4I1-2

大規模グラフにおけるフラクタル構造検出の高速化

Fast Algorithm for Detecting Fractal Structures of Large Networks

秋葉 拓哉 *1*2 Takuya Akiba　中村 謙弘 *1 Kenko Nakamura　高口 太朗 *1 Taro Takaguchi

*1国立情報学研究所 National Institute of Informatics　*2千葉工業大学 人工知能・ソフトウェア技術研究センター CIT STAIR Lab

The fractality of a network is determined by so-called *box-cover algorithms*. We present a new box-cover algorithm. We theoretically and empirically show that it is orders of magnitude faster than previous algorithms.

1. はじめに

現実世界のネットワークの一部はフラクタル性を持つことが報告されている [9]．グラフのフラクタル性はユークリッド空間の幾何的図形のフラクタル性 [5] を拡張することにより以下のように定義される．正整数 ℓ に対し，半径が ℓ 以下の頂点集合を**ボックス**と呼ぶ．グラフ全体を被覆するために必要なボックスの数を $b(\ell)$ とおく．$b(\ell)$ が ℓ の冪関数となる場合，即ち $b(\ell) \propto \ell^{-d}$ を満たす場合，グラフはフラクタル性を持つと言われる．図 1 に，あるフラクタル性を有するモデルネットワークに対して提案手法を適用した結果を示す．$b(\ell)$ が冪関数に沿っており確かにフラクタル性をもつことが確認される．

この $b(\ell)$ の値を計算する処理は**ボックスカバー**と呼ばれる．フラクタル性を判定するには，理論的には，$b(\ell)$ が最小となるようボックスの配置を定めることが要求される．しかし，現実的にはその計算は NP-hard であり，効率的な計算アルゴリズムは望めない．従って，近似アルゴリズムが開発され使用されてきた．しかし，既存の近似アルゴリズムは計算時間や使用メモリ量が非常に大きく，数百万頂点規模のネットワークに適用することができなかった [8, 7]．

そこで本研究では，大規模ネットワークに適用可能な新たなボックスカバーの近似アルゴリズムを提案する．核となるアイディアは，全てのボックスを陽に計算せず，各ボックスに対して**スケッチ**と呼ばれる確率的データ構造のみを計算し処理を行う点にある．なお，本研究についての詳細は文献 [1] を参照されたい．

2. 提案手法

以下，グラフを $G = (V, E)$ とし，頂点数を n，辺数を m とおく．

2.1 ボックスのスケッチの生成

最初のステップとして，全ボックスのスケッチを計算する．スケッチのデータ構造としては，*bottom-k min-hash* [3, 4] を用いる．これは，全頂点にランダムなランク値を割り振り，ランク値の小さいものから k 個のみを保持するものである．

以下のような手順で効率的に計算が行える．まず，各頂点に自身のみを含むスケッチを持たせる．次に，近傍のスケッチを自身のスケッチに併合することを ℓ 回繰り返す．前のステップで新たに挿入された要素のみを併合の候補にする工夫により更なる高速化が可能である．

連絡先: 秋葉拓哉，国立情報学研究所，東京都千代田区一ツ橋 2-1-2，takiba@nii.ac.jp

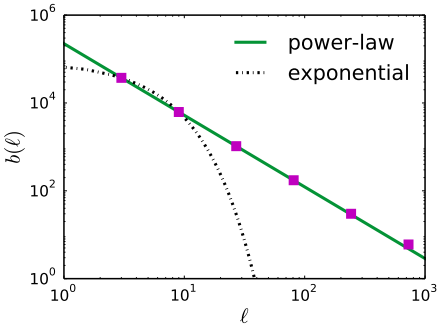

図 1: フラクタル性を有するモデルネットワークにおける $b(\ell)$.

2.2 貪欲法によるボックスの選択

次のステップとして，出力するボックスを選択する．この時，*bottom-k min-hash* による集合のサイズの推定を用いる [3, 4]．この推定値に基づく貪欲法によりボックスを選択してゆく．ここで，推定サイズの大きいボックスから選択するのではなく，被覆された頂点数の推定値が最も大きく増加するボックスを毎回選択することに注意されたい．

このアイディアを素直に実装する場合，最大 n 回選択を行い，毎回 n 頂点について $O(k)$ 時間をかけて評価を行うので，計算量は $O(n^2k)$ となる．しかし，平衡二分探索木や順位キューを用い情報を工夫して管理することにより，$O(nk \log n)$ 時間に高速化することができる．

2.3 理論的解析

BoxCover を全頂点をボックスにより被覆する問題，$(1-\epsilon)$-BoxCover を $(1-\epsilon)n$ 頂点をボックスにより被覆する問題とおく．提案手法の性能について，以下のような理論的な保証を行うことができる．

定理 1 (スケーラビリティ)．提案アルゴリズムは $O((n+m)k \log k \min\{\ell, \log n\})$ 期待時間，$O(nk+m)$ 期待領域で動作する．

定理 2 (精度)．$1 - 1/n$ 以上の確率で，$\epsilon \geq 2\sqrt{5(\ln n)/k}$ なる ϵ に対し，提案アルゴリズムは，$(1-\epsilon)$-BoxCover の解であって BoxCover の最適解から $1 + 2\ln n$ 倍以内のサイズのものを出力する．

表 1: 実行時間（単位は秒）と判定の指標 ($-\log_{10} r_{\mathrm{fit}}$)．DNF は 24 時間以内に終了しなかったかメモリ不足となったことを表す．

グラフ			提案手法		MEMB [8]		GC [8]		MVB [7]		CBB [8]	
モデル	$\|V\|$	$\|E\|$	時間	判定	時間	判定	時間	判定	時間	判定	時間	判定
(2, 2, 4)-flower	172	256	0	0.8	0	1.0	0	28.7	199	1.0	0	28.0
(2, 2, 6)-flower	2,732	4,096	2	1.8	1	2.2	6	27.7	DNF	—	6	27.7
(2, 2, 8)-flower	43,692	65,536	192	2.8	147	4.5	4,749	-6.0	DNF	—	1,839	27.5
(2, 2, 9)-flower	174,764	262,144	982	3.4	3,959	5.7	DNF	—	DNF	—	31,870	27.5
(2, 2, 10)-flower	699,052	1,048,576	8,628	3.5	DNF	—	DNF	—	DNF	—	DNF	—
(2, 2, 11)-flower	2,796,204	4,194,304	62,138	4.0	DNF	—	DNF	—	DNF	—	DNF	—
(2, 1)-BA	250	497	0	-0.9	0	-0.9	0	-0.6	54	-0.5	0	-0.3
(2, 4)-BA	2,000	3,997	1	-2.7	0	-2.0	2	-0.6	DNF	—	404	-0.1
(2, 7)-BA	16,000	31,997	17	-1.3	76	-1.3	154	-0.6	DNF	—	DNF	—
(2, 10)-BA	128,000	255,997	377	-1.5	3,535	-1.5	12,457	-0.6	DNF	—	DNF	—
(2, 13)-BA	1,024,000	2,047,997	6,474	-1.4	DNF	—	DNF	—	DNF	—	DNF	—
(2, 15)-BA	4,096,000	8,191,997	36,125	-1.4	DNF	—	DNF	—	DNF	—	DNF	—

3. 評価実験

3.1 実験方法

本実験には CPU が Intel Xeon 2.67 GHz，メモリが 96GB の Linux サーバを利用した．全てのアルゴリズムは C++ で実装され，gcc 4.8.4 を用いてコンパイルされた．評価には，flower モデル [6] と Barabási-Albert (BA) モデル [2] と呼ばれる人工ネットワークモデルを用いた．flower モデルは（適当なパラメータ設定の下で）フラクタル性を持ち，BA モデルはフラクタル性を持たないことが理論的に示されている [6]．既存のボックスカバーのアルゴリズムとして，GC [8], MEMB [8], CBB [8], MVB [7] との比較を行った．提案手法のパラメータは $k = 128$ とした．

以下の手順でグラフのフラクタル性の判定を行う．ボックスカバーのアルゴリズムによって得られた $b(\ell)$ に対して，冪関数と指数関数のフィッティングをそれぞれ求める．冪関数の残余誤差を指数関数の残余誤差で割ったものを r_{fit} とし，$-\log_{10} r_{\mathrm{fit}}$ を指標とする．$-\log_{10} r_{\mathrm{fit}}$ が正の場合 (即ち $r_{\mathrm{fit}} < 1$ の時)，グラフはフラクタル性を持つと考えられる．そうでない場合，グラフはフラクタル性を持たないとみなす．これは先行研究 [10] にて用いられている手順と等価である．

3.2 実験結果

表 1 が実験結果を表す．まず，提案手法が既存手法と比較して高いスケーラビリティを持つことが読み取れる．既存手法は，メモリ使用量か実行時間のいずれかの問題により数百万頂点を持つグラフを処理できない．一方，提案手法は数百万頂点を持つグラフを処理することが可能である．

次に，フラクタル性の判定性能について，提案手法は誤りなく判定を行うことができていることが分かる．すなわち，判定の指標 $-\log_{10} r_{\mathrm{fit}}$ について，フラクタル性を持つ flower ネットワークでは正の値となり，フラクタル性を持たない BA ネットワークでは負の値となっている．比較的サイズの小さなネットワークについては既存手法も正しくフラクタル性を判定できているが，既存手法が処理できないサイズのネットワークについても正しく判定できていることが提案手法の優位点である．

4. おわりに

本研究では，大規模ネットワークのフラクタル性の検証を可能にするために，ボックスカバーの高速アルゴリズムを提案した．提案手法は集合のスケッチング技術に基づいており，全ボックスのスケッチを高速に計算するアルゴリズムと，スケッチの情報を用いてボックスを貪欲法により効率的に選択するアルゴリズムから成る．実験により，数百万頂点規模の大規模ネットワークに対しても，提案手法が高速かつ正確に動作することを検証した．

謝辞

本研究は日本学術振興会科学研究費補助金 (15H06828)，JST さきがけ，JST ERATO 河原林巨大グラフプロジェクトの支援を受けたものである．ここに記して謝意を表す．

参考文献

[1] T. Akiba, K. Nakamura, and T. Takaguchi. Fractality of massive graphs: Scalable analysis with sketch-based box-covering algorithm. Manuscript, 2016.

[2] A.-L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.

[3] E. Cohen. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. Syst. Sci.*, 55(3):441–453, 1997.

[4] E. Cohen. All-distances sketches, revisited: HIP estimators for massive graphs analysis. *IEEE TKDE*, 27(9):2320–2334, 2015.

[5] K. Falconer. *Fractal Geometry: Mathematical Foundations and Applications, Second Edition*. Wiley-Blackwell, 2003.

[6] H. D. Rozenfeld, S. Havlin, and D. Ben-Avraham. Fractal and transfractal recursive scale-free nets. *New J. Phys.*, 9(6):175, 2007.

[7] C. M. Schneider, T. A. Kesselring, J. S. Andrade, and H. J. Herrmann. Box-covering algorithm for fractal dimension of complex networks. *Phys. Rev. E*, 86(1):016707, 2012.

[8] C. Song, L. K. Gallos, S. Havlin, and H. A. Makse. How to calculate the fractal dimension of a complex network: the box covering algorithm. *J. Stat. Mech.*, 2007(03):P03006, 2007.

[9] C. Song, S. Havlin, and H. A. Makse. Self-similarity of complex networks. *Nature*, 433(7024):392–395, 2005.

[10] K. Takemoto. Metabolic networks are almost nonfractal: A comprehensive evaluation. *Phys. Rev. E*, 90(2):022802, 2014.