構造的類似度に基づくグラフクラスタリングの高速化

塩川　浩昭† 藤原　靖宏† 鬼塚　真†

† 日本電信電話株式会社 NTT ソフトウェアイノベーションセンタ
〒 180–8585 東京都武蔵野市緑町 3–9–11
E-mail: †{shiokawa.hiroaki,fujiwara.yasuhiro,onizuka.makoto}@lab.ntt.co.jp

あらまし グラフクラスタ分析はグラフの中に存在するコミュニティ構造を理解する上で重要な要素技術である．その中でもノード間の構造的類似度を用いたクラスタリング手法 SCAN は，グラフ中のクラスタを抽出するだけでなく，ハブや外れ値などのノードも併せて抽出可能な手法として知られている．しかしながら，SCAN は全てのエッジに対する計算を行うため，グラフに含まれるエッジ数を $|E|$ とした時に $O(|E|)$ の計算量を要する．この SCAN の計算量は，グラフに含まれるノード数を $|V|$ とした時に，最悪の場合 $|E| \approx |V|^2$ となることから最悪計算量が $O(|V|^2)$ となり，大規模なグラフへの適用が難しい．本稿では SCAN の高速化手法を提案する．提案手法では，最短ホップ数が 2 となる様なノードに接続したエッジのみを計算対象としてクラスタリングを行う．これにより，提案手法は SCAN と同一の結果をより高速に抽出ことを可能にする．本稿では，実データに対する評価実験を行い，SCAN の計算時間を最大で約 70% 短縮することを示した．

キーワード グラフ，クラスタリング，コミュニティ抽出

1. はじめに

グラフ構造はデータをノードとエッジで表現した基本的なデータ構造であり，情報推薦や情報検索，科学データ分析などの様々な分野で利用されている．特に近年では，数億ノードから構成される大規模なグラフ構造が登場し，このようなデータに対する高速な解析処理技術への需要が高まっている．例えば，Facebook では 2012 年に 1ヶ月当たりのアクティブユーザ数が 10 億人，また Twitter では一日当たりの投稿数が 3 億 4000 万を突破したと報告されている [1]．このように，大規模グラフは現実に存在し今後も規模をさらに増大させていくことが考えられ，大規模なグラフ構造に対する高速な解析手法は必要不可欠な技術となってくると言える．

グラフ構造解析のひとつとして，クラスタリングが挙げられる．グラフ構造にはクラスタと呼ばれる相互に密な接続を有する部分ノード集合が存在する．例えば，Web グラフではトピックの近いページ集合が互いにリンクすることで，トピックの類似したページ群がコミュニティを形成する傾向にある．グラフ構造中のクラスタは互いに共通した性質を持つことから，グラフ構造の理解や様々なアプリケーションに利用され，重要な要素技術となっている．この背景からこれまで Modularity に基づく手法 [2], [3], [4], [5] や min-max cut による手法 [6], [7] など様々なクラスタリング手法の研究が行われてきた．

構造的類似度に基づいたグラフクラスタリング手法 [8], [9], [10], [11], [12], [13] は特に注目を集めているクラスタリング手法である．これらの手法はグラフ構造からクラスタを抽出するのみではなく，ハブや外れ値といったノードも併せて抽出することを可能にする．グラフ構造において，ハブは複数のクラスタを橋渡しする構造を持ち，周辺のクラスタに影響力のあるノードとして扱われることが多い．一方で外れ値は，クラスタやハブに対してノード単体で接続した構造を有し，多くの場合ノイズとして扱われる．この特性から，ハブや外れ値の抽出は，グラフ構造の理解やマーケティングや情報拡散分析，Web 検索などグラフ構造に基づく幅広い応用に対し有効であることが知られている．そのため，これらの手法はクラスタのみを抽出する従来の手法に対して，より細かな分析を可能にする手法として期待されている．

その中でも，Xu らによる SCAN [8] は，高速な構造的類似度に基づくグラフクラスタリング手法の一つである．SCAN は多次元ベクトルデータに対する密度ベースのクラスタリング手法として有名な DBSCAN [14] をグラフ構造に応用した手法である．SCAN では，事前に 2 つのパラメータを与えることでクラスタリングを実行する．1 つ目のパラメータはクラスタを構成する構造的類似度の閾値 ϵ である．2 つ目のパラメータはクラスタを構成する最小ノード数を示す μ である．これら 2 つのパラメータ ϵ，μ を基にして，SCAN はグラフ構造からクラスタ，ハブ，外れ値の抽出を行う．まず，SCAN はグラフ構造に含まれる全てのエッジに対して，構造的類似度を計算する．この構造的類似度は，隣接する 2 つのノード間で共通して隣接するノード集合の割合，言い換えると 2 つのノードの隣接ノード集合の積集合の割合を測る尺度として定義されている [8]．全てのエッジに対して構造的類似度を計算した後，事前に与えられたパラメータ ϵ，μ を満たす，core と呼ばれるクラスタの核となるノードを見つける．その後，ϵ，μ の制約の下，core を中心にクラスタサイズが収束するまでクラスタを拡張していく．いずれのクラスタにも属さなかったノードに対してハブと外れ値の判定を行い処理を終了する．これまでに述べたとおり，SCAN は Modularity による手法や min-max cut による手法とは異なり，パラメータ ϵ および μ を与えることでノードをクラスタ，ハブおよび外れ値に分類することができる．

しかしながら，SCAN はその計算量の大きさから大規模なグラフ構造を対象としたクラスタリングは難しい．SCAN では，全てのエッジに対して構造的類似度を計算する．そのため，グラフ構造中のエッジ数を $|E|$ とした時に，$O(|E|)$ の時間計算量が生じる．またグラフ構造中のノード数を $|V|$ としたとき $|E| \approx |V|^2$ となるような場合，SCAN の時間計算量は最悪計算量 $O(|V|^2)$ となる．したがって，近年増加する大規模なグラフ構造のクラスタリングに膨大な処理時間を要することになる．

1.1 本研究の貢献

本稿では以下の問題について取り組む．

[**問題定義** 1]（構造的類似度に基づく高速なクラスタリング）

Given: グラフ構造 $G = (V, E)$，構造的類似度の閾値 ϵ，およびクラスタを構成する最小ノード数 μ.

Find: グラフ構造 G から，クラスタ集合 C，ハブ集合 H および外れ値集合 O.

本稿では問題定義 1 を従来よりも大規模なグラフ構造に対して適用可能にするため，SCAN と同一のクラスタ集合 C，ハブ集合 H，外れ値集合 O を高速に抽出するクラスタリング手法を提案する．本稿では従来手法 SCAN の計算コストを削減するために，現実のグラフ構造の高いクラスタ性に着目した．クラスタ性はあるノードの隣接ノード同士が互いエッジで接続しやすい傾向にあるという性質ある．現実のグラフ構造は高いクラスタ性を持つことから，クラスタを形成しやすく密にエッジで接続した部分グラフ構造を内包していると考えられる．そこで本研究では，全てのエッジに対して構造的類似度の計算を必要とした SCAN を高速化するために，高いクラスタ性により密なエッジの接続を有する部分グラフ構造の計算を可能な限り回避するような手法を考える．提案手法では，最短ホップ数が 2 となるような部分ノード集合を抽出し，抽出した部分ノードに含まれるノードに接続したエッジについてのみ構造的類似度計算を行う．本稿では，本手法で抽出する最短ホップ数が 2 となる部分ノード集合を 2-hop away ノードと呼ぶ．2-hop away ノードに接続するエッジについてのみ構造的類似度の計算を行うことで，高いクラスタ性により密なエッジの接続を有するサブグラフ構造を少ない計算回数でクラスタリングする．従来手法である SCAN では，全てのエッジについて構造的類似度計算を行う必要があったのに対し，提案手法は 2-hop away ノードに接続したエッジのみ構造的類似度計算を行う．ゆえに，クラスタリング全体で計算されるエッジの本数を削減することができる．提案手法はグラフ中のノード数 $|V|$ に対して $O(|V|)$ の時間計算量を示す．その結果として，提案手法は以下の特性を有する．

- **高速性**: 先に述べた 2 つのアプローチにより，従来手法 SCAN に対して高速にクラスタリングを行うことができる．
- **正確性**: 提案手法で用いるアプローチは，SCAN におけるクラスタの定義を満たす． したがって，問題定義 1 において SCAN と同一のクラスタを得ることができる．
- **運用性**: 提案手法は事前計算を必要とせず，グラフ構造 G と 2 つのパラメータ ϵ，μ を与えることによりクラスタリングを実行できる．

表 1 記号の定義

記号	定義
$\|V\|$	ノード数
$\|E\|$	エッジ数
ϵ	クラスタを構成するための構造的類似度の閾値
μ	クラスタに含まれる最小ノード数
$\Gamma(u)$	ノード u の構造的隣接ノード集合
$\|\Gamma(u)\|$	$\Gamma(u)$ に含まれるノード数
$\sigma(u, v)$	エッジ (u, v) の構造的類似度
$N_\epsilon(u)$	ノード u の $\epsilon - neighborhood$
$\|N_\epsilon(u)\|$	N_ϵ に含まれるノード数
$K_{\epsilon,\mu}(u)$	core であるノード u
$u \mapsto_{\epsilon,\mu} v$	ノード u からノード v への direct structure reachability
$u \to_{\epsilon,\mu} v$	ノード u からノード v への structure reachability

本稿で提案する手法は我々の知る限り，クラスタリングの高速性と正確性の両方を同時に満たす最初の手法である．従来手法である SCAN は高い計算コストを有するものの，クラスタだけでなくハブや外れ値を抽出できることから幅広いアプリケーションで利用されている．本稿で提案する手法は，既に従来技術が利用されているアプリケーションや将来的に利用が予測される分野において，その処理性能の向上に貢献する．

本稿の構成は，次の通りである．2. 節で本稿の前提となる知識について概説する．3. 節にて提案手法の詳細について説明し，4. 節において提案手法の評価と分析を行う．5. 節にて関連研究について述べ，6. 節にて，本稿をまとめ，今後の課題について論ずる．

2. 事 前 準 備

従来手法 SCAN [8] を基に，本稿の前提について述べる．本稿では無向重みなしグラフ $G = (V, E)$ に対して，クラスタ集合 C，ハブ集合 H および外れ値集合 O を抽出することを考える．表 1 にて本稿で用いる記号とその定義を示す．

従来手法では 2 つのノード間で共有される隣接ノード集合の割合を評価することで構造類似度を計算していく．ここで用いる隣接ノード集合は以下のように定義される．

[**定義** 1]（構造的隣接ノード集合） $u \in V$ とするとき，隣接ノード集合はノード u にエッジで接続するノードとノード u 自身から構成される集合 $\Gamma(u)$ で与えられる．

$$\Gamma(u) = \{v \in V | \{u, v\} \in E\} \cup \{u\}.$$

また先に述べたように，クラスタリングで用いられる 2 ノード間の構造的類似度は定義 1 に基づき以下のように定義される．

[**定義** 2]（構造的類似度） $u, v \in V$，$|\Gamma(u)|$ を隣接ノード集合に含まれるノード数とするとき，ノード u，v 間の構造的類似度は $\sigma(u, v)$ となる．

$$\sigma(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{\sqrt{|\Gamma(u)||\Gamma(v)|}}.$$

定義 2 に示したように，ノード u，v 間の $\sigma(u, v)$ は共有されるノードがない場合 $\sigma(u, v) = 0$，互いに全て共有する場合には $\sigma(u, v) = 1$ となる．

構造的類似度に基づくクラスタリング手法 SCAN はクラスタを構成するための類似度の閾値として ϵ を導入し，類似度 ϵ 以上で接続する隣接ノード集合 ϵ-neighborhood を定義する．

[**定義** 3]（ϵ-neighborhood） $u \in V$，$\epsilon \in \mathbb{R}$ とするとき，ϵ-neighborhood $N_\epsilon(u)$ は以下のように定義される．

$$N_\epsilon(u) = \{v \in \Gamma(u) | \sigma(u,v) \geqq \epsilon\}.$$

ここで，クラスタを構成する最小ノード数としてパラメータ μ を導入し，特別なノードのクラスとして *core* を定義する．

［**定義** 4］（Core） $u \in V$, $\epsilon \in \mathbb{R}$, $\mu \in \mathbb{N}$ および $|N_\epsilon(u)|$ をノード u における ϵ-neighborhood のノード数とするとき，*core* は以下のように定義される．

$$K_{\epsilon,\mu}(u) \Leftrightarrow |N_\epsilon(u)| \geqq \mu.$$

従来手法では定義 4 を満たす core ノード u をクラスタの中心として，N_ϵ をクラスタのメンバとして同一のクラスタに所属させる．この手順により，パラメータ ϵ, μ を用いてクラスタの形を決定し，μ を用いてクラスタの最小サイズを定める．この考えは direct structure reachability として以下に定義される．

［**定義** 5］（Direct structure reachability） $u, v \in V$, $\epsilon \in \mathbb{R}$ および $\mu \in \mathbb{N}$ とするとき，ノード u とノード v における direct structure reachability $u \mapsto_{\epsilon,\mu} v$ は以下のようになる．

$$u \mapsto_{\epsilon,\mu} v \Leftrightarrow K_{\epsilon,\mu}(u) \wedge v \in N_\epsilon(u).$$

定義 5 はノード u, v が core である場合対称であるが，どちらか一方が core でない場合には非対称となる．ゆえに，$u \mapsto_{\epsilon,\mu} v$ かつノード v が core でない場合，ノード v は $K_{\epsilon,\mu}(u)$ が構築するクラスタの境界に面したノードとなる．本稿ではこのようなノード v を *border* と呼ぶ．定義 5 をより一般的な形に拡張した structure reachability を示す．

［**定義** 6］（Structure reachability） $u, v \in V$, $\epsilon \in \mathbb{R}$ および $\mu \in \mathbb{N}$ とすると，ノード u とノード v における structure reachability $u \to_{\epsilon,\mu} v$ は以下に定義される．

$$u \to_{\epsilon,\mu} v \ s.t. \ (u_i \mapsto_{\epsilon,\mu} u_{i+1}) \wedge (u_1 = u) \wedge (u_n = v).$$

ここで定義された structure reachability は推移律を満たし，非対称である．この structure reachability は direct structure reachability の推移閉包であるり，定義 5 の対称性から，推移閉包の構成要素である $u_1, \ldots, u_{n-1}$ は core ノードである必要がある．core 同士が direct structure reachability を示す時，それらの ϵ-neighbor は同一クラスタに属することになる．この考えは structure connected クラスタとして以下に定義される．

［**定義** 7］（Structure-Connected クラスタ） ノード $u \in V$, $\epsilon \in \mathbb{R}$ および $\mu \in \mathbb{N}$ とすると，$K_{\epsilon,\mu}(u)$ から求まるクラスタ $C[u] \in C$ が structure-connected クラスタである必要十分条件は (1) $u \in C[u]$; (2) $\forall v \in V$, $u \to_{\epsilon,\mu} v \Leftrightarrow v \in C[u]$.

この定義より，structure-connected クラスタはその中に含まれる core により一意に決めることができることがわかる．ここで，border ノードは，複数のクラスタの core から direct structure reachability となる可能性があることに注意されたい．この場合，border ノードは border ノード自身が core でない限り，複数のクラスタに属する．

パラメータ ϵ, μ に対するクラスタリング結果が C で与えられる時，ノード集合 V にはいずれのクラスタ集合 C にも属さないノードが存在する場合がある．これらのノードはハブもしくは外れ値に分類される．

［**定義** 8］（ハブ） クラスタ集合 C，ハブ集合 H とすると，ノード $h \in V$ がハブである (*e.g.* $h \in H$) 必要十分条件は (1) $h \notin \forall C_i \in C$, (2) $u, v \in \Gamma(h)$ $s.t$ $u \in C_i, v \in C_j, i \neq j$.

［**定義** 9］（外れ値） クラスタ集合 C，ハブ集合 H，外れ値集合 O とすると，ノード $o \in V$ が外れ値 (*e.g.* $o \in O$) である必要十分条件は $o \notin C \wedge o \notin H$.

2.1 SCAN のアルゴリズム

従来手法 SCAN のアルゴリズムを Algorithm1 に示す．SCAN はグラフ構造中の全てのエッジを全て走査することにより，与えられたパラメータに応じた全ての structure-connected クラスタを抽出する．クラスタが決定しなかったノードに対してハブもしくは外れ値のいずれであるか判定を行う．

structure-connected クラスタの抽出アルゴリズムについて述べる．SCAN の開始時点において，全てのノードは *unclassified* というクラスタに所属させる．SCAN は unclassified となっている各ノードに対して，そのノードが core の条件を満たすかどうか判定を行う．この際に，SCAN が選択した unclassified なノードの構造的隣接ノード集合に含まれる全てのノードに対して，構造的類似度を計算する必要がある．SCAN が判定を行ったノードが core の時，このノードを中心に structure-connected クラスタの探索を実行し，ノードが core で無かった場合には，そのノードを *non-member* というクラスタに所属させる．

SCAN は core と判定されたノードを中心にクラスタを探索するために，新たなクラスタ ID として clusterID を生成する．以後の処理では，core と判定されたノードから structure reachability で到達可能な全てのノードの所属クラスタを clusterID としていく．まず，SCAN はキュー Q へ core と判定されたノードの ϵ-neighborhood に含まれる全てのノードを挿入する．そして，キュー Q に入力された全てのノードに対して，direct structure reachable となるノード集合 R を計算する．R を求めるために SCAN はキュー Q に含まれる全てのノードの全ての構造的隣接ノードに対して構造的類似度を計算する．最後に R に含まれるノードの所属クラスタが unclassified の場合，新たにキュー Q へと挿入する．R に含まれるノードが unclassified もしくは non-member の場合，先に生成した clusterID を割り当てる．この処理をキュー Q に含まれるノードがなくなるまで継続し，structure-connected クラスタの抽出を行う．

次に，ハブと外れ値の判定アルゴリズムについて述べる．全てのノードに対する structure-connected クラスタの抽出処理が終了した後，SCAN は全ての non-member となったノードを走査しハブと外れ値の判定を行う．このとき，ある non-member のノードが 2 つ以上の異なるクラスタに隣接している場合は，このノードをハブと判定し，1 つ以下のクラスタにのみ隣接している場合には外れ値と判定する．

SCAN は全てのノードの構造的隣接ノード集合に含まれるノードに対して，構造的類似度計算を行う必要がある．したがって，SCAN の時間計算量は $O(|E|)$ となり，$|E| \approx |V|^2$ に近づく場合，最悪計算量として $O(|V|^2)$ の計算量を必要とする．

Algorithm 1 SCAN

```
Require: G = (V, E), ϵ ∈ ℝ, μ ∈ ℕ;
Ensure: clusters C, hubs H, and outliers O;
 1: for each unclassified node u ∈ V do
 2:     if K_{ϵ,μ}(u) then
 3:         generate new clusterID;
 4:         insert ∀x ∈ N_ϵ(u) into queue Q;
 5:         while Q ≠ ∅ do
 6:             y ∈ Q;
 7:             R = {x ∈ V | y ↦_{ϵ,μ} x};
 8:             for each x ∈ R do
 9:                 if x is unclassified then
10:                     insert x to queue Q;
11:                 end if
12:                 if x is unclassified or non-member then
13:                     assign current clusterID to x;
14:                 end if
15:             end for
16:             remove y from Q;
17:         end while
18:     else
19:         assign non-member to u;
20:     end if
21: end for
22: for each non-member node u do
23:     if ∃x, y ∈ Γ(u), x.clusterID ≠ y.clusterID then
24:         label u as hub;
25:     else
26:         label u as outlier;
27:     end if
28: end for
```

3. 提 案 手 法

本節では提案手法について概説する．提案手法は従来手法と同一のクラスタリング結果をより高速に抽出することができる．最初に提案手法を構成する基本的なアイデアについて述べ，その詳細について 3.2 節以降で説明する．

3.1 基本アイデア

従来手法である SCAN ではグラフ構造中の全てのエッジ E に対し類似度計算を行うことから，従来手法は最悪の場合 $O(|V|^2)$ の時間計算量を要する．ゆえに，クラスタリングにかかる時間を短縮するためには，構造的類似度が計算されるエッジの数を減らすことが重要である．

そこでまず本稿ではグラフのクラスタ性に着目した．グラフのクラスタ性とは，エッジで接続する 2 つのノード u，v と，同じくエッジで接続する 2 つのノード v，w が存在するときに，ノード u，w がエッジで接続している割合を表したものである．一般的に現実のグラフ構造では高いクラスタ性を示すことが知られており，本研究でクラスタリングの対称とするグラフ構造も例外ではない．このことから，グラフ構造中の多くのノードは高い確率でその隣接ノード同士がエッジで接続されていると考えられる．また同様に，エッジで接続した隣接ノード同士は他のノードから共に接続されている確率も高い．すなわち，あるノードの隣接ノード集合は他のノードの隣接ノード集合である可能性が高く，パラメータ μ が適切に設定されている状態を仮定すると，隣接ノード集合に隣接するノードについてのみ構造的類似度を計算することで，隣接ノード集合に対する構造的類似度計算を補完することができると考えられる．

そこで提案手法では，時間計算量 $O(|V|^2)$ を削減するために，最短ホップ数が 2 となるような部分ノード集合を抽出し，抽出した部分ノード集合に含まれるノードに接続したエッジについてのみ構造的類似度計算を行う．本稿では最短ホップ数が 2 となるような部分ノード集合を 2-hop away ノード集合と呼び，その詳細については 3.2 節で定義する．このような方式を採ることで，2-hop away ノード集合から最短ホップ数が 1 となるノード集合に接続したエッジに対する構造的類似度計算を削減することができる．

提案手法は 2 つの利点を有する．1 つ目は，現実世界に数多く存在する複雑ネットワークに対して，従来手法 SCAN よりも高速にクラスタリングできるという点である．複雑ネットワークは，先に述べた様に高いクラスタ性を有することが知られている．このような特性をもつ現実のグラフデータに提案手法を用いることで，数多くのエッジに対する構造的類似度計算を少数のエッジに対する構造的類似度計算により補完し，計算量を削減することができる．具体的には，ノード数が $|V|$ となるグラフ構造に対して $O(|V|)$ の時間計算量でクラスタリングを実行することができる．

2 つ目は，クラスタリング結果の正確性である．提案手法は，従来手法の SCAN と異なり，一部のエッジに対してのみ構造的類似度の計算を行うが，出力されるクラスタ，ハブ，および外れ値は同一のパラメータに対して同一の結果を出力する．その理由として 2-hop away ノード集合のクラスタ包含性という特性が挙げられる．提案手法は，計算量削減のため，2-hop away ノード集合を逐次的に選択していくが，選択したノード集合とその隣接ノードから構成されるサブグラフ構造内に structure-connected クラスタが完全に包含される特性を有する．言い換えると，提案手法で選択した 2-hop away ノード集合で到達不可能なノードは structure-connected クラスタに含まれないということが保証されている．ゆえに，クラスタの正確性が保証されている．2-hop away ノード集合のクラスタ包含性の詳細については 3.3 節で述べる．

3.2 2-hop away ノードによるクラスタリング

提案手法は，任意のノード u を選択し，そのノード u に接続した全てのエッジに対して構造的類似度を計算する．類似度計算後，ノード u を起点に 2-hop away ノード集合を取得する．起点とされるノード u を本稿では *pivot* と呼び，ノード u を pivot とする 2-hop away ノード集合の定義を以下に示す．

［**定義** 10］（2-hop away ノード集合） ノード $u \in V$ を pivot，ノード w を $w \in N_\epsilon(u) \backslash \{u\}$ とするとき，ノード u に対する 2-hop away ノード集合 $H(u)$ は，

$$H(u) = \{v \in V | (u, v) \notin E \wedge (v, w) \in E\},$$

で与えられるノード集合である．

定義 10 で与えられる，ノード u の 2-hop away ノード集合は，$N_\epsilon(u)$ に含まれるノードに隣接し，ノード u からの最短ホップ数が 2 となるノードの集合である．ここで従来手法 SCAN ではノード u の隣接ノード集合 $\Gamma(u)$ に含まれる全てのノードに接続したエッジについて構造的類似度を計算した．これに対し，提案手法は $\Gamma(u)$ に含まれるノードに接続したエッジについては構造的類似度を計算せず，定義 10 により取得した 2-hop away ノードに含まれるノードに接続したエッジに対してのみ構造的類似度を計算する．これにより，提案手法は pivot であるノード u にに関する構造的類似度計算を削減する．

その後，$H(u)$ に含まれる全てのノードを pivot とし，新たに選択された pivot に基づき 2-hop away ノード集合を拡張す

る．この際に，2-hop away ノード集合を拡張する時点までに選択された pivot から direct structure reachable とされたノード集合は拡張された 2-hop away ノード集合から除外する．本稿では拡張された 2-hop away ノード集合を拡張 2-hop away ノード集合と呼び，以下に定義する．

［**定義** 11］（拡張 2-hop away ノード集合） ノード u_n を新たに選択した pivot，ノード $u_1, u_2, \ldots, u_{n-1} \in V$ をノード u_n が pivot として選択される以前に選択された pivot とする．ただし，ノード u_{i-1} と u_i に対して，ノード u_{i-1} が先に選択されたものとする．また，ノード w を $w \in \Gamma(u_n)$ とする．このとき，u_n が pivot として選択された際に得られる拡張 2-hop away ノード集合 $H(u_n)$ は，

$$H(u_n)=\{v \in V | (u,v) \notin E \wedge (v,w) \in E \wedge v \notin \bigcup_{i=0}^{n-1} N_\epsilon(u_i) \cup H(u_i)\},$$

で与えられるノード集合である．

提案手法は選択した pivot の集合を P とするとき，$\{\bigcup_{i=0}^{n} H(u_i)\} \backslash P = \emptyset$ となるまで，拡張 2-hop away ノード集合を取得し，取得したノード集合に接続するエッジに対して構造的類似度を計算する．ノード集合 $\{\bigcup_{i=0}^{n} H(u_i)\} \backslash P$ が収束する条件は (1) 提案手法が全てのノードを走査し終えた場合，もしくは，(2) $\bigcup_{i=0}^{n} N_\epsilon(u_i)$ が収束した場合の 2 通りである．我々の検証では，事前に与えられるパラメータ ϵ および μ は極めて小さい場合を除き，提案手法は (2) の理由で収束する．

提案手法では，グラフ中に未計算のノードが存在する限り定義 10 および定義 11 で定義される（拡張）2-hop away ノード集合の取得と構造的類似度の計算を繰り返す．提案手法のアルゴリズムの詳細については Algorithm2 を参照されたい．

3.2.1 非計算対象ノードの後処理

提案手法では定義 7 で与えられた structure-connected クラスタと同一のクラスタを抽出するために，2 つ以上のクラスタに属する未計算のノードに対して後処理を行う．

2 つ以上のクラスタに属するノードは，そのノードが core である場合，定義 6 により隣接するクラスタが同一の structure-connected クラスタとなる．ゆえに，2 つ以上のクラスタに属するノードが存在する場合，そのノードが core となるかを判定する必要がある．提案手法では所属クラスタ数の多いノードから順に選択し，所属クラスタ数がパラメータ μ 以上の場合には，類似度計算を必要とせずに core と判定し，関連するクラスタのクラスタラベルを更新する．上記の処理が終了した後，依然として複数のクラスタに属しているノードについてのみ，未計算のエッジに関して構造的類似度を計算し core の判定を行う．

この後処理は，一見すると提案手法の計算量を従来手法 SCAN と同等のものに近づけてしまう手法にみえる．しかしながら，2 つの理由により計算量の増加が回避されている．第 1 の理由は，複数クラスタに属する全てのノードを後処理する必要がない点である．グラフ構造のクラスタ性に着目すると，同一の core に隣接するノードは複数存在することが示唆される．このような場合，複数存在するノードのうち，どれか一つでも core であることが判明すれば，その他のノードについては後処理する必要がない．第 2 の理由は，先行研究 [8] に示された知見によるものである．文献 [8] では，*"an ϵ value between 0.5 and 0.8 is normally sufficient to achieve a good clustering result. We recommend a value for μ, of 2."* と示されている．この知見に従い，よいクラスタが得られる $\mu = 2$ をパラメータとして与えた場合，複数のクラスタに属するノードは自明に core となる．以上の理由から現実のグラフ構造に対しては計算量の増加が回避されている．実際に我々の評価実験では，未計算のエッジに対する構造的類似度計算の計算は一度も発生せず，かつ，core の判定に要する時間もクラスタリング全体にかかる計算時間に対して無視できる程度に小さいという結果が得られている．

Algorithm 2 Proposed method

Require: $G = (V, E)$, $\epsilon \in \mathbb{R}$, $\mu \in \mathbb{N}$;
Ensure: clusters C, hubs H, and outliers O;
for each *unclassified* node $u \in V$ **do**
  $P = \{u\}$;
  if $K_{\epsilon,\mu}(u)$ **then**
    generate new clusterID id;
    assign id to $\forall v \in N_\epsilon(u)$;
  else
    label u as *non-member*;
  end if
  while $\{\bigcup_{\forall u \in P} H(u)\} \backslash P \neq \emptyset$ **do**
    for $v \in \{\bigcup_{\forall u \in P} H(u)\} \backslash P$ **do**
      if $K_{\epsilon,\mu}(v)$ **then**
        generate new clusterID id;
        assign id to $\forall v \in N_\epsilon(v)$;
      else
        label v as *non-member*;
      end if
    end for
    for $u \in \{\bigcup_{\forall u \in P} H(u)\} \backslash P$ **do**
      $P = P \cup H(u)$;
    end for
  end while
end for
while each node u which labeled as several id **do**
  if the number of $ids < \mu$ **then**
    compute structural similarities for non-evaluated edges
  end if
  if $K_{\epsilon,\mu}(u)$ **then**
    u is core;
    refine cluster ids
  end if
end while
for each non-member node u **do**
  if $\exists x, y \in \Gamma(u), x$.clusterID $\neq y$.clusterID **then**
    label u as *hub*;
  else
    label u as *outlier*;
  end if
end for

3.3 提案手法の正確性

提案手法により抽出されるクラスタの正確性について証明する．本稿でいう正確性とは同一のグラフ構造およびパラメータを与えた際に，従来手法である SCAN と同一のクラスタリング結果を出力することである．

クラスタの正確性を示すためには，2-hop away ノード集合によって計算されるノード集合が 2-hop away ノード集合に含まれる core ノードが構築する structure-connected クラスタを全て含んでいる必要がある．これは 2-hop away ノード集合のクラスタ包含性により証明できる．まず本節ではクラスタ包含性を示すために，2-hop away ノード集合の non-direct structural reachability を補題 1 で示す．補題 1 では，拡張 2-hop away ノード集合の抽出が収束した際に得られた pivot 集合を P，pivot 集合に含まれるノードの ϵ-neighborhood の和集合を $\bigcup_{\forall u \in P} N_\epsilon(u)$ とする．また，2-hop away ノード集合抽出時に得られるノード集合を $V_H = \{\bigcup_{\forall u \in P} N_\epsilon\} \cup P$ とする．

［**補題** 1］（non-direct structural reachability） 拡張 2-hop away ノード集合抽出時に走査されるノードの部分集合を V_H，それに含まれない全てのノード集合を $\overline{V}_H = V \backslash V_H$ とするとき，

$\{\bigcup_{i=0}^{n} H(u_i)\}\backslash P = \emptyset$ ならば，$\{\forall(u,v) \in E | u \in V_H \wedge v \in \overline{V}_H\}$ に対して $\sigma(u,v) < \epsilon$ が成立する．

証明　背理法により証明する．まず，$u \in V_H$，$v \in \overline{V}_H$ に対して $\sigma(u,v) \geqq \epsilon$ となるエッジが存在すると仮定する．仮定より，ノード u は自明にノード v の ϵ-neighborhood に含まれる．ノード v は V_H に含まれることから，$v \in P$ もしくは，$v \in \bigcup_{\forall u \in P} N_\epsilon(u)$ である．$v \in P$ のとき，$u \in \bigcup_{\forall u \in P} N_\epsilon(u)$ となることから，$\{\bigcup_{i=0}^{n} H(u_i)\}\backslash P = \emptyset$ に矛盾する．$v \in \bigcup_{\forall u \in P} N_\epsilon(u)$ のとき $u \in P$ となることから，同様に $\{\bigcup_{i=0}^{n} H(u_i)\}\backslash P = \emptyset$ に矛盾する．ゆえに $\{\bigcup_{i=0}^{n} H(u_i)\}\backslash P = \emptyset$ ならば，$\{\forall(u,v) \in E | u \in V_H \wedge v \in \overline{V}_H\}$ に対して $\sigma(u,v) < \epsilon$ となる．　□

補題 1 より，拡張 2-hop away ノード集合の抽出が収束した際の部分ノード集合 V_H は構造的類似度が ϵ よりも小さなエッジでのみ $\overline{V}_H$ と接続する．これにより，補題 2 に示す（拡張）2-hop away ノード集合によって得られる部分ノード集合 V_H のクラスタ包含性が証明できる．

［**補題** 2］（V_H のクラスタ包含性）　ノード $u \in \{u \in V | u \in V_H \wedge K_{\epsilon,\mu}(u)\}$ とし，ノード u による structure-connected クラスタを $C[u]$ とした時，$\forall v \in C[u] \Rightarrow v \in V_H$ が成立する．

証明　補題 1 より，V_H は構造的類似度が ϵ より小さなエッジにのみ接続している．定義 6，定義 7 より，structure-connected クラスタは direct structure-connected である必要があるため，構造的類似度が ϵ より小さなエッジで接続したノードは strcuture-connected クラスタに含まれない．従って，$\forall v \in C[u] \Rightarrow v \in V_H$ が成立する．　□

補題 2 により，2-hop away ノード集合に基づくクラスタリング手法がクラスタの精度に影響を与えないことを示した．このことから正確なクラスタを抽出するためには，2-hop away ノード集合に基づくアプローチにより構造的類似度計算が削減されたノードに対する core 判定を行う必要があるが，ここで述べた core 判定は前節で述べた非計算対象ノードへの後処理によって対応される．ゆえに，提案手法は定義 7 と同一のクラスタを抽出することが可能となる．

3.4　計算量分析

最後に本節で提案手法の計算量を分析する．ノード数 $|V|$，エッジ数 $|E|$ のグラフ構造に対する計算量を定理 1 に示す．

［**定理** 1］（提案手法の計算量）　2-hop away ノード集合に基づく提案手法は $O(|V|)$ の計算量を要する．

証明　各ノードの平均次数を k，クラスタ係数を c とした時に $\beta = 1 - c$ と仮定する．これらの仮定から，各 pivot に対する計算量を考える．2-hop away ノード集合を求める際に与えられる最初に pivot に対しては，次数全てに対して構造的類似度を計算することから $O(k)$．それ以外の pivot については，平均的に ck 本のエッジが他の pivot と共有されているとかんがえられるため，構造的類似度を必要とする計算量は $O(\beta k)$．さらに 2-hop away ノード集合に含まれる pivot ノードの数は，pivot の隣接ノードになったノードが pivot にならない点から，最大で $\frac{|V|-k}{\beta k}$．非計算対象ノードに対する後処理を $O(C)$，ただし $C \approx 0$，とおくと，各 pivot 辺りの計算量と pivot 数から全体の計算量は $O(\frac{|V|-k}{\beta k}\beta k + k + C) = O(|V| + C) = O(|V|)$．　□

一般にグラフ構造において $|V| \ll |E|$ であるため，提案手法は従来手法 SCAN よりも少ない計算量でクラスタリングを実行することができる．

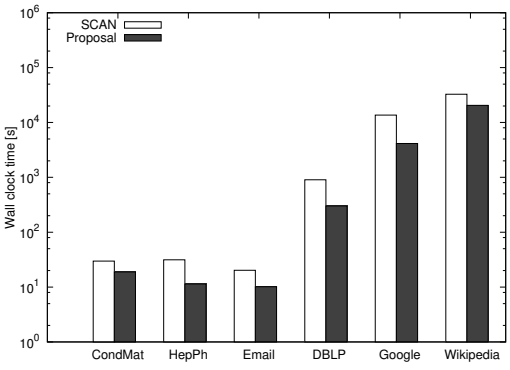

図 1　クラスタリング実行時間の比較

4. 評価実験

提案手法の有効性を評価するために，我々の提案したその高速化手法および Xu らによる SCAN [8] に対し，処理の高速性およびクラスタリング結果の正確性の観点から比較評価を行う．本実験には CPU が Intel Xeon Quad–Core L5640，メモリが 144GB の Linux サーバを利用する．また，提案手法および SCAN は gcc–g++_4.1.2 を用いて実装した．

4.1　高速性

本実験では，グラフ構造とパラメータ ϵ および μ を与えた後，クラスタリング処理が終了するまで処理を行った際の処理時間を示し比較を行う．本実験で用いたパラメータは文献 [8] で用いられているものと同一のものを利用し，いずれのデータセットに対しても $\epsilon = 0.7$，$\mu = 3$ とした．本実験に用いたデータセットは Stanford Large Network Analysis Project (注1)から以下のデータセットを使用した．またデータセットの統計情報を表 2 に示す．

実験結果を図 1 に示す．縦軸が対数表示となっていることに注意されたい．図 1 に示したように，いずれのデータセットに対しても提案手法は従来手法 SCAN に対して約 40% から 約 70% 計算時間を短縮している．特にクラスタ係数の大きなデータセットである Google では計算時間を 69.9% 短縮しており，最も大きく計算時間を短縮できている．これに対し，クラスタ係数の極めて小さな Wikipedia では，計算時間の短縮率が最も小さく 37% という結果を得た．

この結果から，提案手法は従来手法に対してより高速に構造的類似度に基づくクラスタの抽出ができることを示した．また特に，クラスタ係数の大きな複雑ネットワークに対して提案手法は有効であることを示した．

4.2　正確性

提案手法と従来手法 SCAN の出力するクラスタリング結果の正確性について評価を行う．クラスタリング結果の比較には

(注1)：http://snap.stanford.edu/index.html

表 2　データセットの詳細

Dataset	Acronym	$\|V\|$	$\|E\|$	Average cluster coefficient	Diameter	90-percentile effective diameter	Source
ca-ComdMat	CondMat	23,133	93,497	0.6334	14	6.5	[15]
cit-HepPh	HepPh	34,546	421,578	0.2848	12	5	[16]
email-Enron	Email	36,692	367,662	0.4970	11	4.8	[17]
com-DBLP	DBLP	317,080	1,049,866	0.6324	21	8	[18]
web-Google	Google	875,713	5,105,039	0.5143	21	8.1	[17]
wiki-Talk	Wikipedia	2,394,385	5,021,410	0.0526	9	4	[19]

表 3　ARI の比較結果

Dataset	SCAN	Proposal
College football ($\epsilon = 0.5, \mu = 2$)	1.0	1.0
Political books ($\epsilon = 0.35, \mu = 2$)	0.708	0.708

調整ランド指数 (ARI: Adjusted Rand Index) [20] を用いた．ARI はクラスタの正解ラベルに対してするクラスタリング結果の一致度を評価する指数であり，1 に近づくほどよい高い一致度があることを示す．ARI の詳細については文献 [20] を参照されたい．本稿では正解クラスタラベルが与えられている以下のデータセットに対して，クラスタリング結果の ARI を比較した．

- **College football [21]：** 2006 年の NCAA フットボール (Division 1-A) の対戦スケジュールを基に作成したグラフ構造である．ノード数 180，エッジ数 787 であり，ノードがフットボールチームの所属校，エッジが対戦スケジュールを表す．このデータセットでは，所属校が 11 のグループに分割されている．本稿では文献 [8] に基づき $\epsilon = 0.5, \mu = 2$ とし，提案手法と従来手法 SCAN を適用した．
- **Political books (注2) (注3)：** Aamazon.com で販売されるアメリカの政治学に関する本の購買履歴を基に作成されたグラフ構造である．ノード数は 105，エッジ数は 441 であり，ノードが本，エッジが同一の消費者によって購入された事実を表す．このデータセットでは，各本は *liberal*，*neutral*，*conservative* の 3 グループに分割されている．本稿では文献 [8] に基づき $\epsilon = 0.35, \mu = 2$ とし，提案手法と従来手法 SCAN を適用した．

各データセットに対する ARI の比較結果を表 3 に示す．表 3 に示すように，同一のデータセットに対して同一のパラメータが与えられる時，提案手法の ARI は従来手法 SCAN の示す ARI と一致する．すなわち，提案手法は正解ラベルに対して従来手法 SCAN と同等の ARI を示すクラスタリング結果を出力していることがわかる．

5. 関 連 研 究

グラフ構造中からクラスタを抽出するグラフクラスタリングはデータマイニング分野において重要な技術であり，これまで min-max cut [6] や normalized cut [7]，Modularity に基づく手法 [2], [3], [4], [5] など，数多く研究されてきた．本節ではこれらの中でも特に，一般的によく利用される Modularity に基づく手法および，本稿の議論の対象である構造的類似度に基づく手法 [8], [9], [10], [11], [12], [13] の 2 つについて述べる．

5.1 Modularity に基づく手法

Modularity とは Newman らにより提案されたクラスタの質を評価する指標である [21]．Modularity は与えられたクラスタ構造がランダムグラフの構造から離れているほど良いスコアを示す．言い換えると，クラスタ内部のエッジ接続が密であり，クラスタ間のエッジ接続が疎であるようなクラスタ程良いスコアを示す．このことから Modularity に基づくクラスタリング手法 [2], [3], [4], [5] では Modualrity の値を最適化することでクラスタを抽出する．

Modularity に基づくこれらのクラスタリング手法は比較的大規模なグラフ構造に対しても高速にクラスタを抽出できる手法として知られている．特に，Blondel らによる手法 [4] や Shiokawa らによる手法 [5] では，数億ノード規模以上のグラフ構造をそれぞれ数時間から数分で処理可能としている．しかしながら，Modularity に基づくこれらの手法では，グラフ構造中からハブや外れ値などの役割をもつノードを抽出できない．

5.2 構造的類似度に基づく手法

本稿の対象である構造的類似度に基づくグラフクラスタリング手法 [8], [9], [10], [11], [12], [13] は，データマイニング分野において近年利用され始めているグラフクラスタリング手法である．この手法では事前にクラスタを構成するための類似度の閾値 ϵ とクラスタを構成する最小ノード数 μ をパラメータとして与える．これにより任意の大きさ，形状のクラスタを抽出できるだけではなく，クラスタに併せてハブや外れ値といったノードを抽出することが可能である．

代表的な手法として Xu らによる SCAN [8] が挙げられる．SCAN は多次元ベクトルデータに対する密度ベースのクラスタリング手法としてよく知られている DBSCAN [14] の概念をグラフ構造に適用した手法である．1. 節で述べたように，事前に全てのエッジに対して構造的類似度を計算し，パラメータ ϵ，μ に従い，クラスタの核となる core と定義されるノードを見つける．その後 SCAN は検出された core を中心にクラスタを拡張し，最終的にクラスタに属するノード集合とそれ以外の集合にノードを分類する．クラスタに属さなかったノード集合のうち，2 つ以上のクラスタに接続しているノードについてはハブ，それ以外のノードは外れ値と判定される．

SCAN では全てのエッジに対して構造的類似度の計算を行うことでクラスタリングを実行する．したがって，クラスタリングに対して $O(|E|)$ の時間計算量が必要となる．この計算量は最悪の場合 $O(|V|^2)$ に達する．これらの理由から，大規模なグラフ構造に対し SCAN を適用することは難しい．

また，SCAN の拡張手法として Bortner らによる

(注2)：http://www-personal.umich.edu/~mejn/netdata/
(注3)：http://www.orgnet.com/

SCOT+HintClus [10] および Huang らによる gSkeletonClu [12] が挙げられる．これらの手法では，SCAN の問題点としてパラメータ ϵ 決定の難しさを指摘し，ϵ-free なクラスタリング手法をそれぞれ提案している．SCOT+HintClus では DBSCAN の拡張手法として知られる OPTICS [22] の概念を用いて，構造的類似度に基づくノードの走査順を与えることで準最適な ϵ を見つけ出す．gSkeletonClu では，構造的類似度をエッジの重みとして与えた最小全域木を構築し，この全域木の上でパラメータ μ の制約を満たすようなパラメータ ϵ の候補を抽出する．いずれの手法も SCAN と同様に，事前に全てのエッジに対して構造的類似度が計算されていることを前提としている．そのため，従来手法である SCAN と同程度かそれ以上の計算時間を必要とする．本稿は SCAN に基づく高速なクラスタリング手法について論ずるため，これらの手法で議論されている ϵ-free なクラスタリング手法について本稿では議論の対象としない．

本稿で提案する高速化手法は，本節で述べた関連研究とはことなり，グラフ構造の中からクラスタ集合，ハブ集合，外れ値集合を最高で $O(|V|)$ で抽出する手法である．

6. おわりに

本稿では，大規模なグラフ構造に対する高速かつ正確な構造的類似度に基づくグラフクラスタリングの実現に向けた手法を提案し，その概要について示した．提案手法では，最短ホップ数が 2 となる様なノードに接続したエッジのみを計算対象としてクラスタリングを行うことで，構造的類似度の計算が発生するエッジ数を削減した．提案手法は従来手法のクラスタの定義を満たすことから，提案手法は従来手法の SCAN で生成されるクラスタリング結果と同様の処理結果をより高速に抽出可能とする．本稿で示した計算量分析により，提案手法は $O(|V|)$ の時間計算量で動作する．また実験結果で示したように，提案手法はクラスタ係数が大きな現実のグラフ構造に対してより有効に計算時間を削減するとが可能である．これにより，これまでグラフクラスタリングを用いていたアプリケーションにおける分析の幅や処理性能の向上に本手法は貢献できる．

文　献

[1] Christopher Sibona and Jae Hoon Choi. Factors Affecting End–User Satisfaction on Facebook. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM 2012)*. AAAI Press, 2012.

[2] M. E. J. Newman. Fast Algorithm for Detecting Community Structure in Networks. *Physical Review E - PHYS REV E*, Vol. 69, p. 066133, Jun 2004.

[3] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding Community Structure in Very Large Networks. *Physical Review E - PHYS REV E*, Vol. 70, p. 066111, Dec 2004.

[4] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2008, p. P10008, October 2008.

[5] Hiroaki Shiokawa, Yasuhiro Fujiwara, and Makoto Onizuka. Fast algorithm for modularity-based graph clustering. In *AAAI*, 2013.

[6] Chris H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *ICDM*, pp. 107–114, 2001.

[7] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 22, No. 8, pp. 888–905, 2000.

[8] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *KDD*, pp. 824–833, 2007.

[9] Nurcan Yuruk, Mutlu Mete, Xiaowei Xu, and Thomas A. J. Schweiger. Ahscan: Agglomerative hierarchical structural clustering algorithm for networks. In *ASONAM*, pp. 72–77, 2009.

[10] Dustin Bortner and Jiawei Han. Progressive clustering of networks using structure-connected order of traversal. In *ICDE*, pp. 653–656, 2010.

[11] Heli Sun, Jianbin Huang, Jiawei Han, Hongbo Deng, Peixiang Zhao, and Boqin Feng. gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration. In *ICDM*, pp. 481–490, 2010.

[12] Jianbin Huang, Heli Sun, Qinbao Song, Hongbo Deng, and Jiawei Han. Revealing density-based clustering structure from the core-connected tree of a network. *IEEE Trans. Knowl. Data Eng.*, Vol. 25, No. 8, pp. 1876–1889, 2013.

[13] Weizhong Zhao, Venkata Swamy Martha, and Xiaowei Xu. Pscan: A parallel structural clustering algorithm for big networks in mapreduce. In *AINA*, pp. 862–869, 2013.

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pp. 226–231, 1996.

[15] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, Vol. 1, No. 1, March 2007.

[16] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pp. 177–187, New York, NY, USA, 2005. ACM.

[17] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *CoRR*, Vol. abs/0810.1355, , 2008.

[18] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, pp. 3:1–3:8, New York, NY, USA, 2012. ACM.

[19] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 1361–1370, New York, NY, USA, 2010. ACM.

[20] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, Vol. 2, No. 1, pp. 193–218, 1985.

[21] M. E. J. Newman and M. Girvan. Finding and Evaluating Community Structure in Networks. *Physical Review E - PHYS REV E*, Vol. 69, p. 026113, Feb 2004.

[22] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. In *SIGMOD Conference*, pp. 49–60, 1999.