1-1-1
†† 筑波大学計算科学研究センター　〒 305-8577 茨城県つくば市天王台 1-1-1
E-mail: †{naoi,matsugu}@kde.cs.tsukuba.ac.jp, ††shiokawa@cs.tsukuba.ac.jp

あらまし　相関問合せとはグラフデータベースの中からクエリと共起して出現する部分グラフを検出する処理であり，医療や生命科学分野で広く利用されている．これまでグラフデータベースの中からクエリグラフとの相関値が最も高い k 個の部分グラフを求める Top-k 相関グラフ問合せ手法が提案されている．しかし，グラフデータベースを構成するグラフがより大規模になると相関判定のコストは大きくなるため，依然として実行に多くの時間がかかる．そこで本研究ではグラフ構造を要約したグラフを構築することで相関問合せのコストを削減する手法を提案する．本研究では，高精度な相関問合せを行うため乱択アルゴリズムを利用し，高精度を保証するための反復回数の理論的な証明を行った．実データを用いた評価実験を行い，提案手法が既存手法に比べて最大 19.47 倍高速であるとともに，高精度に Top-k 相関グラフを得られることを確認した．

キーワード　グラフデータベース，データ構造・索引，問合せ処理，相関問合せ

1 序　　論

グラフはオブジェクト間の関係を表現する際に利用される一般的なデータ形式であり，科学や生物学，生命科学分野，ソーシャルネットワークなどの分野で広く使われている [18, 19]. 複数のグラフから構成されるグラフデータベースの需要が高まると共に，これまでデータベースの中から頻出した部分グラフ構造を検出する頻出部分グラフ検出の研究 [3, 4, 10, 11] が広く行われてきた．その中でもデータベースの中から共起して出現する部分グラフ対を検出する相関部分グラフ検出の研究 [1, 2, 6, 7, 21] は近年注目を集めている．

相関問合せはグラフデータベースとクエリグラフが与えられたとき，クエリグラフの出現分布に近い部分グラフを特定する問題である．共起して出現する部分グラフ対はデータベースの隠れた特性を示すことがあり，多くのアプリケーションで有用である．例えば，科学化合物のデータベースでは共起して出現する構造対を検出し，新たな知識発見の一助となる．

相関問合せの手法には，複数のグラフから構成されるグラフデータベースに対する相関グラフ問合せ手法 [1, 2, 6] や，単一の大規模なグラフに対する相関部分グラフ問合せ手法 [7] が研究されてきた．また，近年ではグラフデータベースから「興味のある」構造を検出する手法 [21] が提案されている．この手法では，頻出部分グラフ検出 [3] により頻出構造を検出した後に，頻出構造を分割することで相関ルール集合を得る．生成した相関ルールに対して，ユーザが適切に決めた相関閾値を超える相関ルールを全て検出する．これらの手法によってグラフデータベースに対して相関問合せを行うことで，共起して出現する構造対や興味深い構造対を検出することができる．

しかし，大規模なグラフデータベースに対する相関グラフ問合せにはいくつかの課題がある．1 つの課題はグラフデータベース内の全てのグラフの部分グラフが相関グラフの候補，すなわち候補グラフとなる点である．もう 1 つの課題は相関グラフ判定には候補グラフの出現頻度を考えるため膨大なコストがかかるという点である．グラフデータベースにおいて，候補グラフの出現頻度を高速に計算するためにグラフ索引技術 [8, 20] などがこれまで提案されているが，これらの手法を用いてもコストは依然として大きい．これらの問題に対して，これまでユーザによって与えられた閾値より大きな相関値を持つ部分グラフを検出する手法 [2, 5] や，相関値が最も高い k 個の部分グラフを検出する Top-k 相関問合せの手法 [1] が提案されている．この手法は探索する部分グラフ数を削減し，効率的に相関計算を行うことで相関グラフを検出することができる．しかし，大規模なグラフデータベースでは候補グラフの数が大きくなり，相関値の計算のコストが大きくなるため，相関問合せに実行時間が大きくなるという問題がある．

この問題を解決するために，本研究では大規模なグラフデータベースに対する高速な Top-k 相関グラフ問合せ手法を提案する．提案手法はグラフデータベースの各グラフ構造を要約したグラフデータベースを構築することで探索するグラフ数を縮小させると共に，候補となる部分グラフの相関問合せの判定コストの削減を図る．また，提案手法ではグラフ要約によって生じる相関グラフの偽陽性・偽陰性に対応するため，乱択アルゴリズムに基づくランダムな要約グラフを利用する．提案手法では，高精度な相関問合せを実現するために乱択アルゴリズムの反復回数を理論的に導出する．これにより，獲得した相関部分グラフに対して検証を行うことで高速高精度な Top-k 相関グラフ問合せ手法を提案する．

本研究では，提案手法は実際のタンパク質グラフデータベースに対して Top-k 相関問合せ処理が既存研究に比べて最大 19.47 倍高速であることを確認した．また，提案手法は高精度に Top-k 相関グラフを求めることができ，パラメータ k，グラフデータ

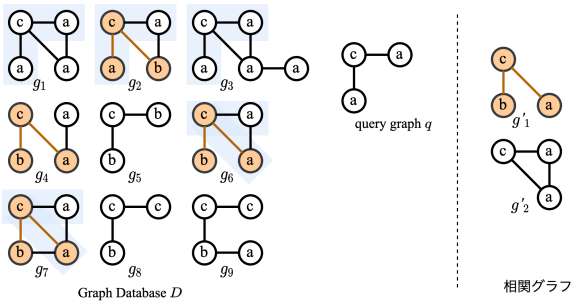


図 1: Top-k 相関問合せの例

ベースのサイズを変えた場合にも，既存手法に比べて高速であることを確認した．

本論文の構成は以下の通りである．2 節で前提知識と本研究で取り扱う問題について説明する．3 節で提案手法について述べ，4 節で実データを用いた性能評価結果を示す．5 節でケーススタディを紹介し，最後に 6 節で本研究のまとめを述べる．

2 事前準備

本研究で対象とするグラフはラベル付無向連結グラフ $g = (V, E, l)$ である．V は頂点集合，E は辺集合，l は各頂点と各辺にラベルを与えるラベル関数である．また N 個のグラフ $g_1, g_2, \ldots, g_N$ から構成されるグラフデータベースを $\mathcal{D} = \{g_1, g_2, ..., g_N\}$ とする．2 つのグラフを $g = (V, E, l)$, $g' = (V', E', l')$ について g は g' の部分グラフ同型であるとき，$g \subseteq g'$ と表記する．また，本論文では $g \subseteq g'$ となるとき，g' は g のスーパーグラフと呼ぶ．

グラフ g の $\mathcal{D}$ での出現頻度の指標として支持度を $supp(g; \mathcal{D}) = \frac{|\mathcal{D}_g|}{|\mathcal{D}|}$ と定める．本論文では $\mathcal{D}$ が文脈上明らかであるときは，$supp(g; \mathcal{D})$ を $supp(g)$ と表記する．2 つのグラフ g_1, g_2 が同時に出現する割合である結合支持度は $supp(g_1, g_2) = \frac{|\mathcal{D}_{g_1} \cap \mathcal{D}_{g_2}|}{|\mathcal{D}|}$ と定める．支持度には逆単調性があり，$g \subseteq g'$ ならば，$supp(g') \leqq supp(g)$ となる．

相関グラフ問合せとは $\mathcal{D}$ とクエリグラフ $q = (V_q, E_q, l_q)$ が与えられたとき，クエリと共起して出現する部分グラフを $\mathcal{D}$ から検出することである．本研究では 2 つのグラフの相関尺度としてピアソン相関係数 [9] を採用し，定義は以下の通りである．

定義 1 (ピアソン相関係数)**.** 2 つのグラフ g_1, g_2 が与えられたとき，それらの相関値 $\phi(g_1, g_2)$ を以下のように定める．

$$\phi(g_1, g_2) = \frac{supp(g_1, g_2) - supp(g_1)supp(g_2)}{\sqrt{supp(g_1)supp(g_2)(1 - supp(g_1))(1 - supp(g_2))}}$$

相関値 $\phi(g_1, g_2)$ は $[-1, 1]$ の範囲の値をとる．また，$\phi(g_1, g_2)$ が正の値をとるとき，2 つのグラフ対はグラフデータベースの中で共起して出現すると言える．本研究で対象とする Top-k 相関グラフ問合せを以下のように定義する．

定義 2 (Top-k 相関グラフ問合せ)**.** グラフデータベース $\mathcal{D} = \{g_1, g_2, ..., g_N\}$，クエリグラフ q，整数 k が与えられたとき，相関値 $\phi(g, q)$ が最も高い k 個のグラフ g を求める．

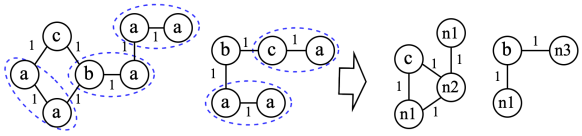


図 2: グラフ要約の例

Top-k 相関グラフ問合せの例として，図 1 では，グラフデータベース $\mathcal{D}$ ととクエリグラフ q に対して， g'_1, g'_2 のようなクエリグラフとの相関値が高いグラフを k 個検出する．

3 提案手法

本節では提案手法について説明する．本研究の目的は，大規模なグラフデータベースに対して高速に Top-k 相関グラフ問合せを行うことである．提案手法ではグラフ要約と乱択アルゴリズムに基づく相関グラフ問合せ手法を提案する．

提案手法の基本アイデアは，データベース内の各グラフを要約することによってグラフサイズを縮小し相関判定のコストを下げることである．グラフ要約とは図 2 のように 2 つの頂点を 1 つの頂点に集約する操作を繰り返し行うことでグラフの頂点数と辺数を小さくすることである．要約されたグラフから構成されるグラフ集合を要約データベースとすると，要約データベース内のグラフはグラフサイズが小さいため，相関グラフ探索の際に生成する候補グラフ数が小さくなる．また，候補グラフの支持度の計算コストが小さくなるため，要約データベースでの Top-k 相関グラフ問合せは高速に行うことができる．しかし，グラフ要約は Top-k 相関グラフ問合せに対して偽陽性・偽陰性となる結果をもたらす場合があるため，提案手法では乱択アルゴリズム [12] を利用することにより確率的に偽陽性・偽陰性となる結果を除外し，高速で高精度な Top-k 相関グラフ問合せを実現する．提案手法は以下の 3 つで構成される．

(1) グラフ要約： グラフデータベース $\mathcal{D}$ の各グラフに対してランダムに辺を集約した要約グラフデータベース $\mathcal{D}'$ を構築する．

(2) 探索： $\mathcal{D}'$ に対して Top-k 相関グラフ検出を行う．グラフ要約は Top-k 相関グラフ問合せに対して偽陽性・偽陰性となる結果をもたらす場合があるため，複数の要約グラフデータベースにおける TopCor の結果である k 個の要約された部分グラフからなる部分グラフ集合 T を複数生成する．

(3) 検証： 複数の T をグラフデータベース $\mathcal{D}$ と照合しグラフ復元を行い，正確な Top-k 相関グラフを求める．

提案手法の概要を図 3 に示す．グラフデータベース $\mathcal{D}$ が与えられ，事前処理としてデータベースの各グラフについてグラフ要約を行い，i 個の要約データベースを構築する．問合せ処理は，与えられたクエリグラフと要約データベース $D'_1, D'_2, \ldots, D'_i$ に TopCor を適用することで，相関グラフの探索を行う．次に探索で得られた相関グラフ集合 $T_1, T_2, \ldots, T_i$ について検証を行い，Top-k 相関グラフ T を得る．最後にグラフ要約，探索，検証をまとめたアルゴリズムを Algorithm 1 に示す．

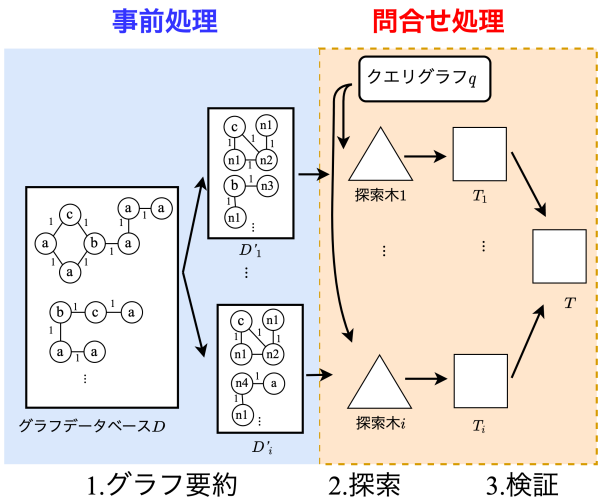


図 3: 提案手法概要

Algorithm 1 提案手法

Input: Graph database $\mathcal{D}$, a query q, and an integer k.
Output: The Top-k correlative graphs with respect to q.

```
Initialize an empty queue, Q_cur , of size k;
for j = 1 to i do
    (D'_j,q'_j)=SUMMARIZE(D, q);
for j = 1 to i do
    T_j = TopCor(D'_j, q'_j);
VELIDATE(D'_1, D'_2, ..., D'_i, T_1, T_2, ..., T_i);
Output the graphs in Q_cur;
```

3.1 グラフ要約

グラフ要約は $\mathcal{D} = \{g_1, g_2, ..., g_N\}$ について $\mathcal{D}$ 内の各グラフを要約したグラフデータベース $\mathcal{D}' = \{g'_1, g'_2, ..., g'_N\}$ を構築する．$\mathcal{D}$ が与えられたとき，グラフ要約はエッジ集合から $\langle l(u), l(u,v)l(v)\rangle$ をラベル対として，ラベル対集合 R_j をサンプリングする．ラベル対集合 R_j を以下の定義 3 に示す．

定義 3 (ラベル対集合)**.** $\mathcal{D} = \{g_1, g_2, ..., g_N\}$ が与えられたとき，ラベル対集合を R_j とし，以下のように定める．

$$R_j = R_{j-1} \cup \{\langle l(u), l(u,v), l(v)\rangle | \sigma(E^{i-1}) = (u,v)\}$$

ただし，$R_0 = \emptyset$ であり，$\sigma(E^{i-1})$ は以下に定義するエッジ集合 E^{i-1} からランダムに 1 つのエッジを選択する関数である．

$$E^{i-1} = \bigcup_{g_j \in \mathcal{D}} \bigcup_{\substack{\langle l(u), l(u,v), l(v)\rangle \\ \in R_{i-1}}} E_j(\langle l(u), l(u,v)l(v)\rangle)$$

ただし，$E_j(\langle l(u), l(u,v)l(v)\rangle) = \{(u',v') \in E_j | l(u') \neq l(v') \neq l(u) \wedge l(u') \neq l(v') \neq l(v)\}$ である．
すなわち，R_j は (性質 1) R_j 内のラベル対は少なくとも 1 つ以上 $\mathcal{D}$ 内のいずれかのグラフで隣接しており，(性質 2) ラベル対の両端点は R_j 内の他のラベル対と重複しない，という 2 つの性質を満たした j 個のラベル対の集合である．提案手法は
(2-1) R_j からラベル対 r を取り出す．
(2-2) $\mathcal{D}$ 内の各グラフ g_i に対して，r を含む場合，それらのノードを 1 ノードに集約する．
この 2 つの手順を R_j の要素がなくなるまで繰り返すことで，要約データベース $\mathcal{D}'$ を構築する．最後にグラフ要約のアルゴリズムを Algorithm 2 に示す．

Algorithm 2 SUMMARIZE$(\mathcal{D}, q)$ AND q'

Input: Graph database $\mathcal{D}$, a query q.
Output: Summarized Graph Databases $\mathcal{D}_i$
集約された頂点 : supernode

```
Sampling, R = {r_1, r_2, ..., r_j};
for each g ∈ D do do
  for each r_i ∈ R do do
    for each e = (v_1, v_2) ∈ E do do
      if v_1 and v_2 are not supernode
      and (l(v_1),l((v_1, v_2)),l(v_2) ) == r_i then
        Summarize v_1 and v_2 to v_i;
In the same way, summarize q to q';
Output D', q';
```

3.2 探　　索

本節では探索について説明する．要約データベース $\mathcal{D}'_j (1 \leqq j \leqq i)$ から，対応するラベル対集合 R_j によって要約されたクエリグラフ q'_j に対する Top-k 相関グラフ T_j をそれぞれ検出する．このとき，i は乱択アルゴリズムによって複数回実行されるイテレーション回数であり，詳細は 3.5 節にて述べる．提案手法では要約データベースにおける Top-k 相関グラフ問合せの処理には先行研究 TopCor [1] を利用する．

この要約グラフデータベース $\mathcal{D}'$ における TopCor による探索は $\mathcal{D}$ に対する探索に比べて探索する候補グラフ数と相関値計算のコストが小さくなると考えられる．したがって $\mathcal{D}'$ での Top-k 相関グラフの探索は $\mathcal{D}$ での Top-k 相関グラフ探索の時間に比べて小さくなると考えられる．

乱択アルゴリズムの導入： $\mathcal{D} = \{g_1, g_2, ..., g_N\}$ とクエリグラフ q が与えられたとき，ラベル対集合 R を用いて $\mathcal{D}$ 内の各グラフを要約した $\mathcal{D}' = \{g'_1, g'_2, ..., g'_N\}$ と要約されたクエリグラフ q' を構築する．このとき，$\mathcal{D}$ における q の Top-k 相関グラフ T と，$\mathcal{D}'$ における q' の Top-k 相関グラフ T' を考える．T と T' はそれぞれ k 個の部分グラフを含む集合である．ある部分グラフ g について R によって要約されたグラフを g' とすると，要約により以下の 2 つの場合が考えられる．

定義 4 (偽陰性)**.** $g \in T$ であるが，$g' \notin T'$ のとき，グラフ要約による偽陰性である．

定義 5 (偽陽性)**.** $g \notin T$ であるが，$g' \in T'$ のとき，グラフ要約による偽陰性である．

グラフ要約処理はランダムに辺を集約するため，集約の順序やラベル対集合 R の内容に依存して Top-k 相関グラフ問合せにおいて，上記の偽陽性・偽陰性を生じさせる場合がある．そこで提案手法では乱択アルゴリズムを導入することで確率的に偽陽性・偽陰性の排除を図る．具体的には，複数のラベル対集合 $R_j (1 \leqq j \leqq i)$ を構築し，それぞれの集合をもとに要約デー

タベース $\mathcal{D}'_j$ を構築する．複数のラベル対集合によって要約されたデータベースは互いに異なるグラフデータベースとなる．

3.3 検　　証

この節では検証について説明する．i 個の要約データベースに対して Top-k 相関グラフ問合せを行った $T_j(1 \leqq j \leqq i)$ に対して正確な Top-k 相関グラフを求めるために検証を行う．T_j に含まれる k 個の要約グラフは，$\mathcal{D}$ の各グラフにおける隣接リスト情報とラベル対集合 R に基づき要約を解除して復元する．

復元の操作としては次の手順である．グラフ g の要約グラフを g'，要約グラフデータベースに相関問合せをした結果獲得した T' とする．最初に g' のある頂点について集約された頂点であった場合は元々の 2 つの頂点を考えるため，R を参照する．2 頂点間の辺をつなげた後に g の隣接リスト情報から隣接する頂点を選択し辺をつなげる．集約された頂点ではない場合は隣接リストから隣接頂点をそのままつなげる．この操作を繰り返し行うことでグラフの復元を行う．このとき，$\mathcal{D}$ において復元したグラフのスーパーグラフとなるグラフを一意に求める．T' の全ての部分グラフに対して同様の操作をし，得られたグラフ集合に対して探索を行う．したがって，検証では復元したグラフを結合したグラフ集合の探索を行い，厳密な Top-k 相関グラフを求めるために検証を行う．

Algorithm 3 VELIDATE($\mathcal{D}'_1, \mathcal{D}'_2, \ldots, \mathcal{D}'_i, T_1, T_2, \ldots, T_i$)

Input: Summarized Graph databases $\mathcal{D}'_1, \mathcal{D}'_2, \ldots, \mathcal{D}'_i$, Graph sets $T_1, T_2, \ldots, T_i$.
Output: The Top-k correlative graphs with respect to q.
i : 反復回数
1: Graph Set G;
2: **for** j=1 to i **do**
3: **for** each $g' \in T_j$ do **do**
4: restore $g \leftarrow g'$;
5: G UNION g;
6: Call TopCor(G,q,k);
7: Output the graphs in Q_{cur};

3.4 Filtering について

本節では，提案手法における Filtering について説明する．提案手法では，乱択アルゴリズムにより，要約グラフデータベースに対して相関問合せを複数実行する．その際に，グラフ要約によって探索範囲が縮小されるが，偽陰性・偽陽性により探索範囲が $\mathcal{D}_q$ に限定されるとは限らない．提案手法における Filtering では探索範囲を $\mathcal{D}_q$ 以下に限定することで探索のコストを削減する．Filtering は探索と検証のそれぞれ段階に適用することができ，以下でそれぞれについて説明する．ただし，グラフデータベースの任意のグラフは整数 GraphID を持ち．探索によって生成される部分グラフには生成される元のグラフの GraphID が対応する．また，要約によって生成される要約グラフにも，要約前の GraphID が割り当てられるものとする．

探索における Filtering：探索では，クエリグラフを要約した q' を含むグラフ集合 $\mathcal{D}_{q'}$ を探索する．探索における Filtering では，$g' \in \mathcal{D}_{q'}$ について，g' の GraphID が $g \in \mathcal{D}_q$ のいずれかの GraphID と一致する場合のみ探索対象とする．この Filtering の操作によって，$\mathcal{D}_{q'}$ に対する探索範囲を $|\mathcal{D}_q|$ 以下に限定することができる．

検証における Filtering：検証では，要約部分グラフ g' を復元し，復元したグラフ集合に対して相関問合せを行う．検証における Filtering では，Algorithm6 の 4 行目において，g' を復元した g が $\mathcal{D}_q$ のいずれかのグラフから成長した部分グラフであるかを調べる．したがって，$g' \in \mathcal{D}_{q'}$ について，g' の GraphID が $g \in \mathcal{D}_q$ のいずれかの GraphID と一致している場合のみ．4 行目の復元操作を行う．この Filtering の操作によって，探索範囲を $|\mathcal{D}_q|$ 以下に限定することができる．

3.5 偽陰性・偽陽性の理論解析

本節では，乱択アルゴリズムを導入することで確率的に偽陽性・偽陰性の排除を図るために反復回数を求める．一回の反復においてグラフ要約によって偽陰性となる確率 P は以下の補題 1 のように定めることができる．

補題 1. 一回のイテレーションで偽陰性が生じる確率 P は $P = 1 - (1 - \frac{k}{l})^{\alpha}$ である．

k, l はそれぞれ，要約されるラベル対の数とグラフデータベースのラベル対の数である．また，α はグラフデータベースの平均次数とクエリグラフの平均次数の差とクエリグラフ頂点数の積で与えることができる．

Proof. 整数 k, l をそれぞれ，一回のイテレーションで集約される $\langle l(u), l(u,v)l(v)\rangle$ の種類数，グラフデータベースのラベル $\langle l(u), l(u,v)l(v)\rangle$ の種類数とする．このとき,$k \leqq l$ である．グラフデータベースのある辺 e_{ij} が $\langle l(u), l(u,v)l(v)\rangle$ を持つとき，e_{ij} が集約される確率は $\frac{k}{l}$ である．対して，ある辺が集約されていない確率は $1 - \frac{k}{l}$ であり，α 本の辺が全て集約されていない確率は $(1 - \frac{k}{l})^{\alpha}$ となる．したがって，α 本の辺の内一本以上が集約されている確率は，$1 - (1 - \frac{k}{l})^{\alpha}$ となる．この α が偽陰性をもたらす辺の本数であれば良いため，$\alpha \approx (\bar{d_G} - \bar{d_g})|V_g|$ となる．このとき，$\bar{d_G}$ はグラフデータベースの平均頂点次数，$\bar{d_g}$ は任意の部分グラフの平均頂点次数，$|V_g|$ は任意の部分グラフの平均頂点数である．したがって， 一回のイテレーションで偽陰性が生じる確率 P は $P = 1 - (1 - \frac{k}{l})^{\alpha}$ である． □

補題 1 を用いて，乱択アルゴリズムの反復回数を以下の定理 1 のように定めることができる．

定理 1. ϵ 以下の確率で偽陰性を排除するためには，以下の t 回以上の反復を行う．ただし，P は補題 1 で求めた一回のイテレーションで偽陰性が生じる確率である．

$$t = \left\lceil \frac{\log(\epsilon)}{\log(P)} \right\rceil$$

Proof. 補題 1 より一回のイテレーションによる偽陰性になる確率は P であるため，一回のイテレーションで偽陰性が生じない確率が $(1 - P)$ となる. したがって，t 回のイテレーション後に偽陰性とならない確率は，$(1 - (1 - P))^t$ となる．これより，

ϵ 以下の確率で偽陰性を排除するためには，以下の変形によって求められる t 回以上の反復を行う．

$$(1-(1-P))^t \leqq \epsilon$$

$$t\log(1-(1-P)) \leqq \log(\epsilon)$$

$$t\log(P) \leqq \log(\epsilon)$$

$$t \geqq \frac{\log(\epsilon)}{\log(P)}$$

よって，乱択回数 t は $t = \left\lceil \frac{\log(\epsilon)}{\log(P)} \right\rceil$ となる． □

提案手法は各グラフデータベースにおいて定理 1 で求めることのできる回数だけ要約データベースを作成し，探索を行う．

4 評価実験

提案手法の計算時間と処理精度の評価を行うために，実データを用いて提案手法と先行研究 TopCor [1] の比較を行った．精度の評価指標には適合率を用いる．適合率は，提案手法が出力した k 個の相関部分グラフと，TopCor が出力した k 個の相関部分グラフが一致した割合とする．実験は以下の 3 つの点に着目して提案手法の有効性を検証する．

- 出現頻度の異なるクエリグラフに対する性能変化
- Top-k の値 k を変化させた場合における性能変化
- データベース $\mathcal{D}$ に含まれるグラフ数に対する性能変化

提案手法は事前計算部分である **(1) グラフ要約**と問合せ処理である **(2) 探索**と **(3) 検証**の 2 つに分けられる．以降の節における提案手法は，使用するデータベース毎に理論解析に基づき，要約グラフデータベースの数を決定する．

4.1 実験環境とデータセット

提案手法のアルゴリズムは C++を用いて実装し，コンパイルオプションには-O3 を用いた．グラフは隣接リスト構造を用いて実装した．部分グラフ同型判定には最新のアルゴリズム [20] を用いた．実験は Intel Xeon Platinum 8268 2.9 GHz , 1 TiB RDIMM で構成される Linux サーバ上で行った．

実験に用いたデータセットは実際の癌や AIDS などのタンパク質や酵素のグラフデータベースである [13–16]．これらのグラフデータベースは原子を頂点とし，原子間の結合を辺として表現されている．データセット nci10k, nci20k, nci40k, nci60k, nci100k はそれぞれ 10,000 , 20,000 , 40,000 , 60,000 , 100,000 個のグラフから構成される．ZINC [16] , DUD-E [14] , LIT-PCBA [15] はそれぞれ特定の性質を持つタンパク質によって構成されるグラフデータベースである．これらのグラフデータベースの詳細を表 1 に示す．実験に用いるクエリグラフは [0.001,0.005] , (0.005,0.01] , (0.01,0.03] の範囲の支持度を持つクエリグラフを利用し，それぞれ Q_1, Q_2, Q_3 とする．

表 1: グラフデータセット

データセット	グラフ数	平均頂点数	ドメイン
nci10k	10,000	32.1994	蛋白質
nci20k	20,000	34.6061	蛋白質
nci40k	40,000	35.2219	蛋白質
nci60k	60,000	35.5829	蛋白質
nci100k	100,000	36.6021	蛋白質
ZINC:GPCR-A	200,000	31.7702	7tm1 膜受容体
ZINC:protease	84,000	33.397	プロテアーゼ受容体
ZINC:reductase	20,000	26.1088	レダクターゼ酵素
DUD-E:AA2AR	34,000	28.0075	アデノシン受容体
LIT-PCBA:ADBR2	312,500	24.9887	アデノシン受容体 β
LIT-PCBA:PKM2	246,069	24.4584	ピルビン酸キナーゼ
LIT-PCBA:ALDH1	145,133	23.8583	アルデヒド脱水素酵素
LIT-PCBA:MAPK1	62,937	24.0814	MAP キナーゼ

4.2 事前処理の性能評価

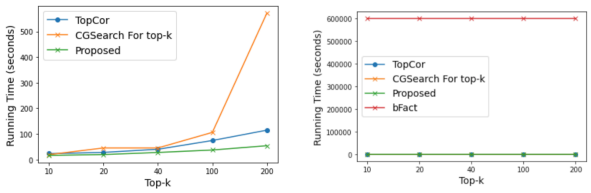
本節では提案手法における事前処理であるグラフデータベースの要約処理について実験を行う．提案手法において，事前処理は複数の要約グラフデータベースを構築し, それぞれのデータベースに対する事前処理の合計の実行時間を表 2 に示す．頂点集約率は要約前のグラフデータベースの平均頂点数を要約後のグラフデータベースの平均頂点数で割った数値である．表 2 より，グラフデータベースに対するグラフ要約には構成グラフ数が大きくなる程，事前処理に時間がかかることがわかる．事前処理は，グラフデータベースの構成グラフ数に比例して実行時間が大きくなるが，4.3 節以降で後述する問合せ処理の時間に比べて小さい．

表 2: 事前処理の実行時間

データセット	グラフ数	実行時間 [sec]	頂点集約率
nci10k	10,000	3.592	0.636
nci20k	20,000	7.969	0.645
nci40k	40,000	15.537	0.660
nci60k	60,000	25.387	0.659
nci100k	100,000	44.315	0.701
ZINC : GPCR-A	200,000	74.325	0.580
ZINC : protease	84,000	32.360	0.569
ZINC : reductase	20,000	6.52	0.589
DUD-E : AA2AR	34,000	11.00	0.606
LIT-PCBA:ADBR2	312,500	89.600	0.603
LIT-PCBA:PKM2	246,069	72.090	0.594
LIT-PCBA:ALDH1	145,133	40.485	0.606
LIT-PCBA:MAPK1	62,937	18.285	0.582

4.3 問合せ処理における比較手法

問合せ処理のにおける比較手法として，TopCor [1] , CGSearch for top-k , bFact [21] がある．TopCor は先行研究であり，グラフデータベースに対する Top-k 相関問合せを行う手法である．CGSearch for top-k は，閾値ベースのグラフデータベースに対する相関問合せを行う CGSearch [2] を Top-k 相関問合せ問題に対応させた手法である．CGSearch は閾値 θ を超える相関値を持つ部分グラフを全列挙する手法であるため，Top-k 問題に対応させるためには，Top-k に対応した閾値 θ を求める必要がある．しかし，異なるクエリグラフに対して閾値 θ を決定す



(a) 3 つの比較手法と提案手法の実行時間比較 (b) bFact と他手法の実行時間比較

図 4: 比較手法との実行時間比較

るのは困難である．そのため，閾値 θ を $\theta = (1 - s * i)$ のように，ステップ $i = 1, 2, 3, ..$ 毎にサイズ s を減らしながら，上位 k 個の相関グラフを検出まで相関問合せを行う．本論文では CGSearch for top-k は $s = 0.1$ を与えるものとする．

bFact [21] は，グラフデータベースから部分グラフの相関ルールを検出する手法である．この手法では，ユーザによって与えられた閾値を超える相関値を持つ 2 つの相関部分グラフ対を検出する．相関尺度は bFact によって定義された相関尺度 $confAll, confTog$ を用いており，詳細な定義は論文を参照されたい．また，その他のパラメータは論文で用いられていた値を用いる．実験では，CGSearch と同様に，Top-k 相関問合せ問題に対応させるために，上位 k 個の相関ルールが検出されるまで，閾値 θ を推移させながら実行を行う．

上記の 3 つの比較手法 TopCor , CGSearch for top-k , bFact について，提案手法との問合せ処理における実行時間を比較する．図 4a, 4b では，nci10k のデータセットに対して，$Q1$ の範囲に属するクエリを与えて，Top-k の k の値を変えて実行時間を比較した．図 4b では， bFact が他の手法に比べて極度に大きいことがわかる．これは bFact が相関ルール分析を行う前処理のための，グラフデータベースに対する頻出部分グラフ検出に大きなコストがかかっているためである．$Q1$ $Q3$ の範囲に属する低頻度なクエリを含む相関ルールを検出するためには，頻出部分グラフ検出において低頻度の部分構造を検出する必要がある．そのため，グラフの支持度の単調性より，頻出部分グラフ検出に与える頻度閾値はクエリグラフの出現頻度より小さくなければならない．したがって，bFact では低頻度クエリに対しては頻出部分グラフ検出に多くの時間を要する．図 4b では実行時間が 7 日を超えた段階で実行を打ち切った．詳細のため，図 4a では，bFact を除いた実行時間の比較を行う．提案手法は，TopCor と CGSearch for top-k に対して，それぞれ最大 2.11 倍，10.74 倍高速に問合せ処理を行うことができた．Top-k の k の値が大きくなるほど，TopCor , CGSearch for top-k に対して，提案手法が高速になることがわかる．特に，CGSearch for top-k では検出する相関グラフ数が大きくなると，ステップサイズを減少させてより多くの CGSearch の実行を必要とするため，実行時間が大きくなる．

4.4 異なるクエリグラフによる影響

提案手法におけるクエリグラフを Q_1 から Q_3 まで変化させたときの問合せ処理の時間と TopCor の計算時間を比較した結果を図 5 に示す．提案手法は Filtering を行わない $GraphUnion$ と探索と検証のそれぞれにおいて Fitlering を行う $FilteringIteration$ と $FilteringValidation$ の三つがある．グラフデータベースは nci10k , nci20k , ZINC:reductase , DUD-E:AA2AR を用いる．

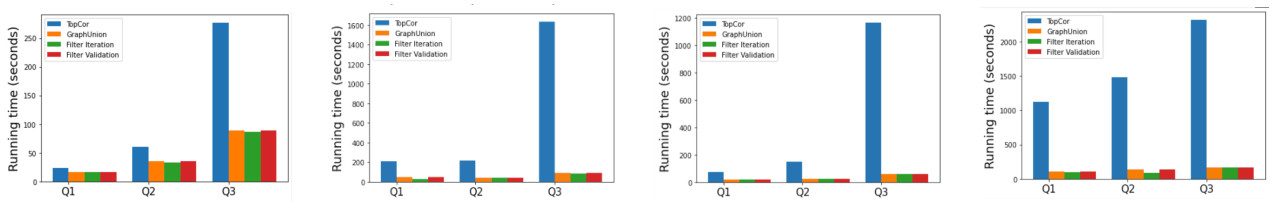
図 5 より，どのクエリグラフに対しても提案手法は TopCor よりも問合せ処理が高速であり，クエリグラフの支持度が大きいほど提案手法が高速になることがわかる．図 5a,5b の Q3 において，提案手法は TopCor よりもそれぞれ最大 3.3 倍，18.14 高速に問合せ処理を行うことができた．また，図 5c では Q3 において，提案手法は TopCor よりも最大 19.47 倍高速に，図 5d では Q3 において，最大 14.17 倍高速に問合せ処理を行うことができた．これはグラフ要約によって相関問合せのの候補グラフの数が小さくなることや相関判定のコストが小さくなることが高速化の要因だと考えられる．また，異なるグラフデータベースに同じ範囲の支持度を持つクエリグラフを与えた場合は同様の傾向が見られることがわかる．

次に，提案手法で得られた Top-k 相関グラフの精度の結果を示す．図 8 は，Q_1 から Q_3 に属する複数のクエリグラフに対して，提案手法で得られた Top-k 相関グラフの適合率の分布を示す．提案手法における $GraphUnion$ と $FilteringIteration$, $FilteringValidation$ のそれぞれに対しての適合率を示し，Filtering による精度の変化について調べる．適合率は，先行研究 [1] で得られてた部分グラフを回答とし，それに対する提案手法によって得られた部分グラフの適合率を考える．図 8 において，提案手法の適合率は多くの場合で 100%になり，それ以外の場合も高い精度であることがわかる．また，Filtering によって探索範囲を縮小させた場合でも，$FilteringIteration$, $FilteringValidation$ の共に高い精度を保っていることがわかる．$FilteringIteration$ で一部の実行で精度低下が見られるが，平均として精度は大きく変わらなかった．これより，理論解析によって得られたイテレーション回数を繰り返すことで高い精度を保証することができると考えられる．

4.5 k による影響

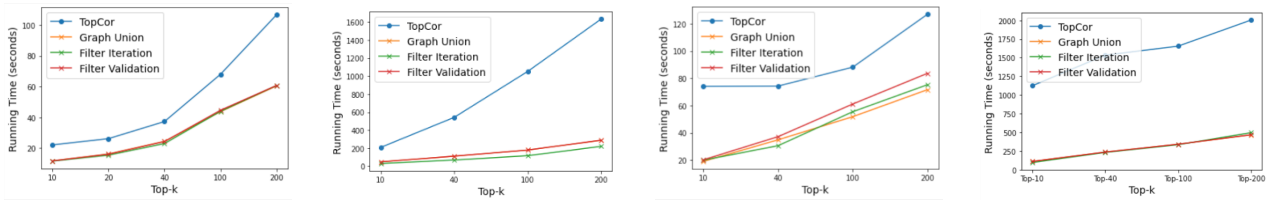
提案手法と既存手法において，k の値を 10 から 200 まで変えて比較実験を行う．図 6 において実験に用いたデータベースは 5 と同様であり，クエリグラフには支持度が Q1 に属するグラフを使用した．

図 6 より，どの k でも提案手法が高速であることが分かる．図 6a，6b では，k が大きくなると徐々に高速化の倍率が上がり，図 6b において $k = 200$ で約 5.70 倍の高速に問合せ処理を行うことができた．また，図 6b において，$FilteringIteration$ は $GraphUnion$ に対して，最大 1.31 倍高速であった．図 6c,6d では，$k = 10$ から $k = 200$ までは高速化の倍率はあまり変わらなかったが，図 6c では最大 3.86 倍，図 6d では最大 14.17 倍高速に問合せ処理を行うことができた．k の値が上昇すると求める相関グラフが多くなるとともに，探索の過程での k 番目の相関値は小さくなるので，TopCor での枝刈りの基準は小さくなる．そのため，k の値が大きくなると実行時間が大きくなる．



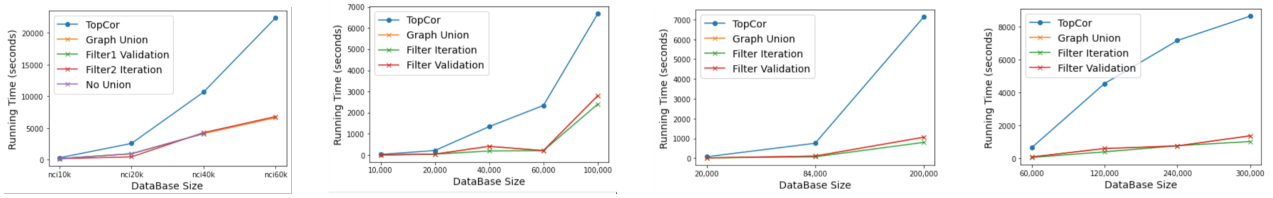
(a) nci10k (b) nci20k (c) ZINC : reductase (d) DUD-E : AA2AR

図 5: 異なるクエリでの問合せ処理時間比較



(a) nci10k-q1 (b) nci20k-q1 (c) ZINC : reductase (d) DUD-E : AA2AR

図 6: 異なる k での問合せ処理時間比較



(a) Nci-Q1 (b) Nci-Q2 (c) ZINC-Q1 (d) LIT-PCBA-Q1

図 7: 異なるデータベースでの問合せ処理時間比較

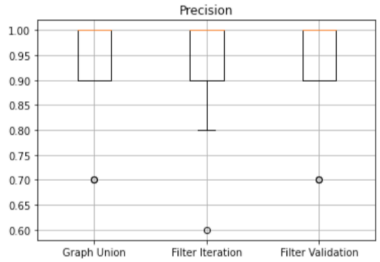


図 8: 適合率

しかし，要約グラフデーベース内での探索は TopCor での探索に比べて候補グラフ数が小さくなるとともに，支持度計算のコストも元々のグラフデータベースに比べて小さいため，提案手法では k の値が大きくなっても問合せ処理の実行時間が抑えられると考えられる．

4.6 データベースのサイズによる影響

データセットのサイズを変えて提案手法と既存手法の比較実験を行う．データセットは nci10k から nci100k と ZINC:GPCR-A,protease,reductase，LIT-PCBA を使用したクエリグラフは Q_1, Q_2 に属するグラフを使用し，k=10 である．これらの実験結果をまとめたものを図 7 に示す．

図 7 の 7a より，nci100k のとき，提案手法は最大で 6.67 倍高速に処理することができることが分かる．また，図 5c では，200,000 個のグラフから構成される GPCR-A において，提案手法は最大で 8.90 倍高速に処理することができた．これはデータベースが大規模になると，候補グラフ数が大きくなると共に支持度計算のコストが大きくなるのに対して，提案手法では要約データベースを考えることで候補グラフの探索する範囲を縮小し，相関問合せのコストを小さくしたためであると考えられる．また，図 5c では $FilteringIteration$ は $GraphUnion$ に対して，最大 1.32 倍高速であった．これは，グラフデータベースが大きくなると，グラフ要約による僅かな偽陽性によって探索範囲が大きくなることが考えられる．グラフデータベースが大きくなる場合では，Filtering によって探索範囲を限定することがより有効に働くと考えられる．

5 ケーススタディ

この節ではケーススタディについて説明する．

アデノシン受容体における活性化構造の検出：提案手法の有効性を検証するために，タンパク質の活性化物質と非活性化物質，デコイの構造物質から構成されるグラフデータベースに対して，活性化物質の構造をクエリグラフを与えた場合の相関部分グラフを検証する．グラフデータベースは，DUD-E [14] の AA2AR のデータセットを用いる．このデータセットはアデノシン受容体のグラフデータから構成されるグラフデータベースである．AA2AR のデータセットは，active と inactive , decoy のタンパ

(a) クエリ　　(b) 回答

図 9: アデノシン受容体におけるケーススタディ

ク質グラフデータから構成される．ここで，active は活性化物質の構造を表現したグラフデータであり，inactive と decoy はそれぞれ非活性化物質と構造や特性は類似しているが，活性化物質ではない構造を表現したグラフデータである．AA2AR は，active と inactive , decoy はそれぞれ 3,000 個，200 個，30,000 個あり，計 33,200 個のタンパク質から構成される．実験に使用するパラメータは $k = 50$ である．クエリグラフは AA2AR の active の区分のみで出現する部分グラフを与える．

実験に与えたクエリグラフと得られた回答グラフの例を図 9 に示す．AA2AR のデータセットに active の図 9a のようなクエリグラフを与えた場合，active のみに出現する構造図 9b を検出することができた．また，この場合に提案手法が出力する相関部分グラフに対して，active だけのデータベースと active,inactive,decoy をすべて含むデータベースでの出現頻度を計測する．結果として提案手法で得られた 50 個の部分グラフの内 84%においてアデノシン受容体を活性化させる化合物を獲得することができた．これより，提案手法は活性化物質をクエリグラフとして与えることで，活性化物質と共に出現する活性化物質の部分構造を得ることができた．アデノシン受容体の活性化物質は，一般に抗炎症性として分類できる一連の応答を生成する [23]．これにより提案手法ではタンパク質データベースなどの領域にて有用な知識発見の扶助となることが考えられる．

6　結論と今後の展望

本研究では事前に要約したデータベースを構築することで相関判定コストを下げ，高速で高精度な Top-k 相関グラフ問合せ手法を提案した．実データを用いた評価実験により，提案手法は既存手法よりも高速に計算できることが示された．

また，グラフ要約に関しても提案手法では集約されたノードは集約しないという要約手続きを行ったが，要約率を上げるために集約されたノードも集約することでさらに集約率を上げることも考えられる．これによって要約率が上がることで相関問合せのコストをさらに下げることができ，より高速な相関問合せが可能になると考えられる．

謝　　辞

本研究の一部は，JST さきがけ (JPMJPR2033) ならびに JSPS 科研費 (JP22K17894) の支援を受けたものである．

文　　献

[1] Yiping Ke, James Cheng, and Jeffrey Xu Yu. Top-k Correlative Graph Mining. In Proc. SIAM International Conference on Data Mining , Pages 1038-1049, 2009.

[2] Yiping Ke, James Cheng, and Wilfred Ng. Correlation search in graph databases. In Proc. ACM SIGKDD international conference on Knowledge discovery and data mining , Pages 390-399, 2007.

[3] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In Proc. IEEE International Conference on Data Mining , Pages 721, 2002.

[4] M. Kuramochi and G. Karypis. Frequent Subgraph Discovery. In Proc. IEEE International Conference on Data Mining , 2001.

[5] Y. Ke, J. Cheng, and W. Ng. Efficient correlationsearch from graph databases. In Proc. IEEE Transactions on Knowledge and Data Engineering , Pagea 1601-1615 , 2008.

[6] Y. Ke, J. Cheng, J. Xu Yu Efficient Discovery of Frequent Correlated Subgraph Pairs. In Proc. IEEE International Conference on Data Mining , Pages 239-248 , 2009

[7] A. Prateek, A. Khan, A. Goyal and S. Ranu Mining Top-k Pairs of Correlated Subgraphs in a Large Network. In Proc. the Very Large Data Base Endowment , Pages 1511-1524 , 2020.

[8] J. Cheng, Y. Ke, and W. Ng. FG-Index: Towards verification-free query processing on graph databses. In Proc. ACM SIGMOD international conference on Management of data , Pages 857-872, 2007.

[9] H. Reynolds. The analysis of cross-classifications. The Free Press, New York, 1977

[10] C. Chen, C. X. Lin, M. Fredrikson, M. Christodorescu,X. Yan, and J. Han. Mining Graph Patterns Efficiently via Randomized Summaries. In Proc. the Very Large Data Base Endowment , Pages 742-753, 2009.

[11] M. Worlein, T. Meinl, I. Fischer, and M. Philippsen. A Quantitative Comparison of the Subgraph Miners MoFa, gSpan, FFSM, and Gaston. In Proc. Conference on Principles and Practice of Knowledge Discovery in Databases , Pages 392-403 , 2005.

[12] R. Motwani, P. Raghavan Randomized algorithms First published 1995 , the Press Syndicate of the University of Cambridge , ISBN 0-521-47465-5

[13] NCI Databases. https://cactus.nci.nih.gov/ , (February 24th, 2023 Accessed)

[14] DUD-E A Database of Useful Decoys: Enhanced https://dude.docking.org/ , (February 24th, 2023 Accessed)

[15] LIT-PCBA: A dataset for virtual screening and machine learning https://drugdesign.unistra.fr/LIT-PCBA/ , (February 24th, 2023 Accessed)

[16] ZINC20 https://zinc.docking.org/ , (February 24th, 2023 Accessed)

[17] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs. In Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence , Pages 1367-1372, 2004.

[18] H. Shiokawa, T. Amagasa, H. Kitagawa. Scaling Fine-grained Modularity Clustering for Massive Graphs. In Proc. the Twenty-Eighth International Joint Conference on Artificial Intelligence Main track. Pages 4597-4604, 2019.

[19] H. Shiokawa, M. Onizuka. Scalable Graph Clustering and Its Applications. Encyclopedia of Social Network Analysis and Mining , Springer , 2017

[20] S. Sun, Q. Luo. caling Up Subgraph Query Processing with Efficient Subgraph Matching. In Proc. IEEE International Conference on Data Engineering 2019.

[21] M. S. CHOWDHURY , C. F. AHMED , C. K. LEUNG, A New Approach for Mining Correlated Frequent Subgraphs. In Proc. ACM Transactions on Management Information Systems 2022.

[22] Verdonk, M. L.; Berdini, V.; Hartshorn, M. J.; Mooij, W. T. , Murray, C. W.; Taylor, R. D.; Watson, P. Virtual screening using protein − ligand docking: avoiding artificial enrichment. J. Chem. Inf. Comput. Sci.

[23] Haskó G, Cronstein BN Adenosine: an endogenous regulator of innate immunity. Trends in Immunology. 25 (1): 33–39. doi:10.1016/j.it.2003.11.003. PMID 14698282. (January 2004).