1C5-4

大規模ネットワークにおけるコミュニティ抽出手法の改良

Improvement of Community Detection Algorithm in Large Graphs

尾崎直人[*1]　手塚宏史[*1]　稲葉真理[*1]

[*1] 東京大学　情報理工学系研究科　創造情報学専攻

Currently, large-scale graphs have emerged in a variety of analysis such as social networks and it is found that such networks have community structure property. Many algorithms for detecting communities are proposed to analyze networks and Louvain algorithm is well known for it's high speed and high quality. Although Louvain algorithm is high speed compared to other algorithms, higher speed algorithms are required to analyze huge scale networks. We propose a simple pruning technique using scale-free property for Louvain algorithm. Our experiments show that our proposal is faster than twice and almost same quality compared to Louvain algorithm.

1. 研究概要

近年複雑ネットワークに内在するコミュニティ構造の抽出に関する研究が様々な分野において注目を浴びている. コミュニティ構造とは, ネットワークが持つリンクが密な部分グラフのことであり, グラフをコミュニティ毎に分割する高速なアルゴリズムがこれまで数多く提案されてきた. しかしながらソーシャルグラフに代表される巨大ネットワークは年々急速に膨張し, さらに高速なコミュニティ抽出手法が求められている.

そこで本研究では, 現在高速かつ精度の高いコミュニティ抽出手法として知られている Louvain 法に対して複雑ネットワークの持つ次数分布のべき乗則の性質を利用し, 精度にあまり影響しない計算処理の枝刈りを行う Louvain-Prune 法を提案し, 実装実験を行い, 精度を同程度に保ちつつ計算速度を倍以上高速化できることを確認した.

2. 関連研究

近年巨大グラフをリンクが密な部分グラフに分割するコミュニティ抽出のアルゴリズムが数多く提案されてきた. ここでは分割のよさを表す指標 Modularity と, これを用いたコミュニティ抽出アルゴリズム Louvain 法を紹介する.

2.1 Modularity

コミュニティ分割の精度を表す指標として, Newman らにより提案された Modularity が広く知られている[Newman 04]. Modularity は実際のコミュニティのインナーエッジの本数から, ランダムグラフとみなした時のインナーエッジの本数の期待値を減算し, 最大値が1となるように正規化した指標である. つまりコミュニティ内頂点のリンクが密で, コミュニティ間のリンクが疎であるほど高い値を出す指標であり, 逆に 0 ではランダムグラフと同等であると言える. Modularity は頂点集合 V の分割 C = $\{V_1, V_2, V_3 \ldots, V_k\}$ を引数とする関数であり, $\bigcup_{i=1}^{k} V_i = V$を満たす. また, ネットワークを隣接行列として表した時の(i,j)成分をA_{ij}, 総リンク数を m, 頂点 i の次数をk_i, 頂点 i が属しているコミュニティをc_i,$\delta(c_i, c_j)$を,頂点 i と j が同じコミュニティに属している場合に 1, それ以外の場合に 0 となるクロネッカーのデルタとしたとき, Modularity Q(C)は式(1)のように表される.

$$Q(C) = \frac{1}{2m}\sum_{i\in V}\sum_{j\in V}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j) \quad (1)$$

コミュニティ抽出問題は多くの場合 Modularity 最大化問題と同等であるが, Modularity 最大化問題は NP 困難に分類されており[Duch 05], これまで数多くの近似解法が提案されてきた.

2.2 Louvain Method

Modularity を用いた代表的な手法として, Blondel らによって提案された Louvain 法[Blondel 08]が挙げられる. Louvain 法は, 局所最適化と同一コミュニティに属する頂点の集約処理により, 高速かつ, 高い Modularity を出すことで知られている.

Louvain 法は2つのフェーズから構成される. 初期状態では, 全頂点がそれぞれ固有のコミュニティに所属するとする.

フェーズ 1 では, 後述するように, 各頂点において最も Modularity を向上させるような隣接コミュニティを探索し, フェーズ2ではフェーズ1において同じコミュニティに属する頂点群を1頂点に集約する. これによりネットワークサイズが縮小される. フェーズ 1 と 2 は Modularity の向上が無くなるまで繰り返され, 収束したときコミュニティ抽出が完了する.

フェーズ 1 においては任意の順に頂点を選択, その頂点の全隣接頂点に対して, 同じコミュニティに属した際に上昇する Modularity の値 ΔQ を求める. この際, 式(1)をそのまま利用すると全頂点, 全リンクを参照する必要があり非効率なため, Blondel らは選択された頂点と, その隣接ノードを統合した際の Modularity 変化量 ΔQ を計算するために, 式(1)から式(2)を導出した.

$$\Delta Q = \left[\frac{\Sigma_{in} + 2_{i,in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \quad (2)$$

ここでは, 頂点 i がコミュニティ C に属し, Σ_{in}を C のインナーエッジの重みの総和, Σ_{tot}を C に含まれる頂点の次数の総和とする. 実際には Blondel らの Louvain 法を実装したプログラムでは式(2)をさらに簡略化した式(3)が利用されている.

$$\Delta Q = e_{iC} - \frac{k_i\,\Sigma_{tot}}{2m} \quad (3)$$

尾崎直人　東京大学　情報理工学系研究科　創造情報学専攻
稲葉真理研究室　修士1年 〈dyrmc694@gmail.com〉

ここでは，頂点 i がコミュニティ C に所属する際，e_{ic}を頂点 i とコミュニティ C に属する頂点 j の間のエッジの重みの総和 $\sum_{j \in C} e_{ij}$とする．式(3)により,最も ΔQ の大きい頂点 i を，コミュニティ C に所属させる処理を全頂点に対して行う．

2.3 Louvain Method の高速化

(1) 塩川らの高速化手法

塩川らは，Louvain 法を逐次集約手法,所属コミュニティが自明な頂点の枝刈り,次数順ノード選択による参照エッジの削減の3つのアプローチにより，Modularity を同程度に保ちつつ，最大で 60 倍高速化可能であることを示した[Shiokawa 13]．一つ目のアプローチは，各頂点がフェーズ 1 の処理において所属するコミュニティが決定した時点で同時に集約処理を行うことにより，フェーズ 1 と 2 が分離していることによる無駄な枝の参照を削減するものである．二つ目は，所属コミュニティが自明な頂点をフェーズ1における探索対象から除外することにより，探索の高速化を図るものである．三つ目は，フェーズ1において次数の昇順に頂点を選択することにより，探索総数を削減するものである．これにより，さらに大規模なネットワークに対して有効な手法であることを示した．

(3) 比較実験

従来の Louvain 法と,塩川らの提案手法の比較実験を行った．本実験で用いたデータセット[Kuneigis 13] [Leskovec 14]を以下に示す．

- Gowalla(196,591 nodes , 950,327 edges)
  - Gowalla ユーザーの交友関係
- DBLP(317,080 nodes , 1,049,866 edges)
  - 論文の共著関係
- Amazon(334,863 nodes , 925,872 edges)
  - 同時購入されやすい商品の関係
- Web-graph(875,713 nodes , 5,105,039 edges)
  - Web ページのハイパーリンク
- Skitter(1,696,415 nodes , 11,095,298 edges)
  - AS ネットワーク
- Youtube(3,223,589 nodes , 9,375,374 edges)
  - Youtube ユーザー同士の交友関係
- Flickr(1,715,255 nodes , 15,550,782 edges)
  - Flickr ユーザー同士の交友関係

実験には CPU が Intel(R) Corei7-3630QM CPU 2.40GHz, メモリが 16GB, OS が Ubuntu(14.04)のものを用いた．プログラムは自ら Java で実装した Louvain 法と，そのプログラムを塩川らの手法に改変したものを使用した．

Graph	Approach	Time(sec)	Modularity
Gowalla	OFF	9.1	0.7087
	ON	8.5	0.6547
DBLP	OFF	55.7	0.8137
	ON	3.4	0.7996
Amazon	OFF	15.0	0.9193
	ON	1.9	0.9082
Web-graph	OFF	95.3	0.9788
	ON	15.5	0.9767
Skitter	OFF	112.5	0.8415
	ON	83.9	0.8204
Youtube	OFF	30.4	0.7126
	ON	45.0	0.6434
Flickr	OFF	107.1	0.6427
	ON	196.4	0.5788

図 1. 従来の Louvain 法と塩川らの手法比較

実行時間は Youtube と Flickr を除くグラフで 1.1～7 倍程度高速化しているが，Modularity は全てのグラフにおいて減少していた．これはアプローチの一つである逐次集約手法によるものであると考えられる．従来の Louvain 法では，フェーズ1は Modularity の向上が収束するまで探索処理が何度も繰り返し行われるが，塩川らの手法では1度の探索のみで隣接頂点と集約されるため，必ずしも最良の頂点同士で集約されるとは限らない．例えば，従来の手法では例え1回目のフェーズ1において頂点 a がコミュニティ P に所属したとしても，フェーズ1の回を重ねてコミュニティが形成されるに従って ΔQ の値が変わり，頂点 a が別のコミュニティ Q に所属を変えるケースがあるが，塩川らの手法ではこのケースが考慮されていない．図 2 は，Flicker のグラフデータでコミュニティ抽出をした際のフェーズ1のループに対しての Modularity(棒グラフ)と処理時間(折れ線グラフ)を表した図であるが，繰り返し探索処理を行う度に Modularity が向上していることが見て取れ，精度を犠牲にしないためには収束するまで引き続き探索処理を行う必要があることが分かる．

3. フェーズ1における探索の枝刈り

ここでは，まず予備実験においてフェーズ1において非効率な探索処理を示し，その結果に基づき枝刈りを行う Louvain-Prune 法を提案する

3.1 予備実験

Louvain 法においてフェーズ 1 は Modularity の向上が収束するまで何度も繰り返されるが，回を繰り返す度に向上の幅は次第に小さくなる．図 2 は，Flicker のグラフデータでコミュニティ抽出をした際のフェーズ1のループの回数(横軸)に対しての Modularity(棒グラフ)と処理時間(折れ線グラフ)を表した図である．3～16 回目では Modularity にはほとんど改善が見られないのにも関わらず 1 秒以上費やしており，非効率であることが分かる．これはフェーズ 1 が実行される度に全頂点に対し隣接頂点との ΔQ を再度計算していることに起因するものであり，さらに大規模なグラフでは繰り返し回数が増える傾向があるため，さらに速度に悪影響を与えることが考えられる．

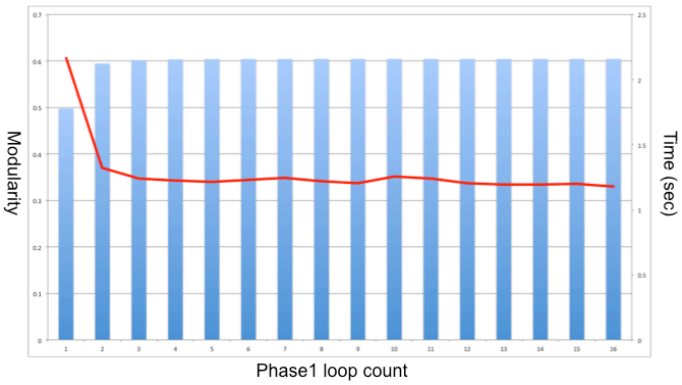

図 2. フェーズ1における計算時間と Modularity

よって精度を保ちつつ速度を向上するには，図 2 における 3～16 回目のような非効率な探索を改善することにより高速化が望めることが分かる．

3.2 Louvain-Prune 法の提案

我々は，式(3)をもとに，頂点の所属先コミュニティが変更しないことが自明な頂点においては探索を行わないことで，精度を保ちつつフェーズ1を高速化することを考える．コミュニティの所属先が次回のフェーズ 1 において変わり得る頂点は，現在の所属先コミュニティ以外のコミュニティに属している頂点に対しての ΔQ が増加する場合か，所属先コミュニティへの ΔQ が減少し，所属先コミュニティ以外への ΔQ を下回っている頂点のみである．

ここでは図 3 のような頂点 i がコミュニティ P からコミュニティ Q に所属先が移った例に沿って隣接頂点に対する ΔQ が変化し，次回の探索処理において所属先が変更しうる4つのケースを例にとって説明する．

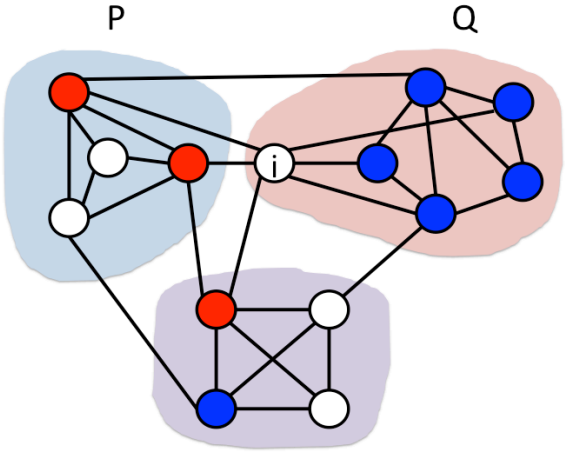

図 3. 頂点 i の所属先が P から Q へ移るケース

1つ目のケースは頂点 i の隣接頂点のうち，Q 以外に属する i の隣接頂点群 R である．R のうち，元々Q に対するリンクを持っていない頂点は次回探索時に Q に所属する機会が生まれ，元々Q に対するリンクを持っていた場合は Q に対する ΔQ が増加する．2つ目のケースは頂点 i の隣接頂点のうち，Q に属する頂点群 S である．式(3)の頂点 i の次数と Q に属する頂点の次数の総和をネットワークのエッジ数で除算した値の増加量に比べて,頂点 i と S の頂点群の間のリンクの重みが小さい場合は S に属する頂点群から Q に対する ΔQ が減少し，相対的に他の隣接コミュニティへの ΔQ が大きくなる．3 つ目のケースは P の隣接頂点群のうち，頂点 i へのリンクをもたない頂点群 T である．P から頂点 i が抜けたことにより，Σ_{tot}が頂点 i の次数分減少するため，T から P に対する ΔQ が増加する．4 つ目のケースはコミュニティ Q の頂点のうち，頂点 i に隣接していない頂点群 U である．頂点 i が Q に所属したことにより，U の頂点からコミュニティ Q に対する ΔQ が減少し，相対的に U から Q 以外への ΔQ が大きくなる．

そこで，上記 4 つのケースに当てはまる頂点のみ次回のフェーズ 1 での探索対象にすることにより，精度を保ちつつ高速化が望める．しかし複雑ネットワークの持つ次数分布がべき乗則に従う性質を考慮すると，大半の頂点の次数は小さいため，式(3)からすると ΔQ は頂点 i と，コミュニティ C 間のリンクの重みに最も影響を受けやすい．4 つのケースでは，直接頂点とコミュニティ間エッジの重みに影響を与えるのはケース1のみである．

以上のような考察に基づき，我々は Louvain-Prune 法を提案する．Louvain 法のフェーズ1において，所属コミュニティが変わった頂点 v の隣接頂点全てについて所属コミュニティを確認，v の新たな所属コミュニティに属さない頂点にマークを行い，次のフェーズ1における探索対象とすることで探索の枝刈りを行い，高速化を実現する．次節で本手法が実験的に効果的であることを示す．

4. 評価実験

Louvain 法と,提案手法を組み込んだ Louvain 法の比較実験を行った．データセットは予備実験で用いたものと同様のデータを使った．

計算機は CPU が 2.3GHz Intel core i7 , メモリが 16GB , OS が OS X Yosemite(10.10.1)のものを用いた．プログラムは Louvain 法の著者らが Web ページにて公開している C++のプログラムと,そのプログラムに提案手法を組み込んだものを利用した．図 4 に実験結果を示す．

Graph	Approach	Time(sec)	Modularity
Gowalla	OFF	2.52	0.707
	ON	1.39	0.708
DBLP	OFF	12.61	0.819
	ON	5.97	0.820
Amazon	OFF	4.85	0.926
	ON	2.42	0.926
Web-graph	OFF	19.83	0.977
	ON	8.81	0.977
Skitter	OFF	28.11	0.8322
	ON	13.53	0.8323
Youtube	OFF	17.65	0.7186
	ON	8.90	0.7184
Flickr	OFF	33.26	0.6349
	ON	16.26	0.6347

図 4. 従来手法と提案手法の比較

Modularity は実験に用いた全てのグラフにおいてほぼ同等の値を示している．一方計算時間においては比較的小規模なグラフである Gowalla を除く全てのグラフにおいて2倍以上の高速化に成功していることが確認できる．

図 5 は，Flickr の1回目のフェーズ1ループの Modularity と計算時間の推移を表したものである．1ループ目では提案手法によるオーバーヘッドにより若干計算時間が増えているが，2～16 ループ目では枝刈りにより大幅に計算時間が減りつつ，ほぼ同じ Modularity を保っていることが確認できる．

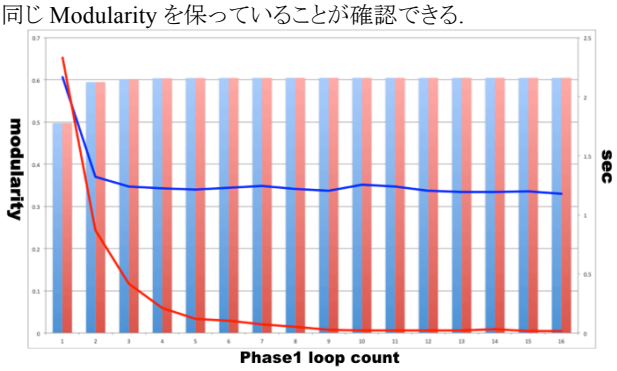

図 5. 提案手法の枝刈り効果

5. おわりに

本稿では高速かつ精度が高いとして広く知られている Louvain 法のフェーズ1において,探索処理に対してヒューリスティックな枝刈りをすることにより,精度を同程度に保ちつつ高速化する手法を提案した.

Louvain 法において 1 回目のフェーズ1は全体の処理時間のうち大半を占めるボトルネックであり,高速化の課題となっていた. 本稿ではフェーズ1をさらに細分化して考えることで, フェーズ1のループ中で全頂点を探索するのではなく,精度向上に寄与する頂点にのみ探索対象頂点を絞ることにより, 精度を同程度に保ちつつ 2 倍以上高速化できることを示した.

今後は本手法をさらに様々なグラフデータを用いて評価, 解析を行うことに加え, 並列分散化し, さらに大規模なグラフデータに対して効果的な手法であることを示す予定である.

参考文献

[Blondel 08] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre : Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, International School for Advanced Studies and IOP Publishing , July (2008)

[Shiokawa 13] H.Shiokawa, Y.Fujiwara and M.Onizuka : Fast Algorithm for Modularity-Based Graph Clustering, Proceedings of the Twenty-Seventh AAAI Comference on Artificial Intelligence(2013)

[Kuneigis 13] J.Kunegis : KONECT : the Koblenz network collection, Proceeding WWW '13 Companion Proceedings of the 22nd international conference on World Wide Web companion(2013)

[Clauset 04] A.Clauset, M.E.J.Newman and C.Moore : Finding community structure in very large networks, Phys. Rev. E 69(2004)

[Newman 04] M.E.J. Newman and M.Girvan : Finding and evaluating community structure in networks, Phys. Rev. E, 69(2004)

[Duch 05] J.Duch and A.Arenas : Community detection in complex networks using External Optimization, Phys. Rev. E 72

[Leskovec 14] J.Leskovec and A.Krevl : Standord Large Network Dataset Collection, http://snap.stanford.edu/data (2014)