SNSのフォローグラフにおけるノードの階級への分割

小川　祐人†　　田島　敬史††

† 京都大学工学部情報学科　〒 606-–8501 京都府京都市左京区吉田本町
†† 京都大学情報学研究科　〒 606-–8501 京都府京都市左京区吉田本町
E-mail: †ogawa.yuto@dl.soc.i.kyoto-u.ac.jp, ††tajima@i.kyoto-u.ac.jp

あらまし　本稿では，SNS のフォローグラフ上でユーザが複数の階級に分かれていると仮定し，その階級への分割の手法を提案する．SNS の構造について以下のように仮定する．SNS 上には，様々な人気度のユーザが存在し，二人のユーザが同程度の人気であれば相互リンクを持ちやすく，片方向リンクはより人気のないユーザからより人気のあるユーザに向かうことが多い．また，相互リンクは，人気があるユーザの間であるほど密になりやすい．以上の仮定に基づき，SNS のフォロー関係による構造を階級構造にモデル化し，SNS 上のフォローグラフからユーザたちを階級構造に分けるアルゴリズムを提案する．その後，提案手法と従来のランキング手法などを違反エッジの最小化によって比較評価する．

キーワード　ソーシャルメディア，ソーシャルネットワーク，ネットワーク分析，Web 情報分析

1　はじめに

Social Networking Services (SNS) は，Web 上で人々が社会的なネットワークを形成できるサービスであり，日々SNS 内で様々なやり取りが行われている．近年 SNS は益々普及しており，人々の娯楽やビジネスツールとして使用され重要なものになっている．代表的な SNS として，Twitter や Facebook，Instagram 等があげられる．そういった代表的な SNS では，自分で好きな内容の投稿をしたり，他人の投稿に対して評価を付けたりできる．また，ユーザをフォローしたり，ブロックしたりすることもできる．フォローとは，投稿やメッセージを継続して読みたいユーザや友達などをリストに登録して保持する機能のことである．あるユーザをフォローしているユーザ達をフォロワーといい，フォロワーの数はユーザの人気を図る一つの指標である．ユーザが互いにフォローしあうこともあり，そのような関係は相互フォローと呼ばれる．SNS には，相互フォローを前提とする SNS とそうでない SNS がある．本研究は，Twitter のような必ずしも相互フォローでなくてもよい SNS を想定する．

1.1　非相互フォローを含む SNS のフォロー関係

人気が高いユーザは，様々なユーザからのフォローを受け，フォロワーの人気も様々である．このような場合，人気が高いユーザ同士では，相互フォローがより発生しやすい．一方で，フォロワーが少ない人気が低いユーザ達であっても，そのようなユーザたちが同じトピックに関心があったり，同じコミュニティに所属するユーザであれば互いにフォローしやすい．このような場合，人気が低いユーザ間でも相互フォローが発生しやすい．以上のことから，相互フォローでつながったユーザたちが一つのグループを作り，それが複数存在するということが推測される．また，グループ間には上下関係があり，より人気が高いユーザからなるグループとより人気が低いユーザからなるグループが存在すると思われる．一方で，人気が高いユーザから人気の低いユーザへのフォローは同程度の人気のユーザ間のフォローより起こりにくい．このような観点から，SNS のユーザたちを複数の階級に分けることを考える．実世界においても，著名な人物や人気が高い人物同士が知り合いであったり，一般の人々が著名な人たちを一方的に知っているという構造がある．逆に，人気が高い人物間で一方的な認識であったり，著名な人物が一般の人を知っているようなケースは少ないと推測される．このように，より人気が高い人物たちのグループとそうではない人たちのグループが現実においても存在するといえる．

1.2　研究の目的

本稿では，SNS のフォローグラフ上でユーザが複数の階級に分かれていると仮定し，その階級への分割の手法を提案する．

SNS の構造について以下のように仮定する．SNS 上には，様々な人気度のユーザが存在し，ユーザたちが複数の階級からなる階級構造を作っている．上の階級ほどユーザ数が少なく，下の階級であるほどユーザの数が多い．二人のユーザが同程度の人気であれば，同じ階級に属し，相互リンクを持ちやすい．片方向リンクは，下の階級に所属するより人気のないユーザから，上の階級に所属するより人気のあるユーザに向かうことが多い．また，相互リンクは，上の階級に所属する人気が高いユーザの間であるほど密になりやすい．これは，より人気が高いユーザからなる相互リンクのグラフは，平均の次数が高いということである．

以上の仮定に基づき，SNS のフォロー関係による構造を階級構造にモデル化し，SNS 上のフォローグラフからユーザたちを階級構造に分けるアルゴリズムを提案する．今回は，階級が三つからなるとして，ノードを分類する．その後，上述の仮定に反する違反エッジの数によって性能を評価する．

違反エッジの最小化は，各ノードの分類パターンを総当たり

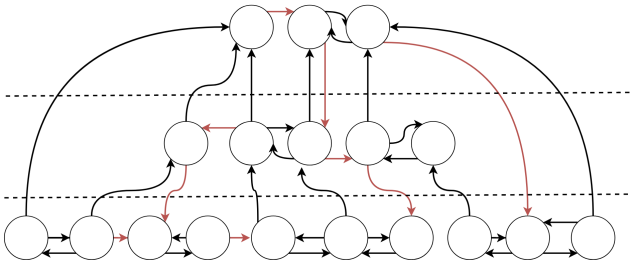

図 2.1　階級に分けたグラフの例
黒色: 違反していないエッジ，赤色: 違反しているエッジ

することで達成できる．しかし，ノード数を N，階級の数を L とすると，L^N 通りのパターンがあり，計算量の観点から実行することは難しい．そこで，本研究では実行可能な計算量で近似解を求める手法を提案する．

本論文の以下の構成は次のようになっている．2 章では，今回扱うグラフの階級化の問題についての定式化を行う．3 章では，問題に関連する研究を簡単に紹介する．4 章では，今回提案する二つの手法について説明する．5 章では，実験とその結果について述べ，6 章で違反エッジとスコアの分布，時間計算量について各手法の考察を行う．

2　問題の定義

本研究で扱う問題は以下のように定式化できる．問題の入力として SNS のフォローグラフが与えられ，このフォローグラフのノードを次のような性質を可能な限り満たすように三つの階級に分類する．

- 同じ階級同士のノード間にエッジが存在する場合は相互エッジを持つ
- 違う階級に分割されるノード間にエッジが存在する場合は，下の階級から上の階級への片方向エッジを持つ

これらの性質に従わないエッジを違反エッジとする．違反エッジは，2 種類あり

- 同じ階級同士のノード間の片方向エッジ
- 違う階級に分割されるノード間の，上の階級から下の階級への片方向エッジ

となる．同じ階級に所属するノード間にエッジが存在しない場合は，違反とならない．

図 2.1 は，分割したグラフの一例である．黒色のエッジは前述の性質に従っているエッジである．赤色のエッジは前述の性質に違反しているエッジである．

これらの違反エッジの合計数を最小化する分割アルゴリズムを提案する．

3　関 連 研 究

SNS 上におけるユーザたちの行動は，ユーザや投稿の間に関係を作る．これらは，ユーザや投稿をノードとし，フォローや評価をエッジとしたグラフ構造を持つデータと考えられる．このグラフデータを解析することは，SNS の構造や特性を明らかにし，マーケティングやレコメンドなどの実用的な分野に応用するために有用である．過去の研究では，これらのグラフデータを用いて，人気ランキングやクラスタリングなどのアルゴリズムなどが開発され，SNS のグラフデータの解析が進められてきた．

3.1　minimum feedback arc set

関連研究として minimum feedback arc set (mFAS) [1] が挙げられる．mFAS は，エッジの集合で，グラフから取り除いたときにグラフがサイクルを持たないようになる要素数最小のエッジ集合を指す．mFAS を求めて除去することでグラフにトポロジカルソートを適用できるようになり，下から上へエッジが伸びるようになるという点で今回の問題設定に関連している．また，mFAS を求めるための効率の良い近似アルゴリズム [2] が過去に開発されている．相違点として，本研究ではノードを階級に分類し，同じ階級での相互エッジを違反としないという点がある．

3.2　クラスタリングアルゴリズム

グラフ構造の分割は，ソーシャルネットワーク分析における基本的な問題であり，様々な分野に対して応用可能である．グラフ構造の分割の目的は，あらかじめ定義した目的関数を最小化・最大化するようにグラフを複数のグループに分けることである．目的関数の例としては，グループ間のカットの最小化 [3] や，グループ分けのモジュラリティの最大化などがある．SNS では，ユーザをノードとするフォローグラフ等に適用することで，ユーザをトピックやコミュニティごとに分ける [4] ことなどができる．Girvan-Newman のアルゴリズム [5] は，グラフのエッジに対して媒介中心性を計算し，それが高いエッジから削除していくことでノードを分離していきクラスタリングを行う．Louvain 法 [6] では，モジュラリティが最大となるようにノードが所属するクラスタを繰り返し変更していき高いモジュラリティを達成するクラスタリングを行う．SCAN [7] は，core と呼ばれるノードから構造類似性を用いてコミュニティを広げていくことでクラスタリングを行う手法である．コミュニティに所属しないノードが存在し，それらはコミュニティ間をつなぐハブや孤立点として分類される．

3.3　ランキングアルゴリズム

前述の三つの階層への分類問題に対しては，従来のランキングアルゴリズムでランキングしたノードを二つの閾値によって分割するというアプローチも考えられる．より多くのノードから片方向リンクを受けているノードが上の階級に行きやすいという点で，ノードの人気度や影響力を考慮するようなアルゴリズムがより適している．

PageRank (PR) [8] は，情報検索や実際の検索エンジンにも利用された有名なランキング手法である．初めは，Web ページ検索に使われたが，一般的なグラフの指標や他のグラフアルゴリズムとの比較などの場面でも多く利用される．PR では，より人気の高いノードからリンクを受けるようなノードもまた人気が高いとして，スコアを計算する．PR は，行列の形で

計算するが，モデルとしては，グラフ上でのランダムウォークなどとして解釈できる．また，Topic-Specfic PageRank [9] や Topic-Sensitive PageRank [9] などの様々な応用が存在する．

PR に類似するランキングアルゴリズムに HITS [10] がある．HITS では，ページ自体の人気度に当たる authority とどの程度人気のページにリンクを張っているかに当たる hub を計算する．両社は相互再帰的に計算され，高い authority のページを指すページは高い hub を持ち，高い hub を持つページからリンクを受けるページは高い authority を持つ．高い authority ノードがあったとしても，高い hub をもつノードが一つだけリンクを張っている場合などがあるので，違反エッジを少なくするという観点からは今回の問題のモデルには PR のほうが適していると思われる．

中心性は，グラフ分析の代表的な指標であり，影響力を表すような中心性として Katz centrality [11] などはランキングにも利用されている．Katz centrality は SNS 等のグラフにおいて，ノードの相対的な影響力を示す指標として使われ，隣接したノード以外のノードも間接的に考慮して計算される．PR なども一種の中心性とされ，機械学習などの手法にも組み込まれることがある．

Elo rating [12] は，チェスのプレイヤーをランキングするために開発された手法であり，現在も様々な競技のランキングに取り入れられている．Elo rating は，式 (3.1) のように計算される．$i \to j$ となるエッジについて，

$$\begin{aligned} h_i &= h_i - K\mu_{ij} \\ h_j &= h_j + K\mu_{ij} \end{aligned} \tag{3.1}$$

$$\mu_{ij} = \frac{1}{1 + 10^{\frac{-(h_i - h_j)}{\xi}}}$$

という式によってスコアを更新するということを全エッジについて繰り返すことでスコアを求める．パラメータ K は，スコアの差を重み付けするパラメータであり，プレイヤーや大会などの種類によって調整される．例えば，試合数の少ないプレイヤーは，K を大きくすることでスコアを変化させやすくする．パラメータ ξ は，スコアの分散に影響するパラメータであり，ξ が大きいほど分散が大きくなる．プレイヤー i のスコアがプレイヤー j より ξ だけ高い場合，プレイヤー i はプレイヤー j より勝つ確率が 10 倍大きいということを意味する．Elo rating では，試合の勝敗に当たる負けた側から勝った側へのエッジを順に処理していき，エッジを出しているノードのスコアを低く，エッジを受けているノードのスコアを高くするようにスコアを更新していく．この時の変化量は，その勝敗がどの程度予期できたかによって決められる．例えば，スコアが低いノードがスコアの高いノードに勝つと，大きな変化量となる．また，本来チェスなどの競技でのランキングに使われるので，試合が行われた順番でエッジが参照されていく．

4 提 案 手 法

本章では，SNS のフォローグラフにおけるノードを三つの階級に分けるアルゴリズムを二つ提案する．以下では，三つの階級を下から階級 0，階級 1，階級 2 とする

4.1 アルゴリズム案 1: ノードスコアの差を考慮した連続値のレイティング

アルゴリズム案 1（以下，案 1 とする）では，各ノードに対して一つの連続値をとるスコアが与えられる．スコアは，各エッジについて，片方向エッジならば式 (4.1)，相互エッジならば式 (4.2) によって更新される．この更新をスコアが収束するまで行う．スコアの初期値は，ノード数を N としたとき，各ノードに対して $\frac{1}{N}$ とする．

$i \to j$ となる，片方向エッジの場合

$$\begin{aligned} h_i^{(t+1)} &= h_i^{(t)} - K_1\mu_{ij}^{(t)}C^{-t} \\ h_j^{(t+1)} &= h_j^{(t)} + K_1\mu_{ij}^{(t)}C^{-t} \end{aligned} \tag{4.1}$$

$$\mu_{ij}^{(t)} = \frac{1}{1 + 10^{\frac{-(h_i^{(t)} - h_j^{(t)})}{\xi}}}$$

相互エッジの場合

$h_i > h_j$ ならば

$$\begin{cases} h_i^{(t+1)} = h_i^{(t)} - K_2\nu C^{-t} \\ h_j^{(t+1)} = h_j^{(t)} + K_2\nu C^{-t} \end{cases} \tag{4.2}$$

$h_i < h_j$ ならば

$$\begin{cases} h_i^{(t+1)} = h_i^{(t)} + K_2\nu C^{-t} \\ h_j^{(t+1)} = h_j^{(t)} - K_2\nu C^{-t} \end{cases} \tag{4.3}$$

$h_i = h_j$ ならば更新しない

$$\nu = \frac{1}{1 + 10^{\frac{-|h_i^{(t)} - h_j^{(t)}|}{\xi}}}$$

$h_i^{(t)}$ は，t ステップ目におけるノード i のスコア，ξ は Elo rating でのパラメータである．C は収束する早さを決定するパラメータである．パラメータ K_1，K_2 は，Elo rating でのパラメータ K に当たるパラメータであり，各式における変化量の大きさを調節するパラメータである．K_1，K_2 を調節することで片方向エッジと相互エッジの変化量のバランスを調節できる．例えば，K_1 を K_2 よりも大きく設定することで，相互エッジよりも片方向エッジの影響を大きくすることができる．

エッジが片方向エッジの場合は，Elo rating で適用される更新式を採用している．片方向エッジでのスコアの更新では，$\mu_{ij}^{(t)}$ の式に従って，エッジを出している側 (ソース側) のスコアを低く，エッジを受けている側 (ターゲット側) のスコアを高く更新する．これは，片方向エッジは下の階級から上の階級に向かうことを意識している．この時のスコアの変化量は，ソース側がターゲット側より高いスコアであるほど大きくなる．Elo rating と異なる点は，収束のための項 C を追加している点である．

エッジが相互エッジの場合は，二つのノードのスコアが互いに近づくように更新される．ν は，スコアの差を変化量に反映

Algorithm 1 案 1 を用いてスコアを計算するアルゴリズム

```
max_step ← 繰り返す最大回数，n ← ノード数
scores ← サイズ n の配列
scores の各要素 ← 1/n
E ← edges
t ← 0，current ← scores
while 収束条件を満たしていない and t < max_step do
    for edge ← E do
        if edge が片方向エッジならば then
            current を現ステップのスコア，
            scores を前ステップのスコアとして式 (4.1) で更新
        else if edge が相互エッジならば then
            current を現ステップのスコア，
            scores を前ステップのスコアとして式 (4.2) で更新
        end if
    end for
    t ← t + 1
    scores ← current
end while
return scores
```

させる項であり，二つのノードのスコアの差が大きいほど大きな値をとる．スコアが近づくように更新するため，スコアの高いほうから変化量分だけスコアを減少させ，スコアの低いほうから変化量分だけスコアを増加させる．ノードのスコアが同じ場合はスコアは変化しない．また，片方向エッジと相互エッジのどちらの場合においても，収束項 C^{-t} を付け加えている．

スコアを計算した後にグリッドサーチを用いてノードを三つの階級に分けるために閾値を二つ決定する．ノードの属する階級は二つの閾値との大小関係で決定される．より高いスコアを持つノードが上の階級に分類される．例えば，スコアを x，閾値が a，$b\,(a<b)$ としたときに，$x \leqq a$ ならば階級 0，$a < x \leqq b$ ならば階級 1，$b < x$ ならば階級 2 に分類される．閾値は，グリッドサーチを用いて違反エッジが最小になるように決定する．

Algorithm 1 に，式 (4.1) と式 (4.2) を用いて各ノードの計算を行う疑似コードを示す．

4.2 アルゴリズム案 2: ノードの種類を考慮した階級への所属確率

アルゴリズム案 2（以下，案 2 とする）では，各ノードに対して連続値をとる三つのスコアが与えられる．三つのスコアは，ノードが三つの階級のうちどれに属しているかの確率を表している．収束するまで計算を繰り返し，最終的なスコアを分類に用いる．スコアの初期値は，階級の数を L とすると，各ノードに対して，要素が $\frac{1}{L}$，要素数が L のベクトルとする．案 2 では，次のような考え方でスコアを更新する．

片方向エッジの場合，ソース側については，L 番目の階級にいる確率 p_L は以下の式 (4.4) に基づいて更新される．

ノード i からノード j に片方向エッジがあるとき，

$$p_{i,t+1}^{(L)} = p_{i,t}^{(L)} \times (p_{j,t}^{(l>L)} + \alpha p_{j,t}^{(L)}) \tag{4.4}$$

$$p_{j,t+1}^{(L)} = p_{j,t}^{(L)} \times (p_{i,t}^{(L>l)} + \alpha p_{i,t}^{(L)}) \tag{4.5}$$

ここで，$p_{i,t}^{(l)}$ は t ステップ目のノード i が階級 l である確率，$p_{i,t}^{(l>L)}$ は t ステップ目のノード i が L 以上の階級である確率である．式 (4.4) の $p_{j,t}^{(l>L)}$ は，ターゲット側がソース側より上の階級にいる確率を表している．式 (4.4) の $p_{j,t}^{(L)}$ は，ターゲット側がソース側と同じ階級にいる確率を示している．この項については，パラメータ α を変化させることで同じ階級での片方向リンクをどの程度許容するかを調節する．すなわち，ソース側がある階級である確率は，ソース側が元々その階級に属する確率にターゲット側がそれより上の階級に属する確率と同じ階級にいる確率をパラメータ α で重みづけした値を掛け合わせたものである．

ターゲット側については，ソース側と対称になっている式 (4.5) を用いて更新する．すなわち，ターゲット側がある階級である確率は，ターゲット側が元々その階級に属する確率にソース側がそれ以下の階級に属する確率を掛け合わせたものである．

相互エッジの場合は，同じ階級にいることが望ましいので，式 (4.6) のように，二つのノードの同じ階級同士のスコアを掛け合わせる．すなわち，相互エッジを持つノードの片方が，ある階級である確率は，そのノードが元々その階級である確率と，相互エッジをもつもう一方のノードが同じ階級である確率を掛け合わせたものである．

$$p_{i,t+1}^{(L)} = p_{i,t}^{(L)} \times p_{j,t}^{(L)} \tag{4.6}$$

式 (4.4) と式 (4.5) を三つの層の場合に適用すると更新式は以下の式 (4.7) と式 (4.8) となる．実験では，スコアは，各エッジについて式 (4.7)， 式 (4.8) によって更新される．この更新をスコアが収束するまで行う．また，各ステップごとに各ノードの三つのスコアが合計 1 になるように正規化を行う．

$i \to j$ となる，片方向エッジの場合

$$\mathbf{p}_{i,t+1} = \begin{cases} p_{i,t}^{(2)} \times \alpha p_{j,t}^{(2)} \\ p_{i,t}^{(1)} \times (p_{j,t}^{(2)} + \beta p_{j,t}^{(1)}) \\ p_{i,t}^{(0)} \times ((p_{j,t}^{(1)} + p_{j,t}^{(2)}) + \gamma p_{j,t}^{(0)}) \end{cases}$$
$$\mathbf{p}_{j,t+1} = \begin{cases} p_{j,t}^{(2)} \times ((p_{i,t}^{(0)} + p_{i,t}^{(1)}) + \alpha p_{i,t}^{(2)}) \\ p_{j,t}^{(1)} \times (p_{i,t}^{(0)} + \beta p_{i,t}^{(1)}) \\ p_{j,t}^{(0)} \times \gamma p_{i,t}^{(0)} \end{cases} \tag{4.7}$$

相互エッジの場合

$$\mathbf{p}_{i,t+1} = \begin{cases} p_{i,t}^{(2)} \times p_{j,t}^{(2)} \\ p_{i,t}^{(1)} \times p_{j,t}^{(1)} \\ p_{i,t}^{(0)} \times p_{j,t}^{(0)} \end{cases} \quad \mathbf{p}_{j,t+1} = \begin{cases} p_{j,t}^{(2)} \times p_{i,t}^{(2)} \\ p_{j,t}^{(1)} \times p_{i,t}^{(1)} \\ p_{j,t}^{(0)} \times p_{i,t}^{(0)} \end{cases} \tag{4.8}$$

$p_{i,t}^{(l)}$ を t ステップ目のノード i が階級 l である確率，$\mathbf{p}_{i,t}$ は t ステップ目のノード i が各階級である確率をベクトルにまとめたものである．α，β，γ は各層における同じ層への片方向エッジの許容度に当たるパラメータである．

各階級への分類は，各ノードにおける確率で重みづけした期待値に最も近い階級に分類を行う．期待値は，各ノードについ

Algorithm 2 案 2 を用いてスコアを計算するアルゴリズム

```
max_step ← 繰り返す最大回数，n ← ノード数
scores ← サイズ n の配列
scores の各要素 ← [1/3, 1/3, 1/3]
E ← edges
t ← 0，current ← scores
while 収束条件を満たしていない and t < max_step do
    for edge ← E do
        if edge が片方向エッジならば then
            current を現ステップのスコア，
            scores を前ステップのスコアとして式 (4.7) で更新
        else if edge が相互エッジならば then
            current を現ステップのスコア，
            scores を前ステップのスコアとして式 (4.8) で更新
        end if
    end for
    for score ← scores do
        score が合計 1 になるように正規化
    end for
    t ← t + 1
    scores ← current
end while
return scores
```

て，階級 x に対応するスコアを $p^{(x)}$ とすると，式 (4.9) のように計算される．

$$score_i = 0 * p_i^{(0)} + 1 * p_i^{(1)} + 2 * p_i^{(2)} \tag{4.9}$$

このスコアに最も近い階級に分類される．

Algorithm 2 に，式 (4.7) と式 (4.8) を用いて各ノードのスコアの計算を行う疑似コードを示す．

5 実　　験

5.1 使用するデータ

実験では，2021 年 12 月から 2022 年 1 月の間にクロールされた 27,500,000 ノードの Twitter 上の日本語ユーザのフォローグラフの部分グラフを使用する．使用する部分グラフは以下の四つである．データサイズが大きすぎるグラフはアルゴリズムの実行に非常に長い時間がかかるため，クロールの起点となった各ノードは，データサイズが大きくなりすぎないように選定した．

- データ A: あるノードから 2 hop のフォロー先に含まれるノード集合から誘導されるグラフ (ノード数:760,866，エッジ数:91,742,833，起点ノードのフォロワー数:57271，起点ノードのフォロー数:217)
- データ B: あるノードから 2 hop のフォロワーに含まれるノード集合から誘導されるグラフ (ノード数:1,304,502，エッジ数:256,221,274，起点ノードのフォロワー数:457，起点ノードのフォロー数:690)
- データ C: あるノードから 1 hop のフォロワーに含まれるノード集合とそれらの 1hop のフォロー先から誘導されるグラフ (ノード数:276,377，エッジ数:81,658,460，起点ノードのフォロワー数:281，起点ノードのフォロー数:0)
- データ D: あるノードから 1 hop のフォロー先に含まれるノード集合とそれらの 1hop のフォロワーから誘導されるグラフ (ノード数:452,082，エッジ数:54,776,233，起点ノードのフォロワー数:24，起点ノードのフォロー数:69)

これらのグラフについて各アルゴリズムを適用し実験を行う．

表 5.1　データ A における各手法における違反エッジ数

違反種	PageRank	Elo	follower	案 1	案 2-1	案 2-2
2→2	9,714,564	2,469,070	1,114,570	3,771,359	13,673,601	17,319,847
2→1	9,846,668	2,071,909	5,799,904	2,097,951	559,832	780,730
2→0	1,814,089	6,805,509	4,584,701	7,399,161	7,748,351	6,751,182
1→1	2,934,401	1,300,234	4,221,270	73,701	7,136	10,644
1→0	630,084	7,409,378	4,405,051	736,704	185,933	162,884
0→0	112,621	3,037,442	2,149,819	2,955,108	713,338	430,078
合計	25,052,427	23,093,542	22,275,405	**17,033,984**	22,888,191	25,455,365

表 5.2　データ B における各手法における違反エッジ数

違反種	PageRank	Elo	follower	案 1	案 2-1	案 2-2
2→2	12,376,266	1,118,271	38,087,012	3,388,870	20,309,733	25,369,127
2→1	40,365,371	6,097,036	16,413,865	674,200	2,412,002	8,504,452
2→0	11,789,483	5,352,396	9,267,382	47,365,906	38,274,295	25,705,563
1→1	2,883,465	2,858,531	263,782	6,273	20,943	183,584
1→0	2,811,735	37,289,718	669,313	603,235	580,359	1,187,587
0→0	326,751	10,585,835	423,830	6,983,825	2,245,094	860,818
合計	70,553,071	63,301,787	65,125,184	**59,022,309**	63,842,426	61,811,131

表 5.3　データ C における各手法における違反エッジ数

違反種	PageRank	Elo	follower	案 1	案 2-1	案 2-2
2→2	6,838,132	5,093,820	7,517,456	2,598,026	21,778,324	24,582,110
2→1	5,928,988	5,029,518	6,043,516	3,122,790	504,605	793,002
2→0	3,998,620	4,557,806	3,852,068	4,468,724	4,018,904	3,232,986
1→1	3,318,816	3,601,481	3,588,173	3,117,716	7,980	17,983
1→0	3,985,655	3,514,847	3,060,376	5,748,244	137,650	146,015
0→0	3,000,564	5,146,099	3,124,377	6,027,299	254,096	126,158
合計	27,070775	26,943,571	27,185,966	**25,082,799**	26,701,559	28,898,254

表 5.4　データ D における各手法における違反エッジ数

違反種	PageRank	Elo	follower	案 1	案 2-1	案 2-2
2→2	157,177	122,320	377,226	135,263	4,199,487	4,073,992
2→1	508,920	370,859	643,983	448,644	185,906	287,364
2→0	255,748	383,801	438,263	556,075	1,557,873	1,241,260
1→1	2,970,491	785,715	2,826,253	298,331	15,486	26,632
1→0	1,804,999	2,158,664	1,725,956	2,046,433	63,906	78,614
0→0	474,509	2,103,053	360,006	2,321,697	394,166	221,114
合計	6,171,844	5,924,412	6,371,687	**5,806,443**	6,416,824	5,928,976

5.2 ベースライン手法

ベースラインとして PageRank と Elo rating，フォロワー数によるランキングを用いる手法を採用する．

PageRank を使う手法では，PageRank をデータに適用する．各階級への分類は，二つの閾値を設定し，その大小関係によってノードを分類する．閾値の決め方については，提案手法 1 と同様にグリッドサーチを用いる．

Elo rating を使う手法では，Elo rating がエッジを処理する順序によってスコアが変わることから，複数回ランダムなエッジの順序で Elo rating を算出し，それらの各ノードのスコアの平均を最終的なスコアとする．その後，グリッドサーチを用い

表 5.5　データ A における各階級のノード数

階級	PageRank	Elo	follower	案 1	案 2-1	案 2-2
2	60,866	60,866	10,856	60,866	373,915	405,223
1	210,000	350,000	75,386	280,000	22,870	31,505
0	490,000	350,000	674,624	420,000	384,081	324,138

表 5.6　データ B における各階級のノード数

階級	PageRank	Elo	follower	案 1	案 2-1	案 2-2
2	134,502	4,502	393,794	4,502	250,953	351,095
1	390,000	130,000	663,396	130,000	81,712	241,480
0	780,000	1,170,000	247,312	1,170,000	971,837	711,927

表 5.7　データ C における各階級のノード数

階級	PageRank	Elo	follower	案 1	案 2-1	案 2-2
2	33,377	33,377	33,362	60,377	192,478	200,133
1	54,000	135,000	54,081	108,000	7,924	12,172
0	189,000	108,000	188,934	108,000	75,975	64,072

表 5.8　データ D における各階級のノード数

階級	PageRank	Elo	follower	案 1	案 2-1	案 2-2
2	47,082	47,082	2,081	2,082	43,935	69,881
1	90,000	90,000	45,657	45,000	48,723	64,143
0	315,000	315,000	404,344	405,000	359,424	318,058

て，二つの閾値を設定して階級に分割する．

フォロワー数を使う手法では，フォロワー数順にノードをランキングし，グリッドサーチを用いて二つの閾値を設定して階級に分割する．

5.3　パラメータについて

ベースラインの Elo rating を使った手法では，$K = 1$，ξ=100 とした場合を実験する．

案 1 のアルゴリズムでは $K_1 = K_2 = 1$，ξ=100，$C = 10$ とした場合を実験する．

案 2 のアルゴリズムでは，$\alpha = 0.05$，$\beta = 0.1$，$\gamma = 0.2$（パターン 1）の場合と $\alpha = 0.2$，$\beta = 0.1$，$\gamma = 0.05$（パターン 2）の場合を実験する．

5.4　評 価 方 法

仮定するモデルの性質は以下のようになっている．

- 同じ階級同士のノード間にエッジが存在する場合は相互エッジを持つ
- 違う階級に分割されるノード間にエッジが存在する場合は，下の階級から上の階級への片方向エッジを持つ

これに従わないエッジを違反エッジとする．違反エッジは，2 種類あり

- 同じ階級同士のノード間の片方向エッジ
- 違う階級に分割されるノード間の，上の階級から下の階級への片方向エッジ

となる．違う階級に属するノード間の相互エッジについては上の階級から下の階級へのエッジのみが違反エッジとして数えられる．同じ階級に所属するノード間にエッジが存在しない場合は，違反とならない．この違反エッジの個数を数えることでどの程度モデルに沿っているのかを評価する．

5.5　結　　果

表 5.1，表 5.2，表 5.3，表 5.4 に各データにおける手法ごとの違反エッジのパターンごとの数と総数を示す．各表の一列目の項目について，$x \to x$ は階級 x 内での違反エッジを示す．$x \to y$ は階級 x から階級 y への違反エッジを表している．

表 5.5，表 5.6，表 5.7，表 6.2 に各データにおける各階級のノード数を示す．ノードが区切りの良い数字で分けられている手法があるのは，閾値を決める際に決まったノード数分だけ候補値をずらしながらグリッドサーチをしているためである．グリッドサーチで区切りの悪い数字になっている部分については，閾値と同じスコアのノードが複数あるときにそれらをまとめて閾値以下として扱うので，その数だけ多めに下の階層に分類されるためである．

6　考　　察

6.1　違反エッジ数について

ベースラインに対して案 1 は，どのデータにおいても違反エッジが少なくなった．

違反エッジのパターンに一定の傾向はみられないが，ベースラインに比べ，各階級において違反エッジが多いパターンと少ないパターンがある．違う階級間と同じ階級内の違反エッジを比べると，同じ階級間での違反エッジは比較的少ないように思われる．どのデータに対しても，同じ階級内での違反エッジよりも，違う階級間での違反エッジのほうが大きな割合を占めている．これは，同じ階級間よりも違う階級間のエッジのほうがノードの組み合わせが多くなりやすいことが考えられる．この考え方は，案 1 以外の手法についても同様であると思われる．そのため，同じ階級内の違反エッジを少なくするよりも違う階級間での違反エッジを少なくすることを優先することでより違反エッジを少なくしやすいと思われる．また，違反エッジが増えたパターンの理由として，片方向エッジに比べ相互エッジの影響が多き過ぎたことが考えられる．仮に相互エッジで二つノードを近づける変化量が大きい場合，それらのノードは同じ階級に入りやすいため，同じ階級内には相互エッジが多くなる．一方，片方向エッジの影響が比較的弱くなるので，本来違う階級に入るべきノードがうまく引き離されなくなってしまう．これによって，同じ階級の違反エッジよりも，違う階級間での違反エッジが多くなってしまったと思われる．案 1 は，パラメータ K_1 と K_2 があり，それぞれを変化させることで相互エッジと片方向エッジの変化量のバランスを変化させることができる．データの特性に合わせてパラメータを適切に設定すれば，前述のような影響のバランスを適切にできると思われる．

ベースラインに対して案 2 は，適切なパラメータを設定することで違反エッジを同程度またはそれ以下に減少させることができている．しかし，案 1 よりも減少している数は少なく，データによっては，ベースラインとあまり変わらない結果になっている．データベースラインの手法に比べて，データ A に

表 6.1　最も違反エッジがベースラインに対する違反エッジの変化量の割合

階級	案 1	案 2 パターン 1	案 2 パターン 2
データ A	−0.2623	−0.0089	+0.1022
データ B	−0.0677	+0.0085	−0.0236
データ C	−0.0691	−0.0090	+0.0725
データ D	−0.0200	+0.0831	+0.0007

ついては，パターン 1 の違反エッジが少なく，データ B については，パターン 2 の違反エッジが少なくなっている．データ C については，パターン 1 の違反エッジが少なく，データ D については，ベースラインと同程度の違反エッジになっている．

違反エッジのパターンに一定の傾向はみられないが，ベースラインに比べ，違う階級間での違反エッジが少ないように思われる．データ A は，フォロー先を 2 hop とったデータであるため，フォローを多く受けているノードは，トピックが類似したノードが多く，人気のノード間には相互エッジが多いと思われる．そのため，2 階級での片方向エッジの多さを加味しているパラメータ α が小さいパターン 1 が適していたと思われる．データ B についても，フォロワーを 2 hop とったデータであり，同様にパターン 2 が適していたため，違反エッジが減少したと思われる．データ C についてもフォロワーをとってからフォロー先をとっているためフォローを受けるようなノードが多く，パターン 1 がより違反エッジが少なくなったと思われる．データ D については，フォロー先をとった後にフォロワーをとっているため，他のノードをフォローするようなノードが多く，パターン 2 のほうが違反エッジが少なくなったと思われる．また，他の手法と比べて階級 1 の数が少なく，この階級 1 の減少が，違反エッジを減少できていない原因の一つだと思われる．このスコアの分布については後述する．

表 6.1 にベースラインの中で最も違反エッジが小さい手法に対する違反エッジの変化量の割合を示す．他のデータに比べて，データ A では，案 1 が最も大きく違反エッジを減らしている．データ A は，2hop 分のフォロー先とっているため三つの階級に分けやすいのではないかと推測される．データ B については，案 2 の中では最も違反エッジを減らしている．データ C，D はデータ A，B よりも違反エッジを減少が比較的小さくなっている．

6.2　計算量について

案 1 の計算量について考察する．案 1 のアルゴリズムは 1 ステップ毎にすべてのエッジを一回ずつ処理する．このエッジの数を E とおく．また，ステップの数はスコアが収束するか，予め決めておいた繰り返しの最大数によって決まる．この繰り返しのステップを T とする．この時，時間計算量は $O(ET)$ となる．閾値を決定する部分については，閾値の候補を M とすると $\frac{1}{2}M(M-1)$ 個の閾値のペアにおいて，違反エッジを評価する．違反エッジの数え上げにはエッジを走査していく必要がある．よって閾値の決定の時間計算量は，$O(EM^2)$ である．案 1 のアルゴリズム全体としては，$O(E(T+M^2))$ となる．ノードの分類パターンを全探索する場合のアルゴリズムは，ノード数 N に対して，$O(3^N E)$ となり，ノード数に対してべき乗の早さで計算に時間がかかる．また，PageRank については，行列の積を繰り返す計算法では，繰り返しの回数を T とすると，$O(N^3T)$ の時間計算量になるので，閾値の決定を含めると全体で $O(N^3T+EM^2)$ となる．Elo rating では，エッジを一回ずつ処理するので閾値の決定を含めると $O(EM^2)$ となる．フォロワー数は，エッジを一回ずつ処理することで数えられるので，Elo rating と同様に $O(EM^2)$ となる．これらと比較すると案 1 はエッジ数に対して線形に計算時間が増えていく．一般的に SNS などのソーシャルグラフではノード数よりもエッジ数が大きい傾向にあるが，全探索と比較すると，グラフのサイズが大きい場合には案 1 のアルゴリズムが短時間で計算できる．PageRank と比較すると，エッジ数とノードの数，繰り返しのステップ数のバランスによって，どちらが早く実行できるか決まると思われる．Elo rating と比較すると，案 1 はスコアの計算に時間がより多くかかってしまうため，実行時間が遅いと思われる．フォロワー数の手法と比較した場合も同様である．

表 6.2　各手法の計算量

手法	PageRank	Elo	follower	案 1	案 2
スコア	N^3T	E	E	ET	ET
分類	EM^2	EM^2	EM^2	EM^2	N
全体	N^3T+EM^2	EM^2	EM^2	$ET+EM^2$	$ET+N$

次に案 2 の計算量について説明する．案 2 のアルゴリズムは 1 ステップ毎にすべてのエッジを一回ずつ処理する．このエッジの数を E とおく．また，ステップの数はスコアが収束するか，予め決めておいた繰り返しの最大数によって決まる．この繰り返しのステップを T とする．この時，時間計算量は $O(ET)$ となる．案 1 と違い閾値を決定する必要はない．各ノードの階級を決めるためにノードを一回ずつ処理する必要があるので，それに $O(N)$ かかる．案 2 のアルゴリズム全体としては，$O(ET+N)$ となる．前述のようにグラフの分類を全パターン網羅する場合は $O(3^N E)$ の時間計算量になるので，案 2 は大きなグラフにおいてより短い時間で実行できると思われる．PageRank と比較すると，閾値の候補数や繰り返しのステップ数によるが，PageRank の N^3 の項が大きくなりやすいため，案 2 がより早く実行されると思われる．Elo rating と比較すると，閾値の候補数や繰り返しのステップ数によってどちらが早いか変わってくると思われる．

案 1 と案 2 を比較する．スコアを計算する部分においては，案 1 と案 2 に時間計算量の観点から大きな差はみられない．一方，各ノードの階級を決めるために案 1 では $O(EM^2)$，案 2 では $O(N)$ の時間計算量となる．前述のようにソーシャルグラフにおいてノード数よりもエッジ数が多くなりやすい点と案 1 が閾値の候補の 2 乗に比例する点から，案 1 よりも案 2 のほうが短い時間で階級を決めることが可能だといえる．よって，アルゴリズム全体としては，案 1 よりも案 2 のほうがより短い時間で実行できると思われる．

7 結　　論

7.1 ま と め

本稿では，SNS のフォローグラフにおけるノードを三つの階級に分割するアルゴリズムを二つ提案した．Twitter のフォローグラフのデータを用いて実験を行い，PageRank，Elo rating，フォロワー数を用いたベースラインと比較を行った．実験データに対して，案 1 は，ベースラインより違反エッジを同等またはそれ以下に抑えることができた．案 2 においては，適切なパラメータを設定することによりベースラインより三つのデータで違反エッジを同等またはそれ以下に減少したことを確認したが，データによってはベースラインよりも違反エッジが増加した．また，案 1 のスコアの分布はべき乗分布に近く，案 2 のスコアの分布は階級 1 付近のスコアが少なくなることを確認した．時間計算量の観点からは，案 1 は全探索よりは高速だがベースラインと比較した場合は同程度か遅いと思われる．案 2 については，閾値を設定する必要がないため全探索やベースラインの手法に比べて高速に実行できる．

7.2 今後の課題

今後の課題としては，アルゴリズムの改良があげられる．案 1 と案 2 の共通の課題として，パラメータの設定方法がある．今回はパラメータをあらかじめ決めていたが，これらを設定する最適な方法は見つかっておらず，計算量の観点からグリッドサーチを行うことも難しい．そのため，違反エッジをより減少させるパラメータをより効率的に見つける方法が重要である．案 2 については，各エッジのパターンごとに更新する式をパラメータで調整することなどがあげられる．例えば，今回は同じ階級内の片方向エッジの許容度をパラメータ化していたが，上の階級から下の階級へのエッジの許容度をパラメータ化することも挙げられる．

問題設定については，今回は階級を三つであると仮定して分割を行ったが，それ以外の数ではどのようになるか検証することも挙げられる．案 2 のスコアの偏りなども階級の数が原因である可能性がある．また，違反エッジを理想的にはどの程度まで減らすことが可能なのか見積もる方法も検討したい．

謝　　辞

本研究は JSPS 科研費 21H03446 の助成を受けたものです．

文　　献

[1] Younger, D.: Minimum feedback arc sets for a directed graph, *IEEE Transactions on Circuit Theory*, Vol. 10, No. 2, pp. 238–245 (1963).
[2] Eades, P., Lin, X. and Smyth, W. F.: A fast and effective heuristic for the feedback arc set problem, *Information Processing Letters*, Vol. 47, No. 6, pp. 319–323 (1993).
[3] Meilă, M. and Pentney, W.: Clustering by weighted cuts in directed graphs, *Proceedings of the 2007 SIAM international conference on data mining*, SIAM, pp. 135–144 (2007).
[4] Leskovec, J., Lang, K. J. and Mahoney, M.: Empirical comparison of algorithms for network community detection, *Proceedings of the 19th international conference on World wide web*, pp. 631–640 (2010).
[5] Newman, M. E. and Girvan, M.: Finding and evaluating community structure in networks, *Physical review E*, Vol. 69, No. 2, p. 026113 (2004).
[6] Blondel, V. D., Guillaume, J.-L., Lambiotte, R. and Lefebvre, E.: Fast unfolding of communities in large networks, *Journal of statistical mechanics: theory and experiment*, Vol. 2008, No. 10, p. P10008 (2008).
[7] Xu, X., Yuruk, N., Feng, Z. and Schweiger, T. A.: Scan: a structural clustering algorithm for networks, pp. 824–833 (2007).
[8] Page, L., Brin, S., Motwani, R. and Winograd, T.: The PageRank citation ranking: Bringing order to the web., Technical report, Stanford InfoLab (1999).
[9] Haveliwala, T. H.: Topic-sensitive pagerank, *Proceedings of the 11th international conference on World Wide Web*, pp. 517–526 (2002).
[10] Kleinberg, J. M.: Authoritative sources in a hyperlinked environment, *Journal of the ACM (JACM)*, Vol. 46, No. 5, pp. 604–632 (1999).
[11] Katz, L.: A new status index derived from sociometric analysis, *Psychometrika*, Vol. 18, No. 1, pp. 39–43 (1953).
[12] Elo, A. E.: *The rating of chessplayers, past and present*, Arco Pub. (1978).