グラフ分割を用いた頻出部分グラフ発見手法

蛯原　良尚†　　太田　学‡　　片山　薫‡　　石川　博‡

†東京都立大学工学部電子・情報工学科　　‡東京都立大学大学院工学研究科

１．はじめに

グラフ集合の中から頻出の部分グラフを発見する頻出部分グラフマイニングは、データマイニングにおいて基礎的な問題であり、この技術は、化学、生物学、コンピュータネットワークなど幅広く応用されている。しかし、NP 完全な部分グラフ同型問題を含むので、現状では膨大な計算コストを要する。この問題に対しては X.Yan らによる gSpan[1]や M.Kuramochi らによる FSG[2]が知られているが、これらの手法ではマイニング対象となるグラフ集合には特別な前処理を加えていない。一方 Messmer ら[3]は、グラフ集合のグラフ一つ一つを部分グラフに再帰的に分解し、同型グラフを重ね合わせてグラフ集合全体を木構造状した、分割グラフを用いて、部分グラフ同型問題を効率的に解く手法を提案している。

本研究では、マイニング対象のグラフ集合を分割グラフにすることで、より効率的に頻出部分グラフを発見する手法を提案する。

２．提案手法

我々の提案する手法は大まかに①グラフ分割の前処理②グラフ分割③分割グラフを用いての頻出部分グラフマイニングの３段階がある。

２．１　グラフ分割の前処理

グラフ分割を効果的に行うには、グラフ集合が頻出ではない部分グラフをできるだけ持っていないことが望ましい（理由は２．２で述べる）。与えられたグラフ集合には頻出でない頂点や枝が含まれており、これが頻出でない部分グラフの元となる。また、頂点や枝が頻出かどうかを調べる計算コストは、部分グラフを調べるそれと比べて非常に小さい。そこで頂点や枝の頻出を調べ、頻出でない頂点や枝をあらかじめ取り除いておく。

２．２　グラフ分割

Messmer らの手法[3]を基にグラフ集合を分割し木構造にする。分割グラフのモデル図を図１に示す。図１にあるように、分割した部分グラフ同士が同型である場合は重ね合わせていく。前処理において、頻出でない部分グラフができるだけないと望ましいと述べたのは、この重ね合わせがおきやすくなるからである。

重ね合わせによって得られる効果を例に示す。図１においてグラフ[Ａ－Ａ]の頻出を調べる際、従来手法では元のグラフ集合のグラフα、βについて部分グラフ同型問題を解くことで調べる。一方分割グラフを用いた場合は、小さいグラフから調べてゆき、γについて部分グラフ同型問題を解いた後は、αとβについては調べる必要がなくなる。なぜならば、α、βの部分グラフであるγが調べ終わり、かつ頻出と分かっているので、おのずとαとβも頻出であると分かるためである。

この効果を利用して、頻出部分グラフマイニングにおける部分グラフ同型問題にかかる計算コストを抑えることができる。分割した部分グラフ同士が同型かどうかを調べるにも計算コストを要するが、グラフ分割の計算コスト増加よりも分割の効果を利用した計算コスト減少が大きいと期待される。

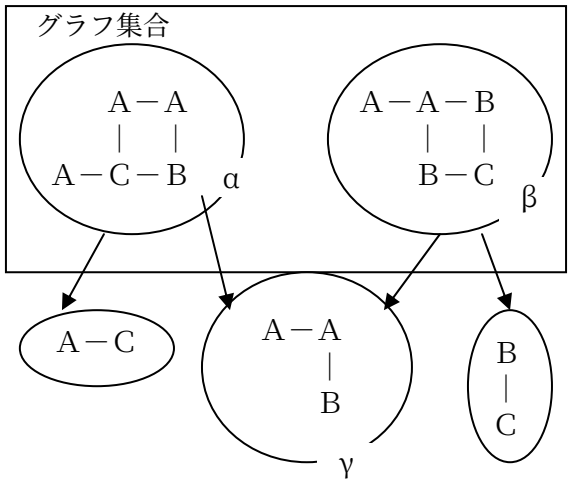

図１：グラフ分割のモデル図

Frequent Subgraph Mining using Graph Decomposition
Yoshinao Ebihara†, Manabu Ohta‡, Kaoru Katayama‡, Hiroshi Ishikawa‡
†Electronics and Information Engineering, Faculty of Engineering, Tokyo Metropolitan University.
‡Graduate School of Engineering, Tokyo Metropolitan University

２．３　頻出部分グラフマイニング

現在、もっとも高速なアルゴリズムとしてgSpan([1])がある。この手法を用いて頻出部分グラフを発見する。頻出であるかどうかを調べるには部分グラフ同型問題を解くことが必要になるが、それに関しても gSpan と同様に従来からこの分野での基本的アルゴリズムとなっている J.R.Ullmann による Backtracking 手法[4]を用いた。

gSpan のアルゴリズムは、部分グラフ同型問題に関しては Backtracking 手法を用いており、独立している。それ以外の部分に関しては、グラフ集合の構造に依存していない。よって本研究では、部分グラフ同型問題を解く際の、対象となるグラフ集合のみを分割グラフに変更した。

３．実験

３．１実験データ

頻出部分グラフマイニングに関する他の方式を参考にできるよう、他の方式 gSpan, FSG([2])の評価実験で共通して使われているデータとして、化合物の化学式のデータ[5]を用いた。

３．２実験方法

自作のアルゴリズムは２．３で述べたように、グラフ分割をせずにマイニングを行うと gSpan と同等のアルゴリズムとなる。そこで実験アルゴリズムとして、以下の２つを用意した。

- 自作アルゴリズムＡ：
  対象となるグラフ集合を分割グラフにする
- 自作アルゴリズムＢ：

対象となるグラフ集合を分割グラフにしない後者を gSpan に見立てて、頻出の閾値（グラフ集合に対して頻出部分グラフを含むグラフの割合の最小値）を 4[%]～20[%]まで変えて、その時の実行時間を測定した（図２）。また[1]にある gSpan の実験結果を併載した。我々が実装したものと性能が大きく異なる原因は、実装技術や実験環境の差によるものと思われるが、今後詳しく検討する予定である。

３．３実験結果

アルゴリズムＡの実行時間にはグラフ分割にかかる時間が含まれているため、頻出閾値が 10[%]から 20[%]にかけては、アルゴリズムＢに比べ実行時間が大きくなった。

しかし、アルゴリズムＢは閾値が小さくなると共に著しく実行時間が増加しているのに対して、アルゴリズムＡでは比較的著しい増加は見られなかった。

これは、頻出閾値が小さくなったときに増加する頻出部分グラフが頻出かどうかを調べる際の、部分グラフ同型問題が、重なり合いを起こしている分割グラフ内で解けることが多く、分割グラフの効果を受けやすいためであると考えられる。

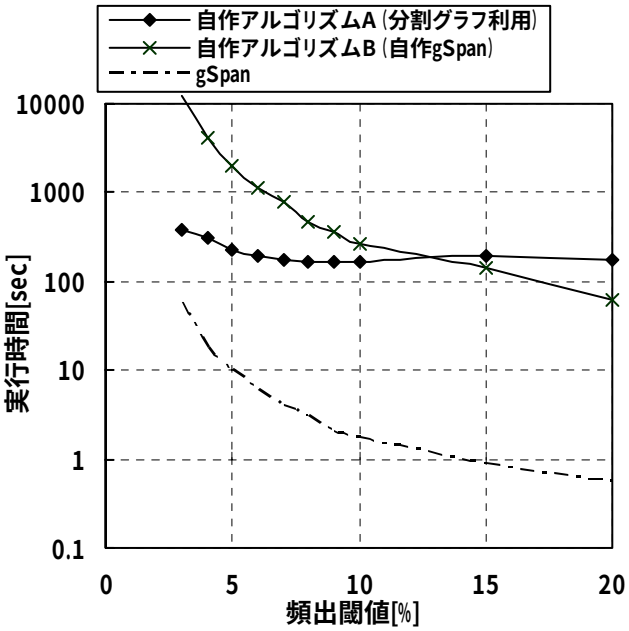

図２：実行時間特性（化合物データ）

４．おわりに

本研究ではマイニング対象のグラフ集合を分割グラフにし、それに基づいて頻出サブグラフマイニングを行った。グラフマイニングでは、主な課題として膨大な計算コストの増加を抑えることにある。本方法では、頻出部分グラフのパターン数が増加した時に対する、著しい計算コストの増加を抑えることができた。

検討の余地としては以下の点があげられる。より重なり合いの多い分割を考案すること、またグラフ集合の総数、ラベル種類数などが異なった場合における、本手法の特性の評価である。

５．参考文献

[1]X.Yan and J.Han. gSpan Graph-Based Substructure Pattern Mining. In Proc. IEEE ICDM’02, pp.721-724, 2002.
[2]M.Kuramochi and G.Karypis. Frequent Subgraph Discovery. In Proc. IEEE ICDM’01, pp.313-320, 2001.
[3]B.T.Messmer and H.Bunke. Efficient Subgraph Isomorphism Detection A Decomposition Approach. IEEE Transactions on Knowledge and Data Engineering, vol.12, No.2, pp.307-323, 2000.
[4]J.R.Ullmann. An Algorithm for Subgraph isomorphism. Journal of the ACM, pp.31-42, 1976.
[5]http://oldwww.comlab.ox.ac.uk/oucl/groups/machlearn.PTE/.