DEIM2021 B13-2

ランダムウォークサンプリングに基づくソーシャルグラフの復元

中嶋　一貴†　　首藤　一幸†

† 東京工業大学 情報理工学院 数理・計算科学系

あらまし　ソーシャルグラフの正確で詳細な分析は，グラフデータへのアクセス制限のために，データ保有者でない研究者にとって挑戦的な課題である．この課題に対処するために，ランダムウォークを介して様々なグラフ統計量の不偏推定量を得るアルゴリズムが研究されてきた．しかし，ほとんどの既存アルゴリズムには，ソーシャルグラフの分析範囲を制限するという問題点がある．本研究では，ランダムウォークで得られたサンプルから元のソーシャルグラフを復元するグラフ復元手法を提案する．提案手法は，推定した局所的統計量とサンプリングで得られた真の構造情報の両方を反映したグラフを生成する．反映するターゲットの局所的統計量は，$d = 0, 1, 2,$ または 2.5 のいずれかの次元 d のノードの統計量である dK 統計量に対応する．提案手法では，生成グラフがターゲットの dK 統計量を満たすように，サンプリングした部分グラフのノードとエッジを補完する．$2.5K$ 統計量をターゲットとする提案手法が，比較したすべての手法の中で，実世界のソーシャルグラフの局所的および大域的な 12 個の統計量と視覚的表現を最も正確に捉えた生成グラフを実現することを示す．サンプリングした部分グラフから元のグラフを復元するという我々の方法は，データへのアクセスが制限されたソーシャルグラフの正確な分析範囲を大幅に改善する．

キーワード　ソーシャルグラフ，グラフサンプリング，ランダムウォーク，グラフ復元

1　はじめに

ソーシャルグラフの分析は，人間のつながりや行動などの世界規模の社会構造を理解する上で大きな役割を果たしてきた．一般的に，グラフデータの保有者でない研究者は，ソーシャルグラフの分析のためにデータのサンプリングを試みる．特に，幅優先探索やランダムウォークなどのクローリング手法は，ノードにクエリを実行して隣接情報を取得できるソーシャルグラフにおいて有効なサンプリング手法である [1–4]．ただし，クローリング手法は一般的に次数が高いノードに偏ったサンプルを招く [3]．このため，少量のデータのサンプルに基づくソーシャルグラフの正確かつ詳細な分析は挑戦的な課題である．

Gjoka らは，高次数ノードへのサンプリングのバイアスに対処する重み付け手法という枠組みを提案した [3]．この枠組みでは，ランダムウォークにより得られた各サンプルに重みを付けて，マルコフ性から導出されるサンプリングのバイアスを修正する．実用的なシナリオでは，現実的な時間内に実行できるクエリの回数が限られているため，少数のクエリを使用して統計量を正確に推定することが重要である [5, 6]．このため，過去 10 年間に，少数のクエリを使用して様々なグラフ統計量の不偏推定量を得る重み付け手法が研究されてきた [3, 5–8]．

しかし，重み付け手法ではソーシャルグラフの分析範囲に限界がある．まず，重み付け手法は各統計量に対して個別に設計する必要があるため，ソーシャルグラフ構造に関する分析者の幅広い関心に対処できない．そして，重み付け手法は，最短経路などの大域的な統計量の推定に適さない．なぜなら，各サンプルの重みを計算するためにほとんどのグラフデータをサンプリングしなければならないからである．さらに，重み付け手法ではグラフ統計量の推定以外の分析を原理的に実行できない．例えば，研究者は重み付け手法を用いて，ソーシャルグラフ分析で重要な課題の 1 つであるグラフの可視化 [9] を実行できない．

我々は，この問題点を解決するために，ランダムウォークによるサンプルに基づいて生成したグラフを分析する手法を研究する．生成されたグラフの分析範囲に制限はない: (i) 統計量ごとに分析手法の設計は必要ない．(ii) 生成グラフの統計量を局所的範囲から大域的範囲まで自由に分析できる．(iii) グラフの可視化をはじめとする，グラフを入力とする分析を実行できる．我々の目標は，ランダムウォークによる少量のサンプルから元のソーシャルグラフを復元する方法を模索し，ソーシャルグラフの正確で詳細な分析を可能にすることである．

我々は，dK シリーズの枠組み [8, 10, 11] に従って，ソーシャルグラフを復元する手法を開発する．dK シリーズは，与えられたグラフの次元 d のノードの統計量を満たす生成グラフの集合である．d の値が増加すると，dK 統計量を満たす生成グラフは，目標のグラフのより詳細な構造を捉える．大規模なグラフに対して，$d = 0, 1, 2$ または 2.5 のいずれかの dK 統計量を指定したグラフを現実的に生成できる [8, 10, 11]．重み付け手法の発展により，ランダムウォークによる少量のサンプルから $d = 0, 1, 2$ または 2.5 の dK 統計量を正確に推定することが可能となった．したがって，ランダムウォークにより推定された dK 統計量を反映したグラフを生成することが可能である．

Gjoka らは，ランダムウォークによるサンプルから $2.5K$ 統計量を推定し，それを反映したグラフを生成する手法を提案した [8]．彼らは，その生成グラフが少量のサンプルで元のソーシャルグラフの局所的および大域的な統計量を正確に捉えることを示した．ただし，Gjoka らの手法は，サンプリングによる

：This is an unrefereed paper.

推定の本質的な特性である一致性 (consistency) を欠いている．つまり，推定した dK 統計量のみを反映した生成グラフは，いくらサンプル数を増やしても元のグラフに一致しない．これは，生成グラフが元のグラフの真の構造情報を含まない，単なる類似したグラフに過ぎないからである．

本研究の貢献: 本研究では，グラフ復元手法を提案する．グラフ復元手法は，$d = 0, 1, 2$ または 2.5 のいずれかの推定した dK 統計量だけでなく，サンプリングによって得られた真の構造情報も反映したグラフを生成する．提案手法では，最初に不偏推定量に基づいてターゲットの dK 統計量を作成する: 推定値は，通常，dK 統計量を満たすグラフの生成条件を満たさないからである．次にターゲットの dK 統計量を満たすように，サンプリングした部分グラフにノードとエッジを補完する．部分グラフに含まれる構造情報は常に真であるため，提案手法は Gjoka らの手法と比較して 2 つの利点を持つ: (i) 生成グラフは，サンプル数が増加するにつれて元のグラフに収束する．(ii) 推定する必要のある元のグラフ構造の範囲が削減される．

9 つの実世界のソーシャルグラフデータセットを使用した広範な実験を通じて，提案手法の有効性を明らかにする．提案手法を従来の部分グラフ手法および Gjoka らの手法 [8] と比較する．各手法を 12 個の基本的なグラフ統計量の精度，グラフの視覚的表現，およびグラフの生成時間の 3 つの観点で評価する．評価実験により，$2.5K$ 統計量をターゲットとする提案手法が，実用的な生成時間で，12 個全ての統計量と視覚的表現を最も正確に捉えた生成グラフを実現することを示す．

2 関 連 研 究

ソーシャルグラフの初期の研究では，幅優先探索などのクローリング手法を用いてサンプリングした部分グラフを分析することが一般的であった [1, 2, 4]．得られた部分グラフは，代表的なサンプルであると暗黙的に想定されていた．しかし，クローリングによる少数のサンプルは，一般的に高次数ノードに偏る [3]．本研究では，少数のサンプルに基づく部分グラフは，統計量の誤差が大きく，視覚的表現で低次数ノードで構成される周辺構造を捉えられないことを示す．

Gjoka らは，ランダムウォークを介して高次数ノードへのバイアスに対処する重み付け手法という枠組みを提案した．過去 10 年間で，重み付け手法に基づいてさまざまな統計量の不偏推定量を得るアルゴリズムが研究されてきた [3, 5–8]．本研究では，重み付け手法ではソーシャルグラフの分析範囲に限界があるという問題点に焦点を当てる: (i) 統計量ごとに個別の重み付け手法を設計する必要がある．(ii) 最短経路などの大域的な統計量の推定に適さない．(iii) グラフ統計量の推定以外の分析を原理的に実行できない．

分析範囲の制限を取り除く 1 つの方法は，クローリングを通じたグラフ生成に基づく分析手法である: 生成グラフの分析範囲に制限は無い．Gjoka らは，ランダムウォークにより $2.5K$ 統計量を推定し，それを反映したグラフを生成する手法を提案した [8]．Gjoka らは次の 2 つを明らかにした: (i) $2.5K$ 統計量の不偏推定量をランダムウォークを介して取得できる．(ii) 推定した $2.5K$ 統計量を反映した生成グラフは，局所的および大域的な主要統計量を正確に捉える．彼らは，サンプリングを通じたグラフ生成に基づく分析手法の可能性を初めて示唆した．しかし，その有望な可能性にもかかわらず，グラフ生成に基づく分析手法はこれまでほとんど研究されてこなかった．本研究では，グラフ復元手法を提案する．提案手法は，$d = 0, 1, 2,$ または 2.5 のいずれかの推定した dK 統計量だけでなく，サンプリングによって得られた真の構造情報も反映したグラフを生成する．提案手法は，Gjoka らの手法と比較して，生成グラフの統計量の精度と視覚的表現を劇的に改善する．

実世界のグラフのさまざまな主要統計量を捉えるグラフの生成モデルは，長く研究されてきた [12]．一般的なグラフ生成モデルはすべてのグラフデータが利用可能であることを想定しているため，任意のグラフ生成モデルをソーシャルグラフ復元の問題に適用することは次の 3 つの理由で自明ではない．まず，サンプリングを通じてモデルへの入力を推定する必要がある．そして，推定値に基づいてグラフの生成条件を満たすモデルへの入力を作成する必要がある．さらに，一般的な生成モデルは空のグラフから生成プロセスを開始する一方で，ソーシャルグラフ復元問題ではサンプリングした部分グラフから生成プロセスを開始する．本研究では，dK シリーズ [10] をソーシャルグラフの復元に適用できることを明らかにする．

ソーシャルグラフの復元は，与えられた行列の要素を補完する行列補完 [13] や，与えられたグラフのノードまたはエッジを補完するリンク検出 [14] およびネットワーク補完 [15] に関連している．ただし，ソーシャルグラフをクローリングして得られる部分グラフは，低い次数ノードに偏った欠測ノードを引き起こす．これは，上記の問題における欠測データの想定とは根本的に異なる．本研究では，クローリング手法によって得られる部分グラフのノードとエッジを補完する手法を設計する．

3 準　　備

まず，表記法を説明し，ソーシャルグラフ復元の問題を定義する．そして，部分グラフ手法，dK シリーズ，dK 統計量 ($d = 0, 1, 2, 2.5$) の不偏推定量を紹介する．

3.1 表記と問題定義

ソーシャルグラフを連結な無向グラフ $G = (V, E)$ で表す．$V = \{v_1, \ldots, v_n\}$ はノード (=ユーザ) の集合，E はエッジ (=ユーザ間の友好関係) の集合である．n と m をそれぞれノード数とエッジ数とする．G の隣接行列を $\boldsymbol{A}$ で表す．ここではループと多重辺を許可するため，要素 $A_{i,j} = A_{j,i}$ はノード v_i と v_j の間のエッジ数である．慣例に従って，ループを 2 本のエッジとして数える [16]．$N(i)$ は v_i の重複を含む隣接ノード集合とし，d_i を v_i の次数とする．$V(k)$ は次数 k を持つノード集合とする．$1_{\{cond\}}$ は条件 $cond$ が成立する場合に 1 を返し，それ以外の場合は 0 を返す指示関数である．

グラフ G のアクセスモデルは，Gjoka らの研究 [3] に従う:

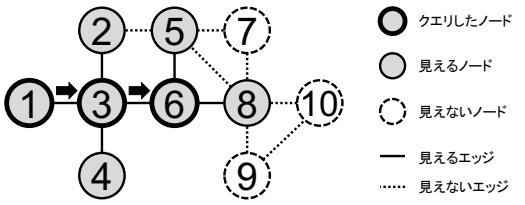


図 1　グラフ上のランダムウォークでノード 1, 3, 6 の順で遷移した例.

(i) ノード v_i にクエリを実行すると，v_i の隣接ノード集合 $N(i)$ を利用できる．(ii) すべてのグラフデータの取得及びそれらへのランダムアクセスは考慮しない．(iii) クローリングを開始する初期ノードを任意に選ぶことができる．(iv) G は静的である．

ランダムウォークは，ノードの隣接関係を利用できるソーシャルグラフにおいて効果的なサンプリング手法である．ランダムウォークでは，初期ノード v_{x_1} から開始して 1 つの隣接ノードへのランダムな遷移を $r-1$ 回繰り返す．ここで，x_s は s 番目にサンプリングされたノードのインデックスを示す．すると，r 個のサンプルノードのインデックスと隣接ノード集合のリストを得る．このリストを $R=\{(x_s, N(x_s))\}_{s=1}^{r}$ で表す．

次の問題を研究する: r 個のサンプルノードのインデックスと隣接ノード集合のリスト R が与えられ，元のグラフ G に対して可能な限り近い構造を有するグラフを生成する．究極の目標は，サンプル R から G を完全に復元することである．

3.2　部分グラフ手法

部分グラフ手法は，サンプリングによりグラフ統計量を推定する基本的な手法である [1, 2, 4]．部分グラフ手法は，サンプリングを通じて見えるエッジの集合 E' の誘導部分グラフ $G'=(V', E')$ を構築する．ここで，V' は 2 つの互いに素な集合 V^{qry} と V^{vis} の和集合である．V^{qry} はクエリしたノードの集合，V^{vis} はクエリノードの隣接ノードとして見えるノードの集合である．部分グラフ手法は，任意のクローリング手法を介して適用可能である．図 1 は，グラフ上のランダムウォークでノード 1, 3, 6 の順序で遷移した例を示す．この例では，$V^{qry}=\{1,3,6\}$，$V^{vis}=\{2,4,5,8\}$，$E'=\{(1,3),(2,3),(3,4),(3,6),(5,6),(6,8)\}$ である．

3.3　dK シリーズ

dK シリーズは，与えられたグラフの d ノードから成る部分グラフ集合のすべての同時次数分布を指定したグラフの有限集合である．dK グラフは，与えられたグラフの dK 統計量を満たすグラフの集合を表す．$0K$ 統計量は，与えられたグラフのノード数 n と平均次数 $\bar{k}=\frac{2m}{n}$ を指定する．$1K$ 統計量は，与えられたグラフのノード数 n と次数分布 $\{P(k)=\frac{n(k)}{n}\}_k$ を指定する．ここで，$n(k)=|V(k)|$ は，次数 k のノード数である．$2K$ 統計量は，与えられたグラフのエッジ数 m と同時次数分布 $\{P(k,k')=\frac{m(k,k')}{2m}\}_{k,k'}$ を指定する．ここで，$m(k,k')=\sum_{v_i\in V(k)}\sum_{v_j\in V(k')}A_{i,j}$ は次数 k と k' のノード間のエッジ数である．$3K$ 統計量は，次数 k, k', k'' のノードで構成される三角形とくさびの数を指定する．重要な特徴の 1 つは，dK グラフが各次元 $i=0,...,d$ の iK 統計量を満たすことである．もう 1 つの特徴は収束性である: nK グラフは与えられたグラフと同型である．

Mahadevan らは，数千ノードの実世界のグラフに対して，$3K$ グラフが局所的および大域的規模の両方で構造的特性を正確に捉えることを示した [10]．残念ながら，現在，大規模ネットワークに対して $d\geqq 3$ に対する dK グラフを生成する効率的なアルゴリズムは存在しない [10, 11]．$2K$ グラフは大規模なグラフに対して効率的に生成できるが，三角形の統計量などいくつかの重要な統計量を捉えられない [8, 11]．

$2.5K$ グラフは，局所的および大域的なグラフ統計量の精度とグラフサイズに対するスケーラビリティの良好なトレードオフを達成する．$2.5K$ 統計量は，与えられたグラフのエッジの数 m，同時次数分布 $\{P(k,k')\}_{k,k'}$，および次数依存のクラスタ係数 $\{\bar{c}(k)=\frac{1}{n(k)}\sum_{v_i\in V(k)}\frac{2t_i}{k(k-1)}\}_k$ を指定する．ここで，$t_i=\sum_{j\neq i,k\neq i,j<k}A_{i,j}A_{i,k}A_{j,k}$ は，v_i が属する三角形の数であり，$\bar{c}(0)=\bar{c}(1)=0$ とする．

3.4　dK 統計量の不偏推定量

ランダムウォークによりターゲットの dK 統計量を推定する．既存の重み付け手法の枠組みに基づいて，$0K, 1K, 2K$ および $2.5K$ 統計量の各不偏推定量を得る: ノード数 [5, 7], 平均次数 [3, 6], 次数分布 [3], 同時次数分布 [8], 次数依存のクラスタ係数 [5]．以下では，ノード数，平均次数，次数分布，同時次数分布，次数依存のクラスタ係数のそれぞれの不偏推定量を $\hat{n}, \hat{\bar{k}}, \{\hat{P}(k)\}_k, \{\hat{P}(k,k')\}_{k,k'}, \{\hat{\bar{c}}(k)\}_k$ で表す．

4　提 案 手 法

本章では，グラフ復元手法を提案する．グラフ復元手法は，$d=0$, 1, 2 または 2.5 のターゲットの次元 d とサンプリングリスト R を受け取り，生成グラフ $\tilde{G}=(\tilde{V}, \tilde{E})$ を返す．まず，ターゲットの d に依らずに，リスト R から部分グラフ $G'=(V', E')$ を構築する．見えるノードの集合 V^{vis} が空集合でない限り，部分グラフ G' のノードとエッジの補完を実行する．

提案手法は，3 つの主要な段階に分かれる．

（1）　$d'K$ 統計量の不偏推定量に基づいて，ターゲットの $d'K$ 統計量を作成する．ここで，$d=0,1,2$ の場合，$d'=d$ であり，$d=2.5$ の場合，$d'=2$ である．

（2）　部分グラフ G' の真の構造情報に従って，(1) で作成したターゲットの $d'K$ 統計量を修正する．

（3）　ターゲットの $d'K$ 統計量を満たすように G' にノードとエッジを追加する．さらに $d=2.5$ の場合，推定した $2.5K$ 統計量に近づくように $\tilde{G}$ のエッジを繰り返し再配線する．

以下では，$0K, 1K, 2K$, および $2.5K$ 統計量それぞれをターゲットとする手法を順に設計する．

4.1　$0K$ 統計量をターゲットとする手法

まず，$0K$ 統計量をターゲットとする手法を設計する．$0K$ 統計量は，$m=\frac{1}{2}n\bar{k}$ より，ノード数とエッジ数を指定することと同等である．ターゲットのノード数 n^* とエッジ数 m^* はそれ

ぞれ非負の整数であれば，それを満たすグラフを生成できる．

ノード数と平均次数のそれぞれの不偏推定量 $\hat{n}$ と $\hat{\bar{k}}$ からターゲットの n^* と m^* を以下のように定める:

$$n^* = \max(\mathrm{NearInt}(\hat{n}), 1),$$
$$m^* = \max\left(\mathrm{NearInt}\left(\frac{1}{2}\hat{n}\hat{\bar{k}}\right), 1\right).$$

ここで，$\mathrm{NearInt}(x)$ は x に最も近い整数を返す関数，$\max(a,b)$ は整数 a,b のうち大きい方を返す関数を表す．推定値が得られたならばノードとエッジは少なくとも 1 つあることを仮定する．

さらに，部分グラフ G' に既に $n' = |V'|$ ノードおよび $m' = |E'|$ エッジが含まれるという構造情報を利用すると，より妥当なターゲットの n^* と m^* を設定できる:

$$n^* = \max(\mathrm{NearInt}(\hat{n}), n'),$$
$$m^* = \max\left(\mathrm{NearInt}\left(\frac{1}{2}\hat{n}\hat{\bar{k}}\right), m'\right).$$

次に，部分グラフにノードとエッジを追加して，生成グラフが最終的に n^* ノードと m^* エッジを持つようにする．$\tilde{G}$ にはすでに n' ノードと m' エッジがあるという事実に加えて，部分グラフ G' のクエリしたノードにエッジを追加すべきでないことを示唆する次の補題を考慮する:

補題 1. 部分グラフの各ノード $\tilde{v}_i \in V'$ の真の次数 d_i と部分グラフにおける次数 d'_i は以下の関係を満たす:

$$d_i = d'_i \quad (\text{if } \tilde{v}_i \in V^{qry}),$$
$$d_i \geqq d'_i \quad (\text{if } \tilde{v}_i \in V^{vis}).$$

Proof. 最初の等式は，ノードをクエリするとすべての隣接ノードを取得できる前提より正しい．次の不等式は，見えるノードはクエリされた隣接ノードしか持たないことより正しい． □

従って，最初に $n^* - n'$ ノードを部分グラフに追加し，次に，どちらもクエリしたノードでないような 2 つのノードをランダムに選びエッジを追加することを $m^* - m'$ 回繰り返す．

4.2 $1K$ 統計量をターゲットとする手法

次に $1K$ 統計量をターゲットとする手法を設計する．$1K$ 統計量は，$n(k) = nP(k)$ より，各次数 k を持つノード数 $\{n(k)\}_k$ を指定することと同等である．ターゲットの $\{n^*(k)\}_k$ は以下の条件を満たせば，それを満たすグラフを生成可能である [17]:

（1） 各次数 k に対して $n^*(k)$ は非負の整数.

（2） 次数の総和 $\sum_k kn^*(k)$ は偶数.

4.2.1 不偏推定量に基づくターゲットの $1K$ 統計量の作成

初期化ステップ: まず，ノード数と次数分布のそれぞれの不偏推定量 $\hat{n}$ と $\{\hat{P}(k)\}_k$ から，条件 (1) を満たすように，各次数 $k \in D_{1K}$ のターゲットのノード数 $n^*(k)$ を初期化する:

$$n^*(k) = \max(\mathrm{NearInt}(\hat{n}\hat{P}(k)), 1).$$

ここで，$D_{1K} = \{k \mid \hat{P}(k) > 0\}$ は分布の推定値 $\hat{P}(k) > 0$ であるような次数 k の集合である．

調整ステップ: 次に，初期化した $\{n^*(k)\}_k$ に対して，次数の総和 $\sum_k = kn^*(k)$ が奇数であればその調整を行う．推定値の相対誤差の増加 $\Delta e^{1K}(k)$ が最小であるような奇数の次数 $k \in D_{1K}$ に対して $n^*(k)$ を 1 増やす．ここで

$$\Delta e^{1K}(k) = \frac{|\hat{n}(k) - (n^*(k)+1)|}{\hat{n}(k)} - \frac{|\hat{n}(k) - n^*(k)|}{\hat{n}(k)}$$

と定義する．誤差の増加 $\Delta e^{1K}(k)$ が等しい奇数の次数 k の候補が複数ある場合，生成グラフのエッジ数の増加を最小限にするために，それらの中から最小の次数を選ぶ．次数 k の候補が存在しない場合，$n^*(1) = 1$ として次数の合計を 1 増やす．

4.2.2 ターゲットの $1K$ 統計量の修正

ターゲットの $1K$ 統計量を満たすグラフを生成するために，生成グラフ $\tilde{G}$ の各ノードにターゲットの次数を割り当てる [10]．まず，次数に制約のある部分グラフ内の各ノードにターゲットの次数を割り当てる．次数の割り当てを行いながら，割り当てた各次数のノード数がターゲットの次数のノード数を超えないように修正する．

まず，補題に従って，クエリしたノード $\tilde{v}_i \in V^{qry}$ に部分グラフにおける次数と等しいターゲットの次数を割り当てる:

$$d_i^* = d'_i.$$

ここで，d_i^* はノード $\tilde{v}_i$ のターゲットの次数を表す．各見えるノード $\tilde{v}_i \in V^{vis}$ には，部分グラフにおける次数以上のターゲットの次数を割り当てる:

$$d_i^* \geqq d'_i.$$

部分グラフの全ノードに次数を割り当てた後，ターゲットの $\{n^*(k)\}_k$ を 4.2.1 章の調整アルゴリズムに従って再度修正する．

4.2.3 部分グラフへのノードとエッジの追加

最後に，生成グラフがターゲットの $\{n^*(k)\}_k$ を満たすように，ノードとエッジを部分グラフに追加する．まず，$\sum_k n^*(k) - n'$ ノードを部分グラフに追加する．V^{add} を部分グラフに追加したノード集合とする．次に，V^{add} の各ノードにターゲットの次数を任意に割り当てる．そして，各ノード $\tilde{v}_i \in \tilde{V}$ に $d_i^* - \tilde{d}_i$ 本のエッジの片割れ (スタブと呼ばれる [17]) を持たせる．ここで，$\tilde{d}_i$ は現時点での次数であり，各ノード $\tilde{v}_i \in V^{qry} \cup V^{vis}$ に対して $\tilde{d}_i = d'_i$ であり，各ノード $\tilde{v}_i \in V^{add}$ に対して $\tilde{d}_i = 0$ である．$\tilde{G}$ に空のスタブがなくなるまで，空のスタブを持つノードペアをランダムに繰り返し接続する．

4.3 $2K$ 統計量をターゲットとする手法

$2K$ 統計量をターゲットとする手法を設計する．$2K$ 統計量は，$m(k,k') = 2mP(k,k')$ より，各同時次数を持つエッジ数 $\{m(k,k')\}_{k,k'}$ と同等である．各同時次数を持つターゲットのエッジ数 $\{m^*(k,k')\}_{k,k'}$ は以下の条件を満たす必要がある [18]:

（1） 各同時次数 k,k' に対して $m^*(k,k')$ は非負の整数.

（2） 各次数 k に対して $m^*(k,k)$ は偶数.

（3） 各同時次数 k,k' に対して $m^*(k,k') = m^*(k',k)$.

（4） 各次数 k に対して $\sum_{k'} m^*(k,k') = kn^*(k)$.

4.3.1 不偏推定量に基づくターゲットの $2K$ 統計量の作成

初期化ステップ: まず, ノード数, 平均次数, 次数分布, 同時次数分布のそれぞれの不偏推定量 $\hat{n}, \hat{\bar{k}}, \{\hat{P}(k)\}_k, \{\hat{P}(k,k')\}_{k,k'}$ に基づいて, 条件 (1), (2), (3) を満たすように各同時次数のターゲットのエッジ数を初期化する. $D_{2K} = \{(k,k') \mid \hat{P}(k,k') > 0\}$ を推定分布 $\hat{P}(k,k') > 0$ であるような同時次数の集合とする. $k \neq k'$ であるような各同時次数 $(k,k') \in D_{2K}$ に対して,

$$m^*(k,k') = \max(\mathrm{NearInt}(\hat{n}\hat{\bar{k}}\hat{P}(k,k')), 1),$$

および, 各同時次数 $(k,k) \in D_{2K}$ に対して,

$$m^*(k,k) = \max(\mathrm{NearEven}(\hat{n}\hat{\bar{k}}\hat{P}(k,k)), 2),$$

と初期化する. ここで, $\mathrm{NearEven}(x)$ は x に最も近い偶数を返す関数である.

調整ステップ: 次に, 条件 (4) を満たすように, 初期化した $\{m^*(k,k')\}_{k,k'}$ を加減により調整する. ここで, 条件 (1) の違反を防ぐために, 下限値 $\{m_{min}(k,k')\}_{k,k'}$ を設定する. また, 条件 (2), (3) の違反を防ぐために, 要素 $m^*(k,k')$ の増減と同じ分だけ要素 $m^*(k',k)$ も増減する. これにより, $k \neq k'$ の場合は $\{m^*(k,k')\}_{k,k'}$ の対称性が保持され, $k = k'$ の場合は $m^*(k,k)$ が偶数であることが保持される.

各 $k \in D$ に対して, 現在の合計 $s_{cur}(k) = \sum_{k'} m^*(k,k')$ がターゲットの合計 $s_{tgt}(k) = kn^*(k)$ に等しくなるように, ある次数 $k' \in D$ に対して $m^*(k,k')$ を 1 ずつ増減することを繰り返す. ここで $D = \{k \mid n^*(k) > 0 \vee k = 1\}$ であり, $\{n^*(k)\}_k$ は各次数のターゲットのノード数である.

合計を調整する次数の順序を $k_1, k_2, ..., k_{|D|}$ とする. i 番目の次数 k_i の合計を調整するとき, それ以前に合計を調整した各次数 $k' \in \{k_1, ..., k_{i-1}\}$ に対する要素 $m^*(k_i,k')$ の変更を禁止する. この規則は, i 番目のステップの終わりに条件 (4) が次数 $k_1, ..., k_i$ に対して成り立つことを保証する. 従って, 最後の $i = |D|$ ステップの終了時に, 各同時次数のターゲットのエッジ数は条件 (4) を満たす.

4.3.2 ターゲットの $2K$ 統計量の修正

ターゲットの $2K$ 統計量を作成した後, 4.2.2 章に従って, 部分グラフ内の各ノードにターゲットの次数を割り当てる. そして, 部分グラフの各ノードに割り当てられたターゲットの次数 d_i^* に基づいて, 部分グラフの各エッジ $(\tilde{v}_i, \tilde{v}_j) \in E'$ にターゲットの同時次数 (d_i^*, d_j^*) を割り当てる. この割り当ての後, 各同時次数のターゲットのエッジ数 $\{m^*(k,k')\}_{k,k'}$ を 4.3.1 章の調整アルゴリズムに従って再度調整する.

4.3.3 部分グラフへのノードとエッジの追加

最後に, ノードとエッジを部分グラフに追加して, 生成グラフ $\tilde{G}$ が各同時次数の目標のエッジ数 $\{m^*(k,k')\}_{k,k'}$ を満たすようにする. まず, $\sum_k n^*(k) - n'$ ノードを部分グラフに追加する. 次に, 追加された各ノードにターゲットの次数を任意に割り当てる. 第三に, 各ノード $\tilde{v}_i \in \tilde{V}$ に $d_i^* - \tilde{d}_i$ 本のスタブを持たせる. そして, 各同時次数 k および k' のターゲットのエッジ数に達するまで, 次数 k および k' のノードの空のスタブをランダムに繰り返し接続する.

表 1 データセット.

グラフ名	ノード数	エッジ数
Anybeat [19]	12,645	49,132
Enron [20]	33,696	180,811
Brightkite [19]	56,739	212,945
Douban [19]	154,908	327,162
Epinions [20]	75,877	405,739
Slashdot [19]	77,360	469,180
Facebook [19]	63,392	816,886
Gowalla [19]	196,591	950,327
Livemocha [19]	104,103	2,193,083

4.4 2.5K 統計量をターゲットとする手法

最後に $2.5K$ 統計量をターゲットとする手法を設計する. まず 4.3 章に従ってターゲットの $2K$ 統計量を満たすグラフ $\tilde{G}$ を生成する. 次に推定した次数依存のクラスタ係数 $\{\hat{\bar{c}}(k)\}_k$ に近づくようにエッジを繰り返し再配線する. エッジの再配線には, 既存研究 [11] のアルゴリズムを用いる. ただし, エッジの再配線では, 部分グラフに含まれるエッジの再配線を禁止することによって, 部分グラフの構造情報を厳密に保持する.

5 実　　験

生成グラフの 12 個の基本的なグラフ統計量の精度, 生成グラフの視覚表現, および生成時間の観点から, 提案手法を評価する. 実験の目標は, 以下の 3 つの質問に答えることである.

- 提案手法は, 実世界のソーシャルグラフの統計量と視覚的表現を既存手法より正確に捉える生成グラフを実現するか?
- $d = 0$, 1, 2, または 2.5 の dK 統計量をターゲットとする提案手法のうち, 最も正確に統計量と視覚的表現を捉える生成グラフを実現する手法はどれか?
- 提案手法は, 既存手法と比較してどのくらい効率的にグラフを生成するか?

5.1 実 験 準 備

データセット: 公開された 9 個のソーシャルグラフデータセットを用いる. すべてのグラフを連結された無向グラフとして扱う. 表 1 に各グラフのノード数とエッジ数を示す.

対象とするグラフ統計量: 局所的範囲および大域的範囲のグラフ構造を捉える 12 個の統計量に焦点を当てる:

(1) ノード数: n.

(2) 平均次数: $\bar{k} = \frac{2m}{n}$.

(3) 次数分布: $\{P(k)\}_k$.

(4) 次数相関: $\{\bar{k}_{nn}(k) = \frac{1}{n(k)} \sum_{v_i \in V(k)} \frac{1}{k} \sum_{v_j \in V} A_{i,j} d_j\}_k$.

(5) 平均クラスタ係数: $\bar{c} = \frac{1}{n} \sum_{v_i \in V} \frac{2t_i}{d_i(d_i-1)}$.

(6) 次数依存の平均クラスタ係数: $\{\bar{c}(k)\}_k$.

(7) エッジの共有ノード数分布: $\{P(s)\}_s$. $P(s) = \frac{1}{m} \sum_{(v_i,v_j) \in E, i<j} 1_{\{sp(i,j)=s\}}$. $sp(i,j) = \sum_{v_k \in V, k \neq i,j} A_{i,k} A_{j,k}$.

(8) 平均最短距離: $\bar{l} = \frac{2}{n(n-1)} \sum_{v_i,v_j \in V, i<j} l_{i,j}$. $l_{i,j}$ は v_i と v_j の間の最短距離である.

(9) 最短距離分布: $\{P(l) = \frac{2}{n(n-1)} \sum_{v_i,v_j \in V, i<j} 1_{\{l_{i,j}=l\}}\}_l$.

（10） 直径: d.

（11） 次数依存の媒介中心性: $\{\bar{b}(k) = \frac{1}{n(k)} \sum_{v_i \in V(k)} b_i\}_k$. b_i は v_i の媒介中心性である.

（12） 隣接行列 $\boldsymbol{A}$ の最大固有値: λ_1.

誤差指標: 統計量の誤差指標は，正規化平均絶対誤差 (Normalized Mean Absolute Error, NMAE) [8] を用いる．$\boldsymbol{x}, \hat{\boldsymbol{x}}$ をそれぞれ元のグラフと生成グラフの統計量を表すベクトルとすると，$\mathrm{NMAE}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \frac{\sum_i |x_i - \tilde{x}_i|}{\sum_i x_i}$ と定義される.

比較する手法: 4 つの手法を比較する:

（1） 幅優先探索 (Breadth-first search, BFS) による部分グラフ手法.

（2） ランダムウォーク (Random walk, RW) による部分グラフ手法.

（3） $d = 0$, 1, 2, または 2.5 いずれかの dK 統計量をターゲットとする Gjoka らの手法 [8].

（4） $d = 0$, 1, 2, または 2.5 いずれかの dK 統計量をターゲットとする提案手法.

公平な比較のために，(i) 同じ初期ノードから幅優先探索とランダムウォークを実行する．(ii) ランダムウォークで得られた同一のサンプルを用いて手法 (2), (3), (4) を実行する．初期ノードは，各実験で独立に，すべてのノードからランダムに選択される．ノードの総数に対するクエリしたノードの比率 (= サンプル率) が目標値に達するまで，BFS と RW を実行する.

5.2 グラフ統計量の精度

まず，推定した dK 統計量とサンプリングで得た真の構造情報の両方を生成グラフに反映する提案手法の有効性を検証する．図 2 は，Anybeat グラフにおいて RW に基づく 3 つの手法の 12 個の統計量にわたる平均 NMAE を示す．3 つの手法は，RW による部分グラフ手法，dK 統計量をターゲットとする Gjoka らの手法，および dK 統計量をターゲットとする提案手法である．サンプル率は 1%から 50%まで変化させた．RW による部分グラフ手法の平均 NMAE は，ターゲットの dK 統計量に関係なく，図 2(a)–(d) で同一である.

$1K$, $2K$, および $2.5K$ 統計量をターゲットとする提案手法は，サンプル率が特に 10%未満の場合に，RW による部分グラフ手法の平均 NMAE を改善した (図 2(b)–(d))．部分グラフにノードとエッジをランダムに追加する $0K$ をターゲットとする手法が精度を改善しないことは，部分グラフのノードとエッジの補完が自明ではないことを示唆している (図 2(a))．サンプル率が低い場合の提案手法による精度の改善は，データへのアクセスが制限されるソーシャルグラフにおいて意義を持つ.

各 dK 統計量をターゲットとする提案手法は，すべてのサンプル率において Gjoka らの手法の平均 NMAE を改善した．推定した $0K$ または $1K$ 統計量のみを反映した生成グラフは，元のグラフ統計量を正確に捉えない (図 2(a)–(b))．しかし，生成グラフに真の構造情報も反映させると，サンプル率 10%のときの平均 NMAE がそれぞれ 82.6%, 91.0%改善した．$2K$ と $2.5K$ をターゲットとした場合，提案手法はサンプル率 10%のときの平均 NMAE をそれぞれ 37.9%と 47.0%改善した (図 2(c)–(d))．さらに，サンプル率が増加すると提案手法の平均 NMAE は 0 に収束する．これは，提案手法による生成グラフには，サンプリングした真の構造情報が反映されるためである.

次に，各 dK 統計量をターゲットとする提案手法による生成グラフの統計量の精度を比較する．図 3 は，4 つのグラフにおいて，各 $0K$, $1K$, $2K$, および $2.5K$ をターゲットとする提案手法の 12 個の統計量にわたる平均 NMAE を示す．サンプル率は 1%から 10%まで変化させた．提案手法は，より大きな次元 d の dK 統計量をターゲットにすることにより，元のグラフ統計量をより正確に捉えた．より大きな d の推定した dK 統計量を反映した生成グラフがより詳細な構造を捉えることは自明ではない．なぜなら，推定した dK 統計量の推定精度は，次元 d が大きくなるにつれて低下するためである．この結果は，推定値の誤差の増加を最小限に抑えて各 dK 統計量を作成および修正するアルゴリズムを設計した成果である.

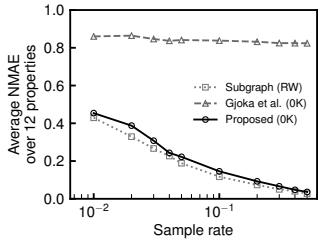
表 2 は，すべてのデータセットにおける 4 つの手法の 12 個の統計量の NMAE の平均と標準偏差を示す．4 つの手法は，BFS による部分グラフ手法，RW による部分グラフ手法，$2.5K$ 統計量をターゲットとする Gjoka らの手法，および $2.5K$ 統計量をターゲットとする提案手法である．サンプル率は 10%とした．$2.5K$ をターゲットとする提案手法は，ほとんどすべてのデータセットで最も低い平均 NMAE と標準偏差を達成した．つまり，$2.5K$ をターゲットとする提案手法は，12 個の統計量を満遍なく最も正確に捉える生成グラフを実現した.

5.3 グラフの可視化

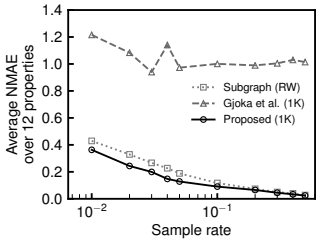
次に，生成グラフの視覚表現の観点から各手法を比較する．図 4 は，Anybeat グラフにおいて，元のグラフと，サンプル率 10%で 4 つの手法により生成された各グラフの可視化結果を示している．各グラフの可視化には，Gephi ソフトウェア [21] を使用した．まず，BFS と RW による部分グラフは，高次数ノードからなる中心構造を捉えることはできるが，低次数ノードからなる周辺構造を捉えることができない (図 4(b)–(c))．これは，BFS および RW が高次数ノードに偏ってサンプリングするためである [3]．$2.5K$ をターゲットとする Gjoka らの手法は，元のグラフの真の構造を視覚的に捉えることができない．既存手法に対して，$2.5K$ をターゲットとする提案手法は，真の構造を視覚的に捉え，高次数ノードの中心構造だけでなく，部分グラフ手法では捉えきれない低次数ノードの周辺構造も捉えた最も正確な視覚表現を実現した.

5.4 生成時間

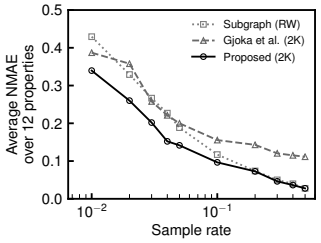
最後に，提案手法の効率性を評価する．図 5 は，Anybeat と Epinions においてサンプル率が 10%の場合の各手法の生成時間を示す．部分グラフ手法は，構築時間が $O(|E'|)$ であるため，非常に高速である．次に，$0K$, $1K$, および $2K$ をターゲットとする場合，提案手法にはターゲットの dK 統計量を修正する処理を行うため，Gjoka らの手法よりも生成時間が長い．ただし，実時間での差はわずかである．最後に，$2.5K$ をターゲットとする提案手法は，Gjoka らの手法より Anybeat において



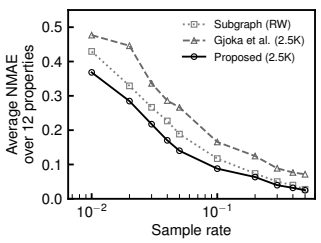
(a) 部分グラフ手法 (RW), Gjoka らの手法 (0K), および提案手法 (0K)



(b) 部分グラフ手法 (RW), Gjoka らの手法 (1K), および提案手法 (1K)

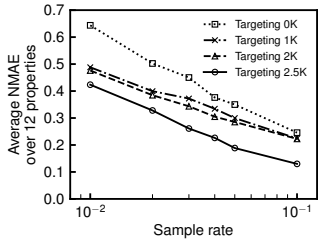


(c) 部分グラフ手法 (RW), Gjoka らの手法 (2K), および提案手法 (2K)

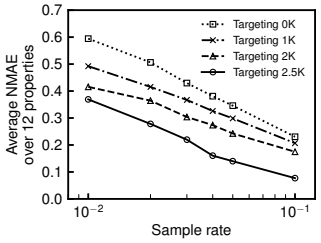


(d) 部分グラフ手法 (RW), Gjoka らの手法 (2.5K), および提案手法 (2.5K)

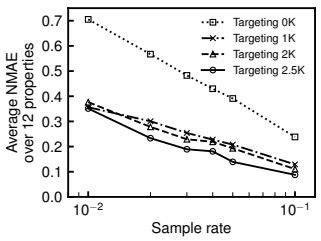
図 2　Anybeat におけるランダムウォークに基づく 3 つの手法の 12 個の統計量にわたる平均 NMAE．サンプル率は 1%から 50%まで変化させた．すべての結果は 10 回の実行の平均値である．



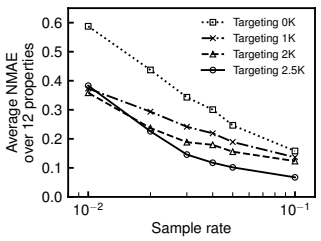
(a) Enron



(b) Brightkite



(c) Douban



(d) Epinions

図 3　4 つのグラフにおける各 dK 統計量をターゲットとする提案手法の 12 個の統計量にわたる平均 NMAE．サンプル率は 1%から 10%まで変化させた．すべての結果は 10 回の実行の平均値である．

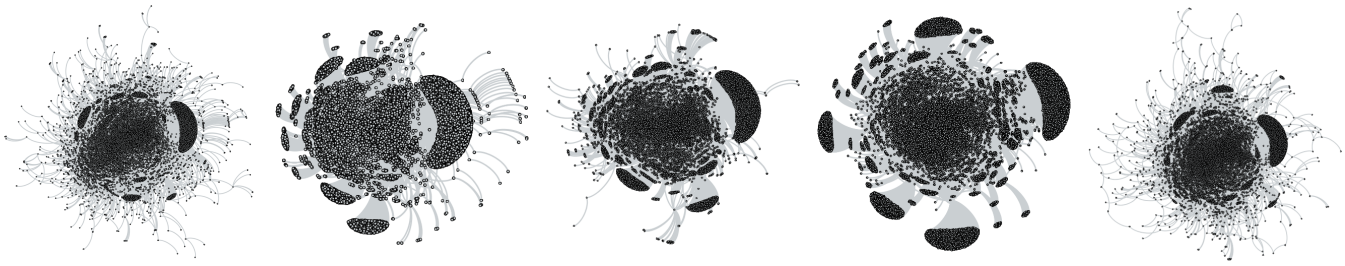
表 2　9 つのグラフにおける 4 つの手法の 12 個の統計量の NMAE の平均と標準偏差．サンプル率は 10%とした．すべての結果は平均 ± 標準偏差として示されている．すべての結果は，10 回の実行の平均値である．

グラフ名	部分グラフ手法 (BFS)	部分グラフ手法 (RW)	Gjoka らの手法 (2.5K)	提案手法 (2.5K)
Anybeat	0.304 ± 0.151	0.117 ± 0.079	0.166 ± 0.140	**0.088 ± 0.065**
Enron	0.258 ± 0.154	0.198 ± 0.128	0.175 ± 0.152	**0.130 ± 0.087**
Brightkite	0.290 ± 0.198	0.195 ± 0.158	0.159 ± 0.191	**0.077 ± 0.062**
Douban	0.129 ± 0.115	0.093 ± **0.077**	0.183 ± 0.254	**0.088** ± 0.087
Epinions	0.198 ± 0.190	0.167 ± 0.162	0.120 ± 0.129	**0.067 ± 0.063**
Slashdot	0.168 ± 0.137	0.128 ± 0.102	0.161 ± 0.197	**0.069 ± 0.069**
Facebook	0.397 ± 0.247	0.255 ± 0.159	0.203 ± 0.197	**0.167 ± 0.123**
Gowalla	0.304 ± 0.218	0.175 ± 0.145	0.224 ± 0.238	**0.098 ± 0.088**
Livemocha	0.231 ± 0.185	0.147 ± **0.117**	0.264 ± 0.327	**0.121** ± 0.155

8.0 倍, Epinions において 9.2 倍高速であった．これは，提案手法が部分グラフの構造を厳密に保持することによって，2.5K グラフの生成時間のボトルネックであるエッジの再配線回数を Gjoka らの手法の $O(|\tilde{E}|)$ から $O(|\tilde{E}|-|E'|)$ に削減するためである．ほぼすべてのデータセットで，Gjoka らの手法よりも提案手法が高速であることを確認した (図 6 を参照)．

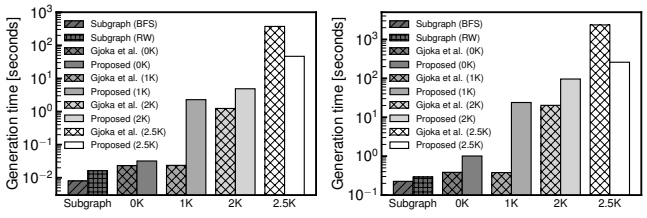
6 おわりに

本研究では，ランダムウォークにより得られたサンプルからソーシャルグラフを復元するグラフ復元手法を提案した．提案手法は，推定した dK 統計量とサンプリングによって得られた真の構造情報の両方を反映したグラフを生成する．実世界のソーシャルグラフデータセットを使用した広範な実験を通じて，提案手法の有効性を検証した．実験結果は，比較されたすべての手法の中で，2.5K 統計量をターゲットとする提案手法が，実用的な生成時間で，局所的および大域的な 12 個の統計量とグラフの視覚的表現を最も正確に捉える生成グラフを実現することを示した．本研究で提案したグラフの復元手法は，データへのアクセスが制限されたソーシャルグラフを正確かつ詳細に分析するための有望な方法となるに違いない．



(a) 元のグラフ　(b) 部分グラフ手法 (BFS)　(c) 部分グラフ手法 (RW)　(d) Gjoka らの手法 (2.5K)　(e) **提案手法 (2.5K)**

図 4　Anybeat におけるグラフの可視化結果．(a) 元のグラフ．(b)–(e) サンプル率 10%で 4 つの手法により生成されたグラフ．



(a) Anybeat　(b) Epinions

図 5　Anybeat と Epinions における各手法の生成時間．サンプル率は 10%とした．すべての結果は，10 回の実行の平均値である．

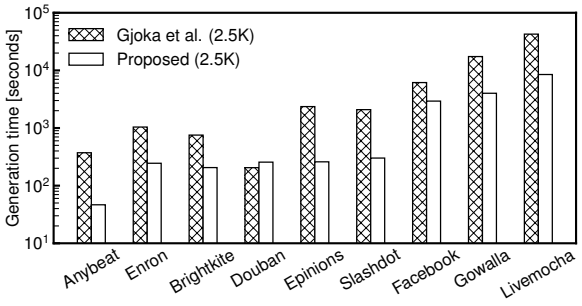


図 6　2.5K 統計量をターゲットとした提案手法と Gjoka らの手法の生成時間．サンプル率は 10%とした．全ての結果は 10 回の実行の平均値である．

謝辞　本研究の一部は，国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務として行われました．

文　献

[1] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proc. IMC*, pp. 29–42, 2007.

[2] Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. Analysis of topological characteristics of huge online social networking services. In *Proc. WWW*, pp. 835–844, 2007.

[3] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. Walking in facebook: A case study of unbiased sampling of osns. In *Proc. INFOCOM*, pp. 1–9, 2010.

[4] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proc. WWW*, pp. 591–600, 2010.

[5] Stephen J Hardiman and Liran Katzir. Estimating clustering coefficients and size of social networks via random walk. In *Proc. WWW*, pp. 539–550, 2013.

[6] Anirban Dasgupta, Ravi Kumar, and Tamas Sarlos. On estimating the average degree. In *Proc. WWW*, pp. 795–806, 2014.

[7] Liran Katzir, Edo Liberty, and Oren Somekh. Estimating sizes of social networks via biased sampling. In *Proc. WWW*, pp. 597–606, 2011.

[8] Minas Gjoka, Maciej Kurant, and Athina Markopoulou. 2.5 k-graphs: from sampling to generation. In *Proc. INFOCOM*, pp. 1968–1976, 2013.

[9] Stephen P Borgatti, Martin G Everett, and Jeffrey C Johnson. *Analyzing social networks*. SAGE, 2018.

[10] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. Systematic topology analysis and generation using degree correlations. *ACM SIGCOMM Computer Communication Review*, Vol. 36, No. 4, pp. 135–146, 2006.

[11] Chiara Orsini, Marija M. Dankulov, PolSimón, Almerima Jamakovic, Priya Mahadevan, Amin Vahdat, Kevin E. Bassler, ZoltánToroczkai, MariánBoguñá, Guido Caldarelli, Santo Fortunato, Dmitri Krioukov. Quantifying randomness in real networks. *Nature Communications*, Vol. 6, p. 8627, 2015.

[12] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, Vol. 2, No. 2, pp. 129–233, 2010.

[13] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, Vol. 9, No. 6, p. 717, 2009.

[14] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, Vol. 58, No. 7, pp. 1019–1031, 2007.

[15] Myunghwan Kim and Jure Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *Proc. SDM*, pp. 47–58, 2011.

[16] Mark Newman. *Networks*. Oxford university press, 2018.

[17] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, Vol. 64, No. 2, p. 026118, 2001.

[18] Isabelle Stanton and Ali Pinar. Constructing and sampling graphs with a prescribed joint degree distribution. *Journal of Experimental Algorithmics*, Vol. 17, pp. 3–1, 2012.

[19] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proc. AAAI*, pp. 4292–4293, 2015.

[20] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[21] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *Proc. ICWSM*, pp. 361–362, 2009.