基調構造を利用したグラフクラスタリングの高速化

塩川　浩昭† 天笠　俊之† 北川　博之†

† 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: †{shiokawa,amagasa,kitagawa}@cs.tsukuba.ac.jp

あらまし　モジュラリティクラスタリングは複雑な構造を持ったグラフから密な接続を持つクラスタを特定する手法であり，グラフ分析応用において重要な要素技術となっている．しかしながら，大規模なグラフを対象とした場合，モジュラリティクラスタリングには (1) クラスタの精度が著しく低下する，(2) クラスタリングに膨大な計算時間を要するという問題がある．これらの問題に対しこれまで様々な手法が提案されてきたが，依然として両問題を同時に解決する手法は存在しない．そこで本稿では，大規模なグラフに対して高速かつ高精度にクラスタを検出する手法 gScarf 法を提案する．gScarf 法では，既存の高精度なモジュラリティクラスタリング手法を基に，グラフの基調構造（トポロジー）を利用した高速化を行う．これにより，gScarf 法は高いクラスタリング精度を保ちつつ，高速なクラスタリングを大規模なグラフに対して実現する．本稿では実データならびに人工データを用いた評価実験を行い，近年提案された手法と比較して gScarf 法は高精度なクラスタを最大 1,100 倍程度高速に計算できることを確認した．

キーワード　グラフ，クラスタリング，モジュラリティ，高速化

1　は じ め に

本稿では大規模なグラフに対する高速かつ高精度なクラスタリング手法として gScarf 法 [1] を提案する．モジュラリティクラスタリング [2] は複雑な構造をもつグラフの中からノードが密に接続したクラスタを検出するためのアルゴリズムである．このクラスタリング手法はモジュラリティ [2] と呼ばれるクラスタリング指標に基づき，この指標を最大化するようなクラスタ集合をグラフの中から探索的に求める．一般的にモジュラリティを最大化するクラスタは良い精度を示すことが知られており [3,4]，モジュラリティクラスタリングはデータ工学分野や人工知能分野をはじめとする幅広い応用で利用されてきた [5,6]．

モジュラリティクラスタリングはこれまで幅広く利用されてきたが，大規模なグラフを対象とした際には次の 2 つの問題点が存在する．1 つ目の問題点はモジュラリティ解像度限界 [7] によるクラスタリング精度の低下である．モジュラリティクラスタリングはモジュラリティを最大化するクラスタを出力する．しかし Fortunato らの理論解析により，グラフのエッジ数を m とした時に，各クラスタが最低 $\sqrt{m}$ 本のエッジを含むまでモジュラリティが増加し続けることが明らかとなった [7]．すなわち，大規模グラフにおいては，モジュラリティクラスタリングは極めて粗い粒度のクラスタしか出力できず，結果としてクラスタリングの精度が低下することになる．2 つ目の問題点は膨大な計算コストを必要とすることである．近年の応用事例の多くでは数億から数十億エッジ規模のグラフを対象とする必要がある [5,8]．しかし，モジュラリティクラスタリングはグラフ内の全てのノードとエッジを反復計算する必要がある．その結果として，最新のモジュラリティクラスタリング手法 [9] を用いた場合でも，数十時間から数日程度の計算時間を必要とする．

1.1　既存研究と本研究の位置付け

上述の問題点を解決するために，これまで多くの手法が提案されてきた．その中でも主要なアプローチは，モジュラリティ解像度限界を回避する新たなクラスタリング指標を導入することである．例えば，局所性に基づくモジュラリティ指標 [10–12] は代表的な手法のひとつとして挙げられる．従来のモジュラリティはグラフ全体のエッジ接続密度とクラスタ内部の接続密度を比較して，クラスタの質を評価する．しかしこのような評価指標は，グラフが大規模になった際にどのようなクラスタのとり方をしても指標値が上昇させることとなり解像度限界をもたらす．これに対して，局所性に基づくモジュラリティ指標は，クラスタがその近隣の部分グラフと比較していかに密なエッジ接続密度を持つかを評価する．これにより，数万から数十万エッジ規模のグラフに対しては効果的に解像度限界を回避することができることが知られている．しかしながら，近年 Costa らはこれらの指標を用いた場合においても解像度限界が存在することを理論的に示した [13]．すなわち，局所性に基づく手法は中規模なグラフに対する一時的な精度改善効果しかなく，グラフが数億・数十億エッジ規模となったとき，依然としてクラスタリング精度が低下することを示唆している．

これに対し Duan らは，相関分析とモジュラリティを統合した尤度比相関モジュラリティ指標 (LRM) を提案した [14]．LRM では，従来指標のモジュラリティとアイテム相関分析 [15] を統合することで，解像度限界の要因となるバイアスをモジュラリティから取り除くことに成功した．Duan らは LRM を最大化するクラスタリング手法 [14]（以降，CorMod 法と記す）を提案し，大規模なグラフにおいても解像度限界を回避した精度の高いクラスタを検出できることを示した．しかしながら，CorMod 法は大規模なグラフのクラスタリングに膨大な計算時間を必要とする．従来手法と同様に，CorMod 法は LRM を最

大化するクラスタを検出するために，全てのノードとエッジを反復計算する必要がある．これはグラフのノード数とエッジ数をそれぞれ n，m とすると計算量が $O(nm \log n)$ となる．

モジュラリティクラスタリングの高速化に関する研究ではこれまで多くの手法が提案されてきている．ノード集約に基づく Louvain 法 [9] や IncMod 法 [16] が代表的な手法として挙げられる．しかし，これらの手法は従来の指標であるモジュラリティを最大化する処理を前提としており，LRM のような解像度限界問題を回避可能な指標に対しては適用することができない．我々の知る限り，高速性と解像度限界を回避した高精度なクラスタリングを両立するモジュラリティクラスタリング手法は存在しない．既存研究とは異なり，本研究は高速かつ高精度なモジュラリティクラスタリングを目指すものである．

1.2　本研究の貢献

本稿では高速かつ高精度なグラフクラスタリング手法 gScarf 法を提案する．gScarf 法は CorMod 法 [14] に基づきモジュラリティの解像度限界問題を回避し，数十億エッジ規模の大規模なグラフを高速にクラスタリングする手法である．gScarf 法はクラスタリング処理の高速化のために計算が不要な部分グラフを動的に特定し CorMod 法の計算過程から逐次的に除外する．

計算不要な部分グラフの特定を行うために，gScarf 法は LRM の決定性を利用する．2 節で詳細に述べるが，LRM は部分グラフのもつ基調構造（部分グラフのトポロジー）に対して一意に LRM の値が決まる性質がある．言い換えると，同一の基調構造を持つ部分グラフが複数存在した場合，全ての部分グラフは同一の LRM の値をもつ．すなわち，ある基調構造に対して LRM を一度計算すれば，同一の基調構造を持つ部分グラフに対する LRM の計算を行う必要はない．そこで gScarf 法ではこの性質を利用して，同一の基調構造を持つ部分グラフに対する計算を動的に枝刈りすることで計算コストの削減を目指す．

結果として提案手法 gScarf 法は次の性質を示す．

- **高速性**：gScarf 法は近年提案された手法と比較して最大で 1,100 倍高速である（4.2 節）．また，gScarf 法が既存手法より小さな計算量を持つことを証明した（定理 1）．
- **高精度性**：gScarf 法は既存手法 CorMod 法と同程度の精度を示す（4.3 節）．CorMod 法は解像度限界を回避する手法であるが，gScarf 法はその性質を継承することができる．
- **パラメータフリー**：gScarf 法は既存手法とは異なりパラメータを必要としない (Algorithm 1)．すなわち，既存手法と比較してより容易にクラスタリングを行うことが可能である．
- **再現性**：我々は提案手法 gScarf 法のソースコードを公開した [1]．すなわち，本研究の成果を再現・応用可能である．

我々の知る限り gScarf 法は数億エッジ規模のグラフに対して解像度限界を回避し高速なクラスタリングを実現した最初の手法である．例えば，14 億エッジ規模の Twitter データに対するクラスタリングを gScarf 法は 5 分未満で実行可能である．モジュラリティクラスタリングは様々な応用で利用されてきたが，計算時間と解像度限界が原因となり，大規模なグラフへの適用が難しかった．本研究を通じて高速かつ高精度なクラスタリングを可能とする gScarf 法を提案することにより，幅広い応用においてクラスタリングを利用可能となる．

2　事 前 準 備

本稿ではグラフ $G = (V, E, W)$ を考える．ただし，V，E および W はそれぞれノード集合，エッジ集合およびエッジの重みからなる集合である．すなわち，各エッジ $(i, j) \in E$ は重み $W_{i,j}$ を持ち，$W_{i,j} = 1$ で初期化されているものとする．グラフクラスタリングは G を重複のない部分グラフ（クラスタ）$C_i = (V_i, E_i)$ に分割する操作である．ただし，$V = \bigcup_i V_i$ であり，任意の $i \neq j$ に対して $V_i \cap V_j = \emptyset$ である．本稿では説明を簡単にするため無向グラフについて議論するが，提案手法は本質的に有向グラフの様な他のグラフモデルについても適用可能である．詳細は文献 [1] を参照されたい．

2.1　モジュラリティ Q

モジュラリティクラスタリングはクラスタリング指標モジュラリティQ [2] を最大化するクラスタ集合 $\mathbb{C}$ を求める．モジュラリティQ は各クラスタにおいて，クラスタ内に存在するエッジ数がその期待値と比較してどの程度多いかを定量化する．すなわち，各クラスタが密なエッジの接続を持つ場合，モジュラリティQ は大きくなる．

具体的な定義を次に示す．まず，クラスタ C_i 内に存在するエッジ数を e_i，クラスタ C_i に含まれる全ノードの次数和を a_i とする．このとき，$m = |E|$ とすると，クラスタ C_i 内に存在するエッジ数の割合は $tp(i) = e_i/2m$，クラスタ C_i 内に存在するエッジ数の割合の期待値は $ep(i) = (a_i/2m)^2$ となる．ゆえに，クラスタ集合 $\mathbb{C}$ に対するモジュラリティ$Q(\mathbb{C})$ の定義は

$$Q(\mathbb{C}) = \textstyle\sum_i Q(C_i) = \sum_i \{tp(i) - ep(i)\} \qquad (1)$$

となる．式 (1) に示すように，各クラスタ $C_i \in \mathbb{C}$ が期待値 $ep(i)$ よりも大きな $tp(i)$ を持つとき，モジュラリティQ は大きな値を示す．上述のように，モジュラリティクラスタリングでは式 (1) に示した指標を最大化するような $\mathbb{C}$ を探索的に求める．

しかしながら，近年の研究によってモジュラリティQ の最大化には解像度限界問題が指摘されている [7]．Fortunato と Barthelémy によって，式 (1) で示したモジュラリティ$Q(\mathbb{C})$ は各クラスタ $C_i \in \mathbb{C}$ が $e_i = \sqrt{2m}$ 本のエッジを含むまで増加し続けることを証明されている [7]．すなわち，グラフが極めて大規模なとき，モジュラリティQ の最大化に基づくモジュラリティクラスタリングは極めて粗粒度なクラスタを出力することになり，結果として精度が大きく低下する．

2.2　尤度比相関モジュラリティ LRM

解像度限界問題を解決するために，Duan らは CorMod 法と呼ばれる手法を提案した [14]．CorMod 法はモジュラリティQ の代わりに，相関分析とモジュラリティを統合した尤度比相関モジュラリティ (LRM) をクラスタリング指標として採用する．

1：https://github.com/LazyShion/gScarf

LRM の定義は次式のとおりである．

$$LRM(\mathbb{C}) = \sum_i LRM(C_i) = \sum_i \frac{Pr(tp(i), e_i, 2m)}{Pr(ep(i), e_i, 2m)}. \quad (2)$$

ただし $Pr(p,k,n)$ は確率質量関数 $Pr(p,k,n) = \binom{n}{k} p^k (1-p)^{n-k}$ である．すなわち，$Pr(tp(i), e_i, 2m)$ はグラフ G におけるクラスタ C_i の出現確率を示し，$Pr(ep(i), e_i, 2m)$ はランダムグラフにおいてクラスタ C_i が出現する確率を表す．式 (2) は，各クラスタに対して $Pr(tp(i), e_i, 2m)$ と $\frac{Pr(tp(i),e_i,2m)}{Pr(ep(i),e_i,2m)}$ をバランスさせるように設計されており，これにより各クラスタ C_i が粗粒度にならないように調整している．具体的には，C_i が小さい場合，クラスタ C_i が G において出現する確率 $Pr(tp(i), e_i, 2m)$ は大きくなる事が期待されるが，ランダムグラフにおいても同程度の出現確率を観測できるため $\frac{Pr(tp(i),e_i,2m)}{Pr(ep(i),e_i,2m)}$ は小さくなる．これに対して，C_i が多くのエッジを内包する場合，C_i の出現確率 $Pr(tp(i), e_i, 2m)$ は小さくなるが，$\frac{Pr(tp(i),e_i,2m)}{Pr(ep(i),e_i,2m)}$ は大きくなりやすい．CorMod では LRM を最大化することで，細粒度かつ密なエッジの接続を内包するクラスタ集合 $\mathbb{C}$ を出力する．

3 提案手法 gScarf 法

本稿では LRM を最大化するクラスタを高速に計算する gScarf 法を提案する．本節ではその詳細について述べる．

3.1 基本アイデア

本稿で提案する gScarf 法は先行研究 CorMod 法の精度を保ちつつ大規模グラフを高速に計算する手法である．CorMod 法は LRM を最大化するクラスタを見つけるために，全ての隣接するクラスタ対を同一クラスタに統合する場合を考慮し，LRM が最も向上するクラスタ対を貪欲的に同一クラスタに統合していく．しかしながら，この計算手順は大規模グラフに対して非常に大きな計算時間を必要とする．そこで提案手法 gScarf 法では計算が不必要な部分グラフを特定し，クラスタリング処理の過程から逐次的に除外する．まず，我々は LRM の差分計算式 LRM-gain を理論的に導出する（3.2 節）．この LRM-gain は LRM と同様に，クラスタ対が構成する基調構造に対してその値が決定的に定まる性質がある．そこで gScarf 法ではこの決定性を利用した *LRM-gain* キャッシング（3.3 節）を導入する．この手法では，既に計算したことのあるクラスタ対の基調構造とそれに対応する LRM-gain の値をキャッシュに保持する．これにより，同型なクラスタ対に対する LRM-gain の計算を高々 1 回に抑え，クラスタリング処理を高速化する．最後に，逐次部分グラフ集約（3.4 節）を用いて同一クラスタに含まれるノードに対する重複計算を省き，処理全体のさらなる高速化を図る．

gScarf 法を構成するこれらのアイデアは既存手法に対して次の優位性を持つ．(1) gScarf 法は実グラフを極めて高速にクラスタリング可能である．我々の LRM-gain キャッシングはグラフが持つ基調構造の種類が少ないほど計算速度が向上するが，実グラフがもつ基調構造の種類数は，次数分布がべき乗則に従うことから極めて少なくなることが知られている [17]．すなわち，次数分布に大きな偏りのある実グラフを gScarf 法は高速に処理できる．(2) gScarf 法はモジュラリティの解像度限界を回避する手法 CorMod 法 [14] のクラスタリング精度を損なわない．我々は次節以降にて，提案手法 gScarf 法を構成する各アイデアが LRM 最大化を行う CorMod と等価な処理を行うことを理論的に証明している．これにより，gScarf 法は本質的に CorMod 法と同程度のクラスタリング結果を出力する．

3.2 LRM の差分計算法: LRM-gain

クラスタ対を統合した際の LRM 変化量を次に定義する．

［定義 1］（LRM-gain $\triangle L_{i,j}$） クラスタ C_i と C_j を統合した際の LRM の変化量 (LRM-gain) $\triangle L_{i,j}$ を以下の式で定義する．

$$\triangle L_{i,j} = \triangle P_{i,j} - \triangle Q_{i,j}. \quad (3)$$

ただし，$\triangle P_{i,j}$ と $\triangle Q_{i,j}$ はそれぞれ確率比とモジュラリティの変化量であり，$C_{(i,j)}$ を C_i と C_j を統合してできたクラスタとすると，次式で与えられる．

$$\triangle P_{i,j} = P(C_{(i,j)}) - P(C_i) - P(C_j), \quad (4)$$

$$\triangle Q_{i,j} = Q(C_{(i,j)}) - Q(C_i) - Q(C_j). \quad (5)$$

ここで $tp(i) > 0$ のとき $P(C_i) = tp(i) \ln \frac{tp(i)}{ep(i)}$ であり，そうでない場合 $P(C_i) = 0$ である．

定義 1 は次の性質を満たす．

［補題 1］ クラスタ C_i と C_k を統合したクラスタを $C_{(i,k)}$，C_j と C_k を統合したクラスタを $C_{(j,k)}$ とするとき，$LRM(C_{(i,k)}) \geqq LRM(C_{(j,k)}) \iff \triangle L_{i,k} \geqq \triangle L_{j,k}$.

［証明 1］ 準備としてポアソンの極限定理 [18] を用いて式 (2) を変形する．具体的には，大規模グラフにおいて $tp(i)$ と $ep(i)$ は小さな値となり，$LRM(C_i)$ を以下のように変形できる．

$$LRM(C_i) = \frac{Pr(tp(i),e_i,2m)}{Pr(ep(i),e_i,2m)} = \frac{(2m \cdot tp(i))^{2m \cdot tp(i)} \cdot \mathrm{e}^{-2m \cdot tp(i)}}{(2m \cdot ep(i))^{2m \cdot tp(i)} \cdot \mathrm{e}^{-2m \cdot ep(i)}}$$
$$= \left(\frac{tp(i)}{ep(i)}\right)^{2m \cdot tp(i)} \cdot \mathrm{e}^{-2m \cdot Q(C_i)}. \quad (6)$$

ここで $L(C_i) = \frac{1}{2m} \ln LRM(C_i)$ とおくと次式が成り立つ．

$$L(C_i) = tp(i) \ln \frac{tp(i)}{ep(i)} - \{tp(i) - ep(i)\}. \quad (7)$$

このとき明らかに $\triangle L_{i,k} = L(C_{(i,k)}) - L(C_i) - L(C_k)$ である．

まず $LRM(C_{(i,k)}) \geqq LRM(C_{(j,k)}) \Rightarrow \triangle L_{i,k} \geqq \triangle L_{j,k}$ を示す．$LRM(C_{(i,k)}) \geqq LRM(C_{(j,k)})$ であるため，$LRM(C_{(i,k)}) - LRM(C_i) - LRM(C_k) \geqq LRM(C_{(j,k)}) - LRM(C_j) - LRM(C_k)$ は自明である．すなわち，$\frac{LRM(C_{(i,k)})}{LRM(C_i)LRM(C_k)} \geqq \frac{LRM(C_{(j,k)})}{LRM(C_j)LRM(C_k)}$ が成立する．ここで $L(C_i) = \frac{1}{2m} \ln LRM(C_i)$ を用いると，$L(C_{(i,k)}) - L(C_i) - L(C_k) \geqq L(C_{(j,k)}) - L(C_j) - L(C_k)$ と表すことができる．ゆえに，$LRM(C_{(i,k)}) \geqq LRM(C_{(j,k)}) \Rightarrow \triangle L_{i,k} \geqq \triangle L_{j,k}$ が成り立つ．

$\triangle L_{i,k} \geqq \triangle L_{j,k} \Rightarrow LRM(C_{(i,k)}) \geqq LRM(C_{(j,k)})$ についても同様に証明可能であるが，紙面の都合により省略する． □

補題 1 は LRM の最大化は LRM-gain の最大化と等価であることを示している．すなわち，gScarf 法は LRM を最大化するクラスタを LRM-gain を最大化することで見つけることができる．したがって，gScarf 法では LRM-gain を貪欲法的に最大化していくことを考えることとする．

補題 1 に加えて，次節以降で重要となる定義 1 の性質を示す．

［補題 2］ $\triangle L_{i,j}$ の値は 5 つの変数 e_i, a_i, e_j, a_j, および $e_{i,j}$ から決定的に定まる．ただし，$e_{i,j}$ はクラスタ C_i と C_j を接続するエッジ数の総和である．

［証明 2］ $tp((i,j)) = \frac{e_i+2e_{i,j}+e_j}{2m}$ および $ep((i,j)) = \left(\frac{a_i+a_j}{2m}\right)^2$ であるため，$\triangle P_{i,j}$ は e_i, a_i, e_j, a_j, および $e_{i,j}$ から一意に求まる．さらに，文献 [9] より $\triangle Q_{i,j} = 2\{\frac{e_{i,j}}{2m} - (\frac{a_i}{2m})(\frac{a_j}{2m})\}$ であるため，$\triangle Q_{i,j}$ も自明に a_i, a_j, および $e_{i,j}$ から一意に定まる．したがって, 定義 1 より補題 2 が成り立つ． □

［補題 3］ 任意のクラスタ対 $\langle C_i, C_j\rangle$ に対して, $\triangle L_{i,j}$ は $O(1)$ で計算できる．

［証明 3］ 補題 3 は定義 1 より自明である． □

3.3 LRM-gain キャッシング

計算対象となるノード・エッジを削減する手法 *LRM-gain* キャッシングを説明する．

補題 2 で示したように，LRM-gain $\triangle L_{i,j}$ はクラスタ対を構成する基調構造 $s_{i,j} = \langle e_i, a_i, e_j, a_j, e_{i,j}\rangle$ から一意に定まる．すなわち，$s_{i,j}$ と同型な $s_{i',j'}$ があるとき，$\triangle L_{i,j} = \triangle L_{i',j'}$ が成り立つ．そこで gScarf 法では，一度基調構造 $s_{i,j}$ とそれに対応する $\triangle L_{i,j}$ を計算した場合，それらをキャッシュ領域に保持する．その後，$s_{i,j}$ と同型なクラスタ対に対する LRM-gain の計算が生じた場合，そのクラスタ対を計算はせず，既にキャッシングした結果を再利用する．

より具体的に *LRM-gain* キャッシングを以下に定義する．

［定義 2］（LRM-gain キャッシング） h を $s_{u,v}$ をキーとし対応する $\triangle L_{u,v}$ を値に持つハッシュ関数とする．LRM-gain キャッシング $h(s_{i,j})$ はクラスタ対 $\langle C_i, C_j\rangle$ を成す基調構造 $s_{i,j} = \langle e_i, a_i, e_j, a_j, e_{i,j}\rangle$ について以下のように定義する．

$$h(s_{i,j}) = \begin{cases} \triangle L_{i',j'} & (s_{i',j'} \equiv s_{i,j}\text{となる } s_{i',j'} \text{ がhに存在}), \\ \text{null} & (\text{上記以外}). \end{cases} \quad (8)$$

$s_{i',j'} \equiv s_{i,j}$ は基調構造 $s_{i',j'}$ が $s_{i,j}$ と同型であることを示す．gScarf 法は $s_{i',j'} \equiv s_{i,j}$ となる $s_{i',j'}$ をハッシュ関数 h 内に見つけた場合，$\triangle L_{i,j}$ を h から呼び出し計算を省略する．それ以外の場合は，$\triangle L_{i,j}$ を定義 1 に従い計算する．

定義 2 の理論的な側面を議論するために，実グラフの多くが一般的にもつ性質であるべき乗則に従う次数分布 [17] を導入する．実グラフの次数分布は一般にべき乗則に従うことが知られている．すなわち，この性質の下では，次数 k となるノードの出現確率 $p(k)$ は $k^{-\gamma}$ に比例する．ただし，γ は次数分布の偏りの強さを表す小さな正数である．

グラフの次数分布がべき乗則に従うことを仮定したとき，LRM-gain キャッシングは次の性質を満たす．

［補題 4］ d をグラフの平均次数としたとき，LRM-gain キャッシングはクラスタリング処理全体において $O(d^{2\gamma})$ の時間・空間計算量を必要とする．

［証明 4］ 議論を簡単にするため，すべてのノード $i \in V$ について単純なクラスタ $C_i = \{i\}$ を考える．このとき，$e_i = e_j = 0$ と $e_{i,j} = 1$ が常に成り立つため，$a_i = a_{i'}$ かつ $a_j = a_{j'}$ であれば，$s_{i,j} \equiv s_{i',j'}$ である．すなわち，$s_{i,j}$ の構造は a_i と a_j の値によって一意に定まる．仮定から次数 k のノードの出現確率は $p(k) \propto k^{-\gamma}$ であるため，グラフ G 内に存在する基調構造が $s_{i,j}$ となるクラスタ対の数の期待値は $2m \cdot p(a_i)p(a_j)$ である．したがって，グラフ G 全体を構成するために必要な基調構造の種類数は $O(\frac{2m}{2m \cdot p(a_i)p(a_j)}) \approx O(\frac{1}{p(d)^2})$ である．言い換えると，$O(\frac{1}{p(d)^2})$ 種類の基調構造のみ LRM-gain を計算すればグラフ全体をクラスタリング処理できる．補題 3 より，LRM-gain の計算は $O(1)$ で求まる．ゆえに，LRM-gain キャッシングの時間・空間計算量は $O(\frac{1}{p(d)^2}) = O(d^{2\gamma})$ となる． □

一般的に実グラフの平均次数 d と γ は極めて小さくなることが知られている [16]．すなわち，補題 4 は LRM-gain caching により極めて少ない数の基調構造のみを計算すれば，グラフ全体を計算できることを示している．より詳細な解析については，4 節で実験を通じて議論する．

3.4 逐次部分グラフ集約

gScarf 法をさらに高速化するため，逐次部分グラフ集約を導入する．実グラフは高いクラスタ性を示し 3 部クリーク構造を多く含むことが知られているが [19]，これらの構造に対して CorMod 法などの貪欲法に基づく LRM の最大化は重複した計算が生じる．具体的には，3 部クリーク構造は同一クラスタに対して複数本の重複したエッジを張るが，CorMod 法はそれらすべてのエッジを計算対象としてしまう．このような重複したエッジに対する計算を除外するため，gScarf 法は Shiokawa らにより提案された逐次部分グラフ集約法 [16] を本研究の対象とするクラスタリング問題に拡張する．

まず，部分グラフ集約を次のように定義する．

［定義 3］（部分グラフ集約） グラフ $G = (V, E, W)$ のクラスタ $C_i = (V_i, E_i)$ と $C_j = (V_j, E_j)$ に属するノードをそれぞれ $i \in C_i$，$j \in C_j$ とする．ここで $V \backslash \{V_i \cup V_j\}$ に属するノードをそのノード自身もしくは新たなノード x に射影する関数 f を考える．このとき，ノード i と j に対する部分グラフ集約は新たなグラフ $G' = (V', E', W')$ を構築する手続きである．ただし，$V' = V \backslash \{V_i \cup V_j\} \cup \{x\}$ かつ $E' = \{(f(u), f(v)) | (u,v) \in E\}$ であり，各エッジの重み $W'_{f(u),f(v)}$ は次の通りである．

$$W'_{f(u),f(v)} = \begin{cases} 2W_{u,v} + W_{u,u} + W_{v,v} & (f(u) = x, f(v) = x) \\ W_{i,v} + W_{j,v} & (f(u) = x, f(v) \neq x) \\ W_{u,i} + W_{u,j} & (f(u) \neq x, f(v) = x) \\ W_{u,v} & (f(u) \neq x, f(v) \neq x) \end{cases}.$$

定義 3 より，部分グラフ集約は 2 つのノード i と j を重み付きエッジを用いて等価な 1 つのノード x に変換する操作である．部分グラフ集約はノード i，j を接続するエッジの本数を重み $W_{x,x}$ とする自己ループエッジ $(x,x) \in E'$ を新たなノード x に与える．同様に，ノード x の隣接ノード $k \in \Gamma(x)$ に対して，部分グラフ集約は元のグラフ G が持つエッジ (i,k) と (j,k) を 1 つの重み付きエッジ (x,k) に変換する．これにより，部分グラフ集約はグラフ内のノートとエッジを削減する．

［補題 5］ ノード $i, j \in V$ が同一クラスタに所属するとき，

Algorithm 1 提案手法 gScarf 法

Input: A graph $G = (V, E, W)$;
Output: A set of clusters $\mathbb{C}$;
1: $\mathbb{T} = \emptyset$;
2: **for each** $i \in V$ **do**
3: $C_i = \{i\}$, and $\mathbb{T} = \mathbb{T} \cup \{C_i\}$;
4: **while** $\mathbb{T} \neq \emptyset$ **do**
5: Get C_i from $\mathbb{T}$;
6: $C_{best} = C_i$;
7: **for** $C_j \in \Gamma(C_i)$ **do**
8: $s_{i,j} = \langle e_i, a_i, e_j, a_j, e_{i,j} \rangle$;
9: **if** $h(s_{i,j}) = null$ **then** $h(s_{i,j}) \leftarrow \triangle L_{i,j}$;
10: **if** $h(s_{i,j}) > h(s_{i,best})$ **then** $C_{best} = C_j$;
11: **if** $h(s_{i,best}) > 0$ **then**
12: $C' \leftarrow fold(C_i, C_{best})$ by Definition 3;
13: $\mathbb{T} = \mathbb{T} \backslash \{C_i, C_{best}\} \cup \{C'\}$;
14: **else** $\mathbb{T} = \mathbb{T} \backslash \{C_i\}$;
15: **return** $\mathbb{C}$;

定義 3 によりノード i, j を新たなノード w に集約した場合，$LRM(C_w) = LRM(C_{(i,j)})$ が成り立つ．

［証明 5］ 式 (7) より $L(C_w) = tp(w) \ln \frac{tp(w)}{ep(w)} - \{tp(w) - ep(w)\}$ である．定義 3 より自明に $e_w = e_i + e_j + 2e_{i,j}$ かつ $a_w = a_i + a_j$ であるため，

$$\begin{aligned} L(C_w) &= tp(w) \ln \tfrac{tp(w)}{ep(w)} - \{tp(w) - ep(w)\} \\ &= \tfrac{e_i+e_j+2e_{i,j}}{2m} \ln \frac{\frac{e_i+e_j+2e_{i,j}}{2m}}{\left(\frac{a_i+a_j}{2m}\right)^2} - \tfrac{e_i+e_j+2e_{i,j}}{2m} + \left(\tfrac{a_i+a_j}{2m}\right)^2 \\ &= \tfrac{e_{(i,j)}}{2m} \ln \frac{\frac{e_{i,j}}{2m}}{\left(\frac{a_{(i,j)}}{2m}\right)^2} - \tfrac{e_{(i,j)}}{2m} + \left(\tfrac{a_{(i,j)}}{2m}\right)^2 = L(C_{(i,j)}). \quad (9) \end{aligned}$$

補題 1 より，$L(C_w) = L(C_{(i,j)})$ のとき $LRM(C_w) = LRM(C_{(i,j)})$ であり，式 (9) から補題 5 が成立する． □

補題 5 より，部分グラフ集約は LRM-gain を正確に計算可能であることが言える．ゆえに，gScarf 法は CorMod の精度を損なわずにノード・エッジ数を削減することができる．

定義 3 に基づき，gScarf 法は逐次的に部分グラフ集約を実行する．gScarf 法は任意のクラスタ C_i を選択し，そのすべての隣接のクラスタに対して定義 1 に示した LRM-gain$\triangle L_{i,j}$ を計算する．このとき計算した LRM-gain の中で，最大の正数となったクラスタ対 $\langle C_i, C_j \rangle$ に対して，即座に定義 3 で示した部分グラフ集約を適用する．gScarf 法はこれを収束まで繰り返す．

本節で述べた逐次部分グラフ集約は次の理論的な性質を持つ．

［補題 6］ 平均次数 d のグラフに対して，部分グラフ集約（定義 3）の時間・空間計算量は $O(d)$ となる．

［証明 6］ 定義 3 より，部分グラフ集約はあるクラスタ C_i に隣接するすべてのクラスタに対して更新処理を行う．この処理は，クラスタ対 $\langle C_i, C_j \rangle$ に対して明らかに $O(\min\{|\Gamma(C_i)|, |\Gamma(C_j)|\}) = O(d)$ を必要とする． □

3.5 gScarf 法のアルゴリズム

gScarf 法のアルゴリズムを Algorithm 1 に示す．まず gScarf 法は各ノードを $C_i = \{i\}$ となるクラスタとし，ターゲットノード集合 $\mathbb{T}$ に格納する（1-3 行目）．次に gScarf 法はクラスタリングを開始する（4-15 行目）．gScarf 法は任意のクラスタ C_i を $\mathbb{T}$ から選択し（5 行目），その隣接クラスタの中から最大の正数となる LRM-gain を持つ C_{best} を見つける（6-10 行目）．計算を高速化するため，この計算には定義 2 で述べた LRM-gain キャッシングを行い，既に計算したことのあるクラスタ対と同型な基調構造を持つクラスタ対の計算を除外する（8-10 行目）．gScarf 法は $s_{i',j'} \equiv s_{i,j}$ となる基調構造を関数 h 内に見つけた場合，$h(s_{i',j'}) = \triangle L_{i',j'}$ を LRM-gain の計算結果として再利用する．それ以外の場合は，$\triangle L_{i,j}$ を定義 1 に従い計算する（9 行目）．その後，gScarf 法は定義 3 に基づき部分グラフ集約をクラスタ対 $\langle C_i, C_{best} \rangle$ に対して行う（11-14 行目）．上述の処理を繰り返し，$\mathbb{T} = \emptyset$ となったとき gScarf 法は停止する．

最後に gScarf 法の計算量を解析する．

［定理 1］ gScarf 法の計算量は $O(m + d^{2\gamma})$ である．ただし，m はグラフ G の総エッジ数，d は平均次数，γ は次数分布の偏りの強さを表す小さな正数である．

［証明 7］ Algorithm 1 より gScarf 法は $O(|\mathbb{T}|) \approx O(n)$ 回の反復計算を行う．各反復において，gScarf 法は高々1 回の部分グラフ集約を実行し，これは補題 6 より $O(d)$ の計算量を要する．すなわち，gScarf 法の計算量は全体で $O(nd) = O(m)$ となる．さらに，gScarf 法は LRM-gain キャッシングを各反復で実行する．補題 4 で証明したように，これはグラフ全体で $O(d^{2\gamma})$ の計算量を要する．ゆえに計算量は $O(m + d^{2\gamma})$ となる． □

1 節で述べた通り CorMod 法の計算量は $O(nm \log n)$ であり，提案手法 gScarf 法は CorMod 法よりも計算量が小さいことがわかる．特に，実グラフでは d と γ が極めて小さくなるとなることが知られており [17]，この場合 $d^{2\gamma} \ll m$ となる．具体的には，本稿で評価に用いた実グラフ（表 1）では d と γ は高々$d < 39$ and $\gamma < 2.3$ となり，エッジ数と比較して極めて小さな値であることがわかる．その結果として gScarf 法の計算量は $O(m + d^{2\gamma}) \approx O(m)$ となり，実グラフ上ではグラフの規模に対してほぼ線形の計算コストとなる．加えて，gScarf 法は部分グラフ集約によるさらなる効率化を行う．部分グラフ集約はクラスタ性の高い実グラフにおいて効果的であることがわかっており [19, 20]，実際の gScarf 法の計算コストは定理 1 よりもさらに小さくなることが期待される．

表 1 実グラフデータセットの概要

Name	n	m	d	γ	Ground-truth	Source
YT	1.13 M	2.98 M	2.63	1.93	✓	com-Youtube [24]
WK	1.01 M	25.6 M	25.1	2.02	N/A	itwiki-2013 [25]
LJ	3.99 M	34.6 M	8.67	2.29	✓	com-LiveJournal [24]
OK	3.07 M	117 M	38.1	1.89	✓	com-Orkut [24]
WB	118 M	1.01 B	8.63	2.14	N/A	webbase-2001 [25]
TW	41.6 M	1.46 B	35.2	2.27	N/A	twitter-2010 [25]

4 評価実験

本節では gScarf 法の実行速度と精度を実験的に評価する．

4.1 実験設定

比較手法: 我々は gScarf 法を以下の最新の手法と比較する．

- **CorMod [14]**: 尤度比相関に基づくモジュラリティを用いたクラスタリング手法である．提案手法 gScarf 法と同様に LRM を最大化するクラスタを貪欲法により検出する．
- **Louvain [9]**: 最も標準的なモジュラリティクラスタリング手法である．この手法は式 (1) に示したモジュラリティQ を貪欲法により最大化したクラスタを検出する．

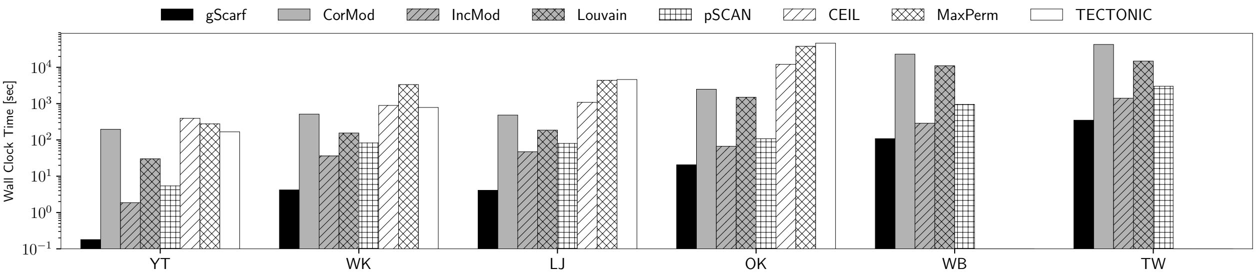


図 1　実グラフデータセットにおける実行時間の比較

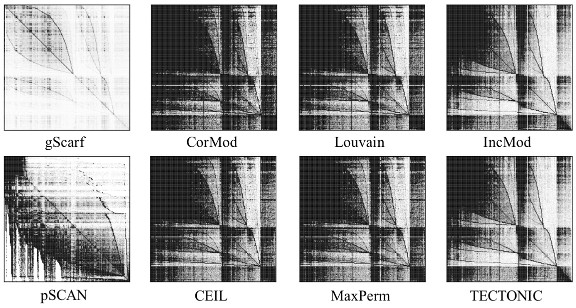


図 2　計算されたエッジ数の比較（データセット：YT）

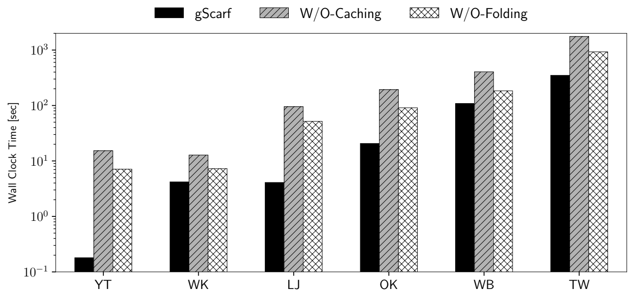


図 3　各高速化手法の効率比較

- **IncMod [16]**: Louvain 法の高速化手法であり，式 (1) に示したモジュラリティQ を貪欲法により最大化する．我々の知る限り最も高速なモジュラリティクラスタリング手法である．
- **pSCAN [21]**: 密度ベースのクラスタリング手法である．密な接続を持つクラスタをしきい値 ϵ と μ を用いて探索する．我々は文献 [21] に従い $\epsilon = 0.6$ および $\mu = 5$ とした．
- **CEIL [12]**: クラスタの内部と外部の接続密度比を考慮することで解像度限界を解消したクラスタリング手法である．
- **MaxPerm [22]**: クラスタの頑健性に着眼することで，解像度限界問題をを解消したクラスタリング手法である．
- **TECTONIC [23]**: モチーフベースのグラフクラスタリング手法である．この手法はしきい値 θ より少ない割合の 3 部クリークに属するエッジをトップダウンに削除することでクラスタを検出する．我々は文献 [23] に従い $\theta = 0.06$ とした．

全ての実験は Intel Xeon E5-2690 CPU 2.60 GHz，128 GB RAM を搭載した Linux サーバ上で実行し，全てのアルゴリズムは C/C++を用いてシングルスレッドプログラムとして実装し，`g++9.2.0` にてオプション`-O3` を用いてコンパイルした．

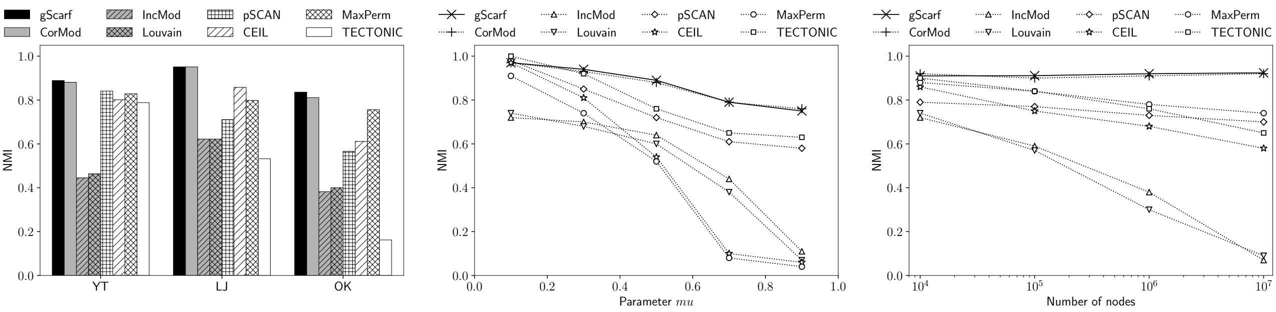
データセット: 本稿では SNAP [24] および LAW [25] で公開されている 6 つの実グラフデータセットを用いる．表 1 に詳細を示す．表の d と γ は各グラフの平均次数とべき乗則に従う次数分布の偏りの強さを示している．表 1 に示したとおり，YT，LJ，ならびに OK のみ ground-truth となるクラスタリング結果が存在している．我々の実験では LFR-benchmark [26] により生成した人工グラフとその ground-truth となるクラスタリング結果も評価に用いる．LFR-benchmark で生成した人工グラフの詳細な設定については 4.3 節で述べる．

4.2　高速性の評価

図 1 にて各手法の実グラフに対するクラスタリング処理を比較する．CEIL，MaxPerm，および TECTONIC は大規模グラフにおいて 24 時間以内に処理が終了しなかったため図 1 から除外した．この結果より，全ての条件下において gScarf 法は他の手法よりも高速であることが確認できる．平均すると gScarf 法は他の手法よりも 273.7 倍高速である．また，gScarf 法は LRM 最大化手法である CorMod 法と比較して最大で 1,100 倍程度高速である．gScarf 法は LRM-gain キャッシングにより，一部のノードとエッジのみを計算することでクラスタリング処理が可能である．さらに，部分グラフ集約によって，クラスタ性の高い部分グラフから誘導される重複した計算を除外する．その結果として，定理 1 で証明したように，グラフの規模に対して概ね線形の計算時間で処理ができる．ゆえに，gScarf 法は他の手法と比較して大幅に高速な処理が可能である．

gScarf 法の効率性を検証するために，図 2 において計算されたエッジ数を比較する．図 2 は YT データセットにおいて各アルゴリズムが計算したエッジを描画した 2 次元ヒストグラム（隣接行列）である．エッジ (i, j) が計算された場合はヒストグラムの (i, j) 要素を黒く描画している．それ以外の場合は対応する要素を白く描画している．図 2 から分かるように，gScarf 法は他の手法と比較して計算されたエッジ数を大幅に削減していることがわかる．具体的には，他の手法と比較して gScarf 法は 83.9〜98.3%少ないエッジの計算回数でクラスタリング処理を完了することができる．この結果は我々の提案アプローチである LRM-gain キャッシングと部分グラフ集約がクラスタリングの高速化に効果的であることを示唆している．表 1 に示したように，YT データセットを含む実データセットの次数分布はべき乗則に従っている．このようなグラフでは補題 4 で証明したように，LRM-gain キャッシングは極めて少ない数の基調構造を計算するだけでグラフ全体を計算することができる．ゆえに，gScarf 法は計算エッジ数を削減することができる．

最後に本稿で提案した LRM-gain キャッシングと部分グラフ集約の効果について検証する．図 3 では提案手法 gScarf 法と



(a) 実グラフデータセット　(b) LFR benchmark （mu を変化させた場合）　(c) LFR benchmark（n を変化させた場合）

図 4　実グラフおよび人工グラフにおける NMI 値の比較

表 2　平均クラスタサイズの比較

	Ground-truth	gScarf	CorMod	Louvain	IncMod	pSCAN	CEIL	MaxPerm	TECTONIC
YT	13.5	13.3	13.3	66.1	50.4	24.3	5.6	11.4	8.2
LJ	40.6	44.2	45.1	111.4	104.5	81.9	11.3	33.3	10.7
OK	215.7	194.9	194.1	16551.9	15676.7	403.7	35.7	121.9	9.61

gScarf 法から LRM-gain キャッシングを除いた *W/O-Caching* および gScarf 法から部分グラフ集約を除いた *W/O-Folding* を比較する．図 3 からわかるように，gScarf 法は W/O-Caching と W/O-Folding はそれぞれ平均して 21.6 倍，10.4 倍高速である．この結果は LRM-gain キャッシングが部分グラフ集約よりも高速化に大きく貢献していることを示唆している．3.5 節で述べたように，実グラフは一般的に小さな d と γ の値を持つことが知られている．例えば，表 1 に示したとおり，本稿で用いたデータセットでも d と γ の値はそれぞれ高々$d \leqq 38.1$ and $\gamma < 2.3$ である．その結果として，LRM-gain キャッシングは $O(d^{2\gamma})$ の計算量を必要とするものの，実グラフにおいては部分グラフ集約と比較して十分に小さな計算コストになる．ゆえに，gScarf 法は効果的に計算時間を削減することができる．

4.3　クラスタリング精度の評価

gScarf 法の貢献点のひとつは，大規模なグラフを高速に計算しつつ，解像度限界を回避した CorMod 法と比較して同程度のクラスタリング精度を示すことにある．gScarf 法の精度を評価するために，本稿では実グラフならびに人工グラフの ground-truth となるクラスタリング結果に対するクラスタリング結果の質を評価する．精度評価のために，本稿では正規化相互情報量 (NMI) [27] を用いる．本稿では各手法と ground-truth となるクラスタリング結果を用いて NMI を評価する．ゆえに NMI は ground-truth に近いクラスタリング結果を出力したとき 1 に近づき，そうでない場合は 0 に近づく．

4.3.1　実グラフにおける評価

図 4 (a) は実グラフデータセット YT，LJ，および OK における NMI の値を比較した結果である．本稿では SNAP で公開されている `top-5000-community` データセット [24] を ground-truth のクラスタリング結果として使用した．このデータセットではノードが複数のクラスタに所属する場合がある．そこで，そのようなノードは隣接ノードの中で最も多くのノードが属するクラスタと同一のクラスタに所属するものとして扱った．

図 4 に示すとおり，gScarf 法は全てのデータセットにおいて最も高い NMI の値を示していることがわかる．加えて，gScarf 法と同様に LRM の最大化を行う CorMod 法と比較して，gScarf 法は概ね同程度の NMI の値を示している．この結果は gScarf 法が CorMod 法と同様に他の手法と比較して極めて高い精度でクラスタを検出できていることを示唆している．これは LRM 最大化が既存のモジュラリティクラスタリングの解像度限界問題を回避できていることにある．また，gScarf 法は補題 1 や補題 5 にて理論的に示したように，LRM を損なわないように計算コストの削減を行うことができる．したがって，LRM-gain の最大化を行う gScarf 法は CorMod 法と同程度のクラスタリング結果を出力することが可能である．ゆえに，gScarf 法は LRM-gain キャッシングや部分グラフ集約により高速にクラスタリングを実行しつつ，既存の手法よりも高い NMI 値を示すクラスタリング結果を獲得することが可能である．

クラスタの解像度をより詳細に分析するために，各手法が出力したクラスタの平均ノード数を表 2 に示す．表 2 からわかるように，gScarf 法と CorMod 法は大規模なグラフにおいて ground-truth と同程度の大きさのクラスタを検出できている．これは上述したとおり，LRM 最大化がモジュラリティの解像度限界を効果的に回避できることによる結果である [14]．これに対して，他の手法は gScarf 法や CorMod 法よりも極めて粒度の粗いクラスタや極めて小さなクラスタのみを出力していることが確認できる．特にモジュラリティ最大化を行う Louvain 法や IncMod 法は，解像度限界によりクラスタが $\sqrt{2m}$ 本のエッジを含むまでモジュラリティQ が増加し続けるため，結果として ground-truth よりも大きなクラスタを出力することになる．

4.3.2　人工グラフにおける評価

次に人工グラフを用いて，グラフの特性に応じたクラスタリング精度の検証を行う．図 4 (b) ではノード数 10^6，平均次数 $d = 20$ のグラフに対して，混合パラメータ mu を 0.1 から 0.9 まで変化させたグラフに対する NMI の比較を行う．混合パラメータ mu は隣接するクラスタ間にエッジを張る確率を調整するパラメータである．mu を大きくするとクラスタ間の接続が密となるため，一般的にクラスタを検出することが難しくなる．これに対して図 4 (c) ではグラフの規模に対するクラスタリング精度の評価を行う．図 4 では，平均次数と混合パラメータをそれぞれ $d = 20$，$mu = 0.5$ に固定し，ノード数を 10^4 から 10^7 まで変化させたグラフに対して NMI の比較を行う．

図 4 (b) より，gScarf 法は混合パラメータ *mu* を変化させた場合においても他の手法と比較して高い NMI の値を示していることが確認できる．また，図 4 (c) に示すように，gScarf 法はグラフのサイズに依存せず他の手法よりも高い NMI の値を出力している．これらの図は，gScarf 法は ground-truth を高精度に近似したクラスタリング結果を大規模なグラフにおいても出力できることを示唆している．これらの結果を通じて，gScarf 法は CorMod 法と比較してクラスタリング精度を犠牲にせず，高速にクラスタリングすることができることを確認した．

5 おわりに

本稿では大規模グラフに対する高速・高精度なクラスタリング手法 gScarf 法を提案した．gScarf 法はグラフの基調構造に着眼して重複計算を除外することによりクラスタリング処理を大幅に高速化する．我々は実グラフならびに人工グラフを通じた評価実験を行い，gScarf 法は最大で 1,100 倍の高速化を実現しつつ，最新の手法と同程度のクラスタリング精度を出力することを確認した．グラフフラスタリングは幅広い応用において基本的かつ重要な要素技術である．本稿を通じて高速・高精度な gScarf 法を提供することで，将来開発される多くのアプリケーションの性能・機能向上に大きく寄与することができる．

謝　　辞

本研究の一部は JST ACT-I ならびに科研費 若手研究 (18K18057) の支援を受けたものである．

文　　献

[1] Hiroaki Shiokawa, Toshiyuki Amagasa, and Hiroyuki Kitagawa. Scaling Fine-grained Modularity Clustering for Massive Graphs. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 4597–4604, 2019.

[2] M. E. J. Newman. Fast Algorithm for Detecting Community Structure in Networks. *Physical Review*, E 69(066133), 2004.

[3] Tomokatsu Takahashi, Hiroaki Shiokawa, and Hiroyuki Kitagawa. SCAN-XP: Parallel Structural Graph Clustering Algorithm on Intel Xeon Phi Coprocessors. In *Proc. ACM SIGMOD Workshop on Network Data Analytics*, pages 6:1–6:7, 2017.

[4] Tomoki Sato, Hiroaki Shiokawa, Yuto Yamaguchi, and Hiroyuki Kitagawa. FORank: Fast ObjectRank for Large Heterogeneous Graphs. In *Companion Proceedings of The Web Conference 2018*, pages 103–104, 2018.

[5] Alireza Louni and K. P. Subbalakshmi. Who Spread That Rumor: Finding the Source of Information in Large Online Social Networks with Probabilistically Varying Internode Relationship Strengths. *IEEE Transactions on Computational Social Systems*, 5(2):335–343, June 2018.

[6] Hiroaki Shiokawa, Tomokatsu Takahashi, and Hiroyuki Kitagawa. ScaleSCAN: Scalable Density-based Graph Clustering. In *Proc. DEXA 2018*, pages 18–34, 2018.

[7] Santo Fortunato and M Barthélemy. Resolution Limit in Community Detection. *Proceedings of the National Academy of Sciences of United States of America*, 104(1):36–41, Jan 2007.

[8] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society. *Nature*, 435(7043):814–818, June 2005.

[9] V.D. Blondel, J.L. Guillaume, R. Lambiotte, and E.L.J.S. Mech. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[10] Stefanie Muff, Francesco Rao, and Amedeo Caflisch. Local Modularity Measure for Network Clusterizations. *Physical Review*, E 72(056107), 2005.

[11] Zhenping Li, Shihua Zhang, Rui-Sheng Wang, Xiang-Sun Zhang, and Luonan Chen. Quantative Function for Community Detection. *Physical Review*, E 77(036109), 2008.

[12] M. Vishnu Sankar, Balaraman Ravindran, and S Shivashankar. CEIL: A Scalable, Resolution Limit Free Approach for Detecting Communities in Large Networks. In *Proc. IJCAI 2015*, pages 2097–2103, 2015.

[13] Alberto Costa. Comment on ”Quantitative Function for Community Detection”. *CoRR*, abs/1409.4063, 2014.

[14] Lian Duan, Willian Nick Street, Yanchi Liu, and Haibing Lu. Community Detection in Graphs Through Correlation. In *Proc. KDD 2014*, pages 1376–1385, 2014.

[15] Gregory Piatetsky-Shapiro. Discovery, Analysis, and Presentation of Strong Rules. *Knowledge Discovery in Databases*, pages 229–248, 1991.

[16] Hiroaki Shiokawa, Yasuhiro Fujiwara, and Makoto Onizuka. Fast Algorithm for Modularity-based Graph Clustering. In *Proc. AAAI 2013*, pages 1170–1176, 2013.

[17] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On Power-law Relationships of the Internet Topology. In *Proc. SIGCOMM 1999*, pages 251–262, 1999.

[18] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill Higher Education, 4 edition, 2002.

[19] Duncan J. Watts and Steven H. Strogatz. Collective Dynamics of ’Small-World’ Networks. *Nature*, 393(6684):440–442, 1998.

[20] Hiroaki Shiokawa, Yasuhiro Fujiwara, and Makoto Onizuka. SCAN++: Efficient Algorithm for Finding Clusters, Hubs and Outliers on Large-scale Graphs. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 8(11):1178–1189, July 2015.

[21] Lijun Chang, Wei Li, Lu Qin, Wenjie Zhang, and Shiyu Yang. pSCAN: Fast and Exact Structural Graph Clustering. *IEEE Transaction on Knowledge Data Engineering*, 29(2):387–401, 2017.

[22] Tanmoy Chakraborty, Sriram Srinivasan, Niloy Ganguly, Animesh Mukherjee, and Sanjukta Bhowmick. On the Permanence of Vertices in Network Communities. In *Proc. KDD 2014*, pages 1396–1405, 2014.

[23] Charalampos E. Tsourakakis, Jakub Pachocki, and Michael Mitzenmacher. Scalable Motif-aware Graph Clustering. In *Proc. WWW 2017*, pages 1451–1460, 2017.

[24] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford Large Network Dataset Collection. `http://snap.stanford.edu/data`, jun 2014.

[25] Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. Layered Label Propagation: A MultiResolution Coordinate-Free Ordering for Compressing Social Networks. In *Proc. WWW 2011*, pages 587–596, 2011.

[26] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the Overlapping and Hierarchical Community Structure in Complex Networks. *New Journal of Physics*, 11(3):033015, 2009.

[27] Rudi Cilibrasi and Paul MB Vitányi. Clustering by Compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.