



Title	逆探索法によるグラフ系列マイニングの高速化
Author(s)	猪口, 明博; 生田, 泰章; 鷺尾, 隆
Citation	2010年度科学技術振興機構ERATO湊離散構造処理系プロジェクト講究録. p.505.
Issue Date	2011-06
Doc URL	http://hdl.handle.net/2115/48326
Type	conference presentation
Note	ERATO湊離散構造処理系プロジェクトシンポジウム(第2回): 第73回情報処理学会全国大会イベント企画. 2011年3月2日(水). 東京工業大学 大岡山キャンパス.
File Information	05.inoguchi_ERATO-GTRACE3.pdf



[Instructions for use](#)

逆探索法によるグラフ系列マイニングの高速化

猪口 明博*1 *2, 生田 泰章*1, 鷲尾 隆*1 *3
 *1大阪大学 産業科学研究所, *2科学技術振興機構 さきがけ, *3 科学技術振興機構 ERATO

背景

頻出グラフマイニング

入力 → 頻出する部分グラフの列挙 → 出力

- 支持度の逆単調性
- グラフ同型問題

グラフ系列

- 人間関係ネットワークの変化
- 人: 頂点, 人間関係: 辺
- ホームページのリンク構造の変化
- HTML文章: 頂点, ハイパーリンク: 辺
- 遺伝子ネットワークの変化(進化)
- 遺伝子: 頂点, 相互作用: 辺
- 機械の組み立て
- 部品: 頂点, 隣接する部品間: 辺

頻出グラフ系列マイニング

グラフ系列の集合が与えられたとき, ある頻度以上出現する頻出する部分グラフ系列の列挙すること

頻出する部分グラフ系列の列挙

対象とするグラフ系列

- 頂点数, 辺数が増減する
- 頂点ラベル, 辺ラベルが変化する
- 各頂点は, IDをもつ

GTRACE

基本アイデア

仮定
 グラフ系列中の連続する2つのグラフの間では, 構造が大きく変化することはない, ごく一部の構造のみが変化する.

系列1: [Graphs]

系列2: [Graphs]

コンパイル
 系列1 $\langle (vi, vi, ei, ei, ei), (vi, ed, ed, vd), (ei, ed, vd), (ed, vd) \rangle$
 系列2 $\langle (vi, vi, vi, ei), (vi, ei), (vi, ei, ei, ed, vd), (ei, ed, vd) \rangle$

系列パターンマイニング
 頻出パターンFTS (頻出変換部分系列)
 $\langle (vi, vi, ei), vi, (ei, ed, vd) \rangle$

頻出パターン

変換規則

頂点や辺の追加, 削除, ラベル変更

赤い頂点の追加 $g^{(j)}$ → $g^{(j+1)}$
 青い頂点と緑の頂点の間の辺の削除 $g^{(j+1)}$ → $g^{(j+2)}$

$\langle \dots, g^{(j)}, g^{(j+1)}, g^{(j+2)}, \dots \rangle \rightarrow \langle \dots, vi, ed, \dots \rangle$

6種の変換規則

- 頂点の追加(vi), 頂点の削除(vd), 頂点ラベルの変更(vr)
- 辺の追加(ei), 辺の削除(ed), 辺ラベルの変更(er)

関連のあるFTS

2番と3番の人物は関連がある。
 3番と4番の人物は関連がある。
 → 2番と4番の人物は3番を通して関連がある。
 1番の人物は他の人と関連が無い。

和グラフ

和グラフが連結なら, グラフ系列は関連のあるFTSである。

課題

GTRACEは関連のあるFTSのみを列挙することができず, 膨大な数の関連のないFTSを列挙するため多くの計算時間を要する。

探索の順序

$\langle vi \rangle$
 $\langle (vi, vi) \rangle$ rFTSではない
 $\langle (vi, vi, vi) \rangle$ rFTSではない
 $\langle (vi, vi, vi), ei \rangle$ rFTSではない
 $\langle (vi, vi, vi), (ei, ei) \rangle$
 $\langle (vi, vi, vi), (ei, ei), ed \rangle$

GTRACE-RS

逆探索

- 列挙問題に対する, 効率的なアルゴリズムの構築を可能にする手法.
- 探索空間を全域木によって表現することができれば, 深さ優先探索によって効率良く探索が可能となる.
- すなわち, 全ての解Sについて, $X \in S$ の唯一の親を返す関数P: $S \rightarrow S$ を定義できれば, 効率良く全ての解を探索可能.

提案手法

- Sを関連のあるFTSの集合とする.
- P: $S \rightarrow S$ をみたく所望の関数P(TS)を3つのパーツで構成する. この関数により探索空間を全域木で表現できる.
- P1(TS): TSの中の頂点に関する変換規則の内, 最後の交換規則を削除する.
- P2(TS): 同一の辺に1つ以上の変換規則が適用される, 辺の変換規則の内, 最後の交換規則を削除する.
- P3(TS): 辺に関する変換規則の内, 和グラフが連結性を保つように最後の交換規則を削除する.

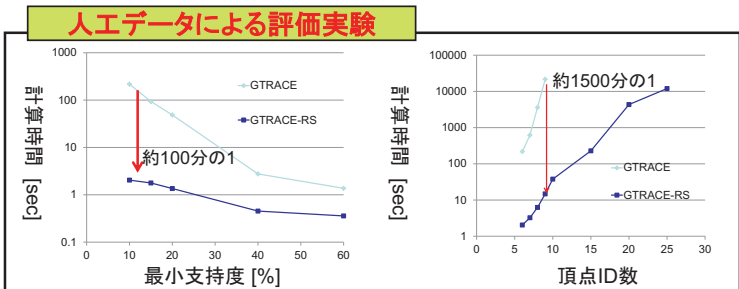
$P_3(\langle ei \rangle) = \langle \rangle$
 $P_3(\langle (ei, ei) \rangle) = \langle (ei) \rangle$
 $P_2(\langle (vi, ei, ei), ed \rangle) = \langle (ei, ei), ed \rangle$
 $P_1(\langle (vi, vi), (ei, ei), ed \rangle) = \langle (vi), (ei, ei), ed \rangle$
 $P_1(\langle (vi, vi, vi), (ei, ei), ed \rangle) = \langle (vi, vi), (ei, ei), ed \rangle$

丸で囲まれた変換規則は付加された変換規則

GTRACEの探索空間

GTRACE-RSの探索空間

探索の順序



まとめ

- 逆探索の原理を用いることによって, 関連のあるFTSのみを探索することが可能となった.
- 従来のGTRACEの探索では関連性のないFTSの探索がほとんどを占めているので, 計算時間, 空間コスト共に大幅に削減することができた.
- 各ステップのグラフがより大きく, 長いグラフ系列についても適用可能となった.