HITS アルゴリズムの拡張による Twitter 上での同一トピックオーソリティの検索

佐々木　心†　　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: †sasakikokoro@dl.soc.i.kyoto-u.ac.jp, ††tajima@i.kyoto-u.ac.jp

あらまし　本研究では，あるトピックに関する Twitter 上のオーソリティを見つけたい際に、そのトピックの既知のオーソリティを複数人指定すると，それらに共通するトピックの他のオーソリティを発見する手法を提案する．Twitter 上での特定トピックの人気ユーザ達は，多くの場合，そのトピックに関するコミュニティを形成し，また，そのトピックに関心を持つ多くのユーザがこれらのユーザをフォローしている．この構造は HITS アルゴリズムにおけるオーソリティとハブの関係に似ているが，HITS と異なりオーソリティ間のコミュニティ構造も考慮する必要がある．そこで本研究では，HITS にオーソリティ間の関係も取り入れたオーソリティ発見手法を提案する．

キーワード　Twitter，ソーシャルネットワーク分析，情報検索

1　はじめに

インターネットが普及した現代において，Twitter は情報発信の主要な場所となっている．Twitter 上では様々なユーザが情報発信を行っているが，その中には特定のトピックにおける人気ユーザが複数存在しており，そのトピックのコミュニティを形成している．またそのトピックに関心を持つ多くのユーザが，これらのユーザをフォローすることで情報を受け取っている．この関係は情報検索において考案された HITS アルゴリズム [1] におけるハブとオーソリティの関係に似ている．すなわち，あるトピックにおいて有用な情報を発信するユーザは多くのユーザからフォローされているオーソリティであり，またこれらのオーソリティを多くフォローしているユーザがハブである．Twitter を情報検索のツールとして用いる場合，このようなあるトピックのオーソリティを見つけることは重要である．

本研究の目的は，フォロー関係のネットワーク構造だけを用いて，あるトピックのオーソリティを探すことである．トピックの指定は，そのトピックのオーソリティであることが分かっているユーザを複数与えることで行う．複数人指定することによって意図するトピックがより明確になると考えられる．以後，指定されたユーザのことをソースユーザと呼ぶことにする．また探す範囲としてはソースユーザのフレンド，フォロワーを組み合わせたグラフ内で考える．具体的には，全ソースユーザの共通フレンド集合と共通フォロワー集合にソースユーザを合わせて得られるユーザ集合をノードとするグラフを作成する..

グラフを作成した後，アルゴリズムによってユーザをランキング付けすることでオーソリティを探す．図 1 に示しているように，トピックのオーソリティは密につながり合っているのに対し，トピック外の人気ユーザはトピックのオーソリティとのつながりが弱いと考えられる．しかし，HITS アルゴリズムでは，ハブからのリンクしか用いておらず，オーソリティ間の関係を考慮していない．そのため，指定したトピック以外の人気ユーザもオーソリティ度が大きくなり，上位にランキングされてしまう可能性がある．提案手法では HITS アルゴリズムを基に，オーソリティ間の関係を取り入れる．具体的にはオーソリティ間の相互フォローと共通フレンドを考えることで，この課題を解決する．

実験ではトピックの広さに応じて二種類の正解判定を行う．一つは比較的広い範囲のトピックで，例えば「ゲームクリエイター」，「漫画家」などがある．もう一つはあるトピックのサブトピックで，例えば「ゲームクリエイター，ゲームサウンドのクリエイター」，「漫画家，ジャンプ系列誌」などがある．最初に指定するソースユーザの数は二人とし，それぞれのトピックに対して二人組を複数選び，各組ごとにグラフを作成する．その後 HITS と提案手法それぞれでランキングを作成し，Precision@20，30 によって評価する．例えば，トピック「ゲームクリエイター，ゲームサウンドのクリエイター」の実験では，まず既知のゲームサウンドのクリエイターの二人組をいくつか選ぶ．そして，それぞれの組に対して，グラフを作成してからアルゴリズムによってオーソリティ度のランキングを作成し，「ゲームクリエイター」での正解判定と，「ゲームクリエイター，ゲームサウンドのクリエイター」での正解判定を行う．

実験の結果，四つのうち，三つのトピックで提案手法は比較手法よりも高い性能を示した．提案手法が良い性能を示したトピックでの実験結果より，Twitter でのオーソリティの検索においては，オーソリティ間の関係を考慮することが重要であると考えられる．またソースユーザを二人指定する場合，その選び方によって，提案手法の性能が比較手法よりも低くなることがあった．これは指定したソースユーザでは意図するトピックを明確に指定しきれず，提案手法のユーザ間のつながりを強く評

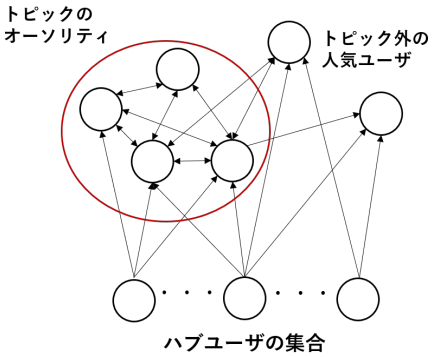

図 1　Twitter におけるハブとオーソリティの関係．トピックのオーソリティは互いに密につながっているが，トピック外の人気ユーザはトピックのオーソリティとの密なつながりを持たない．

価する性質から，異なるコミュニティのユーザを上位にランク付けたからだと考えられる．また，提案手法，比較手法双方において，トピックによる性能の違いがあり，トピックによってユーザ同士のつながり方が違うことが示唆された．これらの問題はソースユーザを増やすことにより解決され，実際に，ソースユーザを三人指定した場合，提案手法は比較手法よりも高い性能を示した．提案手法の性能が低かったトピックでは，意図したトピックとは別のコミュニティのユーザが上位にランク付けされてしまっていた．今後は，このような例でも精度を維持できるように，トピックによるネットワーク構造の違いなどを解析する必要がある．

2　関 連 研 究

この章では本研究に関連がある研究，アルゴリズムを紹介する．

2.1　HITS

本節では HITS アルゴリズム [1] について説明する．まず最初にアルゴリズムの手順について説明した後，その式を変形することで HITS を最大化問題として考える方法を示す．

2.1.1　概　　要

HITS は各ユーザに対してハブ度とオーソリティ度を割り当てるアルゴリズムである．ここでハブ度は多くのオーソリティをフォローしているほど大きく，オーソリティ度は多くのハブからフォローされているほど大きいとされる．具体的には，ハブ度はフォローしているユーザのオーソリティ度の合計，オーソリティ度はフォロワーのハブ度の合計として相互再帰的に定義される．

2.1.2　アルゴリズム

ユーザ i のハブ度とオーソリティ度をそれぞれ h_i, a_i として，これらを成分とするベクトルを $h = (h_1 \cdots h_n)^T$，$a = (a_1 \cdots a_n)^T$ で表す．ただし n はユーザ数を表す．また隣接行列 $L = [l_{ij}]$ を

$$l_{ij} = \begin{cases} 1 & (i \text{ が } j \text{ をフォローしている}) \\ 0 & (\text{それ以外}) \end{cases}$$

とする．

このときアルゴリズムの更新式は以下の式で表される．

$$\begin{cases} h = \mu L a \\ a = \eta L^T h \end{cases}$$

ここで，μ，η は正規化定数とする．これらの式を用いて h と a を反復計算することで最終的な値を得る．

2.1.3　最大化問題としての HITS

この節では，HITS アルゴリズムが最大化問題として定式化できる [2], [3] ことを説明する．

前節で示した更新式において，h を a の式に代入することで，

$$a = \mu\eta L^T L a$$

と表すことができる．$\mu\eta$ を改めて λ とおくと，反復計算によって a の収束値を求めることは，行列 $L^T L$ の固有値 λ に対応する固有ベクトルを求めることになる．特に，この式を用いてべき乗法で固有ベクトルを求めれば，最大固有値に対応する固有ベクトルが求める値になる．

また行列 $L^T L$ は対称行列であることから，最大固有値に対応する固有ベクトルを求めることは，以下の最大化問題を解くことになる．

$$a = \arg\max_{x^T x = 1} x^T L^T L x$$

さらに右辺を書き換えると以下のようになる．

$$\begin{aligned} x^T L^T L x &= \sum_i \sum_j x_i l_i^T l_j x_j \\ &= \sum_i x_i^2 f_i + \sum_{i \neq j} x_i x_j b_{ij} \end{aligned}$$

ここで，l_i は隣接行列 L の i 番目の列ベクトルを表し，f_i はユーザ i のフォロワー数，b_{ij} はユーザ i と j の共通フォロワーの数を表している．

最後の式を最大化するためには，フォロワー数が多いユーザのオーソリティ度を大きくし，かつ，多くのユーザから同時にフォローされているユーザのペアのオーソリティ度をどちらも大きくすれば良い．これより，HITS アルゴリズムでは，フォロワー数という個人の人気度と，共通フォロワー数というオーソリティ間の類似度をを考慮した人気度を用いて，オーソリティ度を計算していることが分かる．

2.2　Twitterrank

Twitterrank [4] は，PageRank [5] にユーザのトピックの類似性を加えて拡張したアルゴリズムである．ユーザ j のフレンド集合を Fr_j とすると，あるトピックが与えられた上でのユーザ j から i への遷移行列 P は以下のようになっている．

$$p_{ij} = \frac{|T_i|}{\sum_{k \in Fr_j} |T_k|} sim(i, j)$$

$|T_i|$ はユーザ i の投稿した全ツイートの数を表している．$sim(i, j)$ はユーザ i と j の与えられたトピックにおける類似度を表しており，DT_i でユーザ i が投稿したツイートの中で，与えられた

トピックに関連したものの割合を表せば，

$$sim(i, j) = 1 - |DT_i - DT_j|$$

と表される．つまり，与えられたトピックに関連したツイートをつぶやいている割合が近いユーザほど，類似度は大きくなる．また，PageRank では等確率にランダムジャンプを行うが，Twitterrank では，全ユーザの与えられたトピックに関連するツイート数のうち，ユーザ i がつぶやいた割合を用いる．

Twitterrank ではツイートの情報を用いて，明示的にトピックの類似度を計算しているため，ネットワークの構造だけを用いるよりも性能は良いと考えられる．しかし，ツイートの情報を取得できないユーザが存在したり，トピックが変わるごとに，あるツイートが与えられたトピックと関連しているかを判定する仕組みを用意しなければならないといった問題点もある．本研究ではネットワークの構造のみを用いて，トピックのオーソリティを探すことを目的としている．

2.3 SimRank

本節では SimRank アルゴリズム [6] を簡単に説明する．SimRank はユーザの類似度を計算するためのアルゴリズムであり，ソースユーザに似たユーザを探すという点で本研究と関連がある．

ユーザ a と b の類似度を $s(a, b)$ と表す．このとき類似度は以下の式で定義される．

$$s(a,b) = \begin{cases} 1 & (a = b) \\ 0 & (|I(a)| = 0 \text{ or } |I(b)| = 0) \\ \dfrac{C}{|I(a)||I(b)|} \displaystyle\sum_{i \in I(a),\, j \in I(b)} s(i, j) & (\text{それ以外}) \end{cases}$$

ここで，$I(x)$ はユーザ x のフォロワー集合，C は減衰率を表す．この式から分かるように，SimRank では似たユーザからフォローされているほど類似度が高くなるようになっている．

2.4 ニューラルネットワーク

ディープラーニングは様々なタスクで高い性能を出しており，グラフデータに対しても GNN [7] や GCN [8] などのようなニューラルネットワークモデルを適用する研究が盛んになっている．また，モデルの解釈性についての研究 [9] や，少数のラベル付きデータに対する研究 [10] なども行われている．本研究では，アルゴリズムの解釈性や計算コストを重視し，グラフニューラルネットワークを用いないことにする．

3 提案手法

本章では，前章の HITS アルゴリズムの考察をもとに，オーソリティ間の関係を取り入れたアルゴリズムを提案する．提案手法は HITS の時のように，最大化問題として定式化し，段階的に二つの式を提案する．

3.1 提案手法（1）

一つ目の式を以下に示す．

$$a = \arg\max_{x^T x = 1} \sum_{i \neq j} x_i x_j b_{ij} m_{ij}$$

ここで，m_{ij} は以下のように定義する．

$$m_{ij} = \begin{cases} 1 & (\text{i と j が相互フォローしている}) \\ 0 & (\text{それ以外}) \end{cases}$$

この式は，HITS の最適化式の二つ目の項において，ユーザ i と j を相互フォローのユーザだけに制限した式である．個人のフォロワー数を用いた場合，異なるトピックのオーソリティのランキング順位を上げてしまうと考えられるため，最大化の式からは外している．同じトピックのオーソリティたちは共通フォロワーが多いと考えられるため，共通フォロワーが多いユーザたちのオーソリティ度を大きくすることは自然である．しかし，相互フォローの制約を入れない場合，異なるトピックのオーソリティであっても共通フォロワーが多いとオーソリティ度が大きくなってしまうという問題がある．同じトピックのオーソリティ同士は互いにフォローし合うことで密なコミュニティを作っていると考えられるため，相互フォローの制約を入れることでこの問題に対処する．

実際にこの式を最大化してオーソリティ度を求める方法についても説明する．この式は i と j に対して対称な形をしており，$B = [b_{ij}]$，$M = [m_{ij}]$ とおいて，$\otimes$ で要素ごとの積を表せば，以下のように書き換えることができる．

$$\sum_{i \neq j} a_i a_j b_{ij} m_{ij} = a^T (B \otimes M) a$$

B と M が対称行列であることから，この式は対称行列に対する二次形式の形になっているため，この式の最大化は，最大固有値に対応する固有ベクトルを求めることになる．つまり，オーソリティ度は，以下の式に基づいたべき乗法で求めることができる．

$$a = (B \otimes M) a$$

B は $L^T L$ から対角要素を除くことで得られ，M は $L \otimes L^T$ によって得られるため，隣接行列から簡単に更新式を作ることができる．

3.2 提案手法（2）

二つ目の式は以下の式で表される．

$$a = \arg\max_{x^T x = 1} \sum_{i \neq j} x_i x_j m_{ij} \sum_k (b_{ik} + b_{jk}) m_{ik} m_{jk}$$

この式では，図 2 で示しているように，相互フォローしているユーザ i と j に対し，i とも j とも相互フォローであるユーザ k それぞれに対して，i と k，j と k の共通フォロワー数を足し合わせている．

提案手法（1）では，相互フォローであるユーザ i と j の共通フォロワーを考えた．しかし，特にサブトピックでのクエリを考えた場合，サブトピックのオーソリティはより広いトピックでのオーソリティとも相互フォローであることが多い．つまり，ユーザ i がサブトピックのオーソリティで，ユーザ j がそ

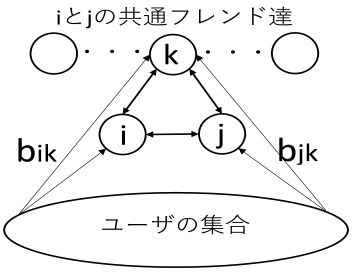

図 2　提案手法（2）の式の意味を表したグラフ．相互フォローであるユーザ i と j に対し，二人と相互フォローである共通フレンド k 全員を考え，i と k，j と k の共通フォロワー数を足し合わせる．

のサブトピックに含まれない，広いトピックでのオーソリティである場合にも，ユーザ i と j が相互フォローである可能性が高い．また，そのようなユーザ i と j の共通フォロワー数も多いと考えられる．そのため，共通フォロワー数を考えるだけでは，ユーザ i と j がどちらもサブトピックのオーソリティである場合と，片方が広いトピックでのオーソリティである場合を区別できず，サブトピックには含まれないがより広いトピックには含まれるオーソリティが，上位にランク付けされる可能性がある．

よって，提案手法（2）では，相互フォローしているユーザ i と j の共通フレンド達，特に相互フォローである共通フレンド達を考える．これは，サブトピックのオーソリティは，広いトピックのオーソリティよりも，サブトピック内で密なコミュニティを形成していると考えられるからである．共通フォロワー数 b_{ik} と b_{jk} を足し合わせることで，ユーザ i，j とその共通フレンド k が似ているほど，またそのような共通フレンドが多いほど値を大きくすることができる．これにより，ユーザ i と j の片方が広いトピックでのオーソリティである場合，両方がサブトピックのオーソリティである場合よりも，最適化式のユーザ i と j に対応する項の値が小さくなり，広いトピックでのオーソリティのオーソリティ度も下がると考えられる．

実際にオーソリティ度を求める方法は提案手法（1）と同様であり，オーソリティ度は以下の式に基づいたべき乗法で求められる．

$$a = (((B \otimes M)M \ + \ ((B \otimes M)M)^T) \otimes M)a$$

4 実　　験

本章では実験方法と，その結果について述べる．実験ではトピックを指定する必要があるが，ソースユーザを選んだ際に，意図しているトピックが広さに応じて複数考えられることがある．本研究では，あるトピックの中に含まれるサブトピックのオーソリティの検索を考え，もとの広いトピックとサブトピック二つでの評価を行う．

4.1 手　　順

実験は以下の手順で行う

（1） クエリに用いるトピックを選ぶ．

表 1　実験に用いたソースユーザの組．Twitter アカウントのユーザ名を表示している．

トピック	ソースユーザのペア	クエリ番号
（1）	midiplex，yuzokoshiro	（1-1）
（1）	KoudenMS，midiplex	（1-2）
（1）	KoudenMS，yuzokoshiro	（1-3）
（1）	KoudenMS，midiplex，yuzokoshiro	（1-4）
（2）	NEBU_KURO，syu1aso	（2-1）
（2）	k_usuta，syu1aso	（2-2）
（2）	k_usuta，NEBU_KURO	（2-3）
（2）	k_usuta，NEBU_KURO，syu1aso	（2-4）
（3）	arimorokoshi415，nasunakanakani	（3-1）
（3）	nasunakanakani，shinomiyaakira	（3-2）
（3）	arimorokoshi415，shinomiyaakira	（3-3）
（3）	arimorokoshi415，nasunakanakani，shinomiyaakira	（3-4）
（4）	ayatsujiyukito，honobu_yonezawa	（4-1）
（4）	honobu_yonezawa，michioshusuke	（4-2）
（4）	ayatsujiyukito，michioshusuke	（4-3）
（4）	ayatsujiyukito，honobu_yonezawa，michioshusuke	（4-4）

（2） 選んだトピックの既知のオーソリティを三人選ぶ．

（3） 三人の中から二人組，あるいは三人組をソースユーザとして選ぶ

（4） すべてのソースユーザの共通フレンド，共通フォロワーのユーザ集合を取得し，それらの集合にソースユーザを加えたすべてのユーザをノードとするグラフを作成する．

（5） 作成したグラフに対して提案手法と比較手法を用いてランキングを作成し，上位のユーザに対してトピックのオーソリティかどうかを判定する．

4.2 実験に用いたデータ

実験では，クエリに用いるトピックとして（1）「ゲームクリエイター，ゲームサウンドのクリエイター」，（2）「漫画家，ジャンプ系列誌」，（3）「お笑い芸人，松竹芸能」，（4）「作家，ミステリ・推理」を選んだ．ここで各トピックは「広いトピック，サブトピック」の順に示してある．今回の実験に選んだソースユーザの組を表 1 に示す．

4.3 評 価 方 法

評価指標は上位 20，30 人における適合率（以後 P@20，P@30 と表記する．）を用いる．上位 20，30 人で評価するのは，現実に検索する際に，20 人から 30 人ほどオーソリティを知りたいと仮定しているからである．

また，他の指標としては MAP（Mean Average Precision）[11] が考えられる．しかし，MAP では正解が上位にあるほど，過剰に評価する傾向があり，今回の問題設定では，上位のユーザが得られた後では，その中でのランキングは不要だと考えられるため，MAP ではなく適合率で評価する．

正解判定は広いトピックでの判定と、サブトピックでの判定どちらも行う．今回の実験では客観的に正解判定が可能なトピックを選んでおり，各トピックでの具体的な正解判定は Twitter のプロフィールやつぶやき，また Web から得られる情

表 2　実験で用いた手法

手法	略称
HITS	HITS
PageRank	PR
フォロワー数	FN
提案手法（1）	（1）
提案手法（1）相互フォロー制約なし	（1’）
提案手法（2）	（2）
提案手法（2）相互フォロー制約なし	（2’）
提案手法（1）+（2）	（3）
提案手法（1）+（2）相互フォロー制約なし	（3’）

表 3　ゲームクリエイターでの実験結果（P@20）

クエリ番号	HITS	PR	FN	（1）	（1’）	（2）	（2’）	（3）	（3’）
（1-1）	0.75	0.80	0.70	0.80	0.75	**1.00**	0.75	**1.00**	0.75
（1-2）	0.90	0.80	0.90	**1.00**	0.90	**1.00**	**1.00**	**1.00**	**1.00**
（1-3）	0.95	0.90	0.90	**1.00**	0.95	**1.00**	0.95	**1.00**	0.95
（1-4）	0.95	**1.00**	0.95	**1.00**	**1.00**	**1.00**	0.95	**1.00**	0.95
平均	0.89	0.88	0.86	0.95	0.90	**1.00**	0.91	**1.00**	0.91

報をもとに行っている．

4.4　実験で用いるアルゴリズム

実験では比較手法として HITS，PageRank，フォロワー数を用いる．PageRank はランダムジャンプをしない確率を 0.85 とする．また提案手法において，相互フォローの制約がない場合，それから提案手法の二つの式を合わせた場合も比較として実験する．提案手法（2）から相互フォローの制約を除く場合は，式に含まれる全ての相互フォロー制約を外すことにする．つまり以下の式を用いる．

$$a = \arg\max_{x^T x=1} \sum_{i \neq j} x_i x_j \sum_k (b_{ik} + b_{jk})$$

また，提案手法の二つの式を合わせる場合は，単純に両方の式を足した以下の式を用い，相互フォローの制約がない場合も同様にする．

$$a = \arg\max_{x^T x=1} \sum_{i \neq j} x_i x_j b_{ij} m_{ij} + \sum_{i \neq j} x_i x_j m_{ij} \sum_k (b_{ik} + b_{jk}) m_{ik} m_{jk}$$

表 2 に実験で用いる手法をまとめる．

4.5　結　　果

まず最初に広いトピックで評価した場合の結果を示し，次にサブトピックで評価した場合の結果を示す．適合率の値は少数第三位を四捨五入している．

4.5.1　広いトピック

表 3 と表 4 に「ゲームクリエイター」での評価結果を，表 5 と表 6 に「漫画家」での評価結果を，表 7 と表 8 に「お笑い芸人」での評価結果を，表 9 と表 10 に「作家」での評価結果を示す．また，表 11 と表 12 に，全トピックにおける，ソースユーザが二人の場合の平均，三人の場合の平均，全クエリの平均の評価結果を示す．

結果から，トピック（1），（2），（3）では，提案手法（2）の性能が概ね最も良いことが分かる．しかし，トピック（4）では，全体的に他のトピックよりも性能が低く，かつ提案手法（2）ではほとんど不正解となってしまっている．そのため，全トピックの平均で見ると，提案手法（2）の性能が，PageRank よりも低くなっている．

表 4　ゲームクリエイターでの実験結果（P@30）

クエリ番号	HITS	PR	FN	（1）	（1’）	（2）	（2’）	（3）	（3’）
（1-1）	0.73	0.83	0.67	0.80	0.73	**1.00**	0.83	**1.00**	0.83
（1-2）	0.93	0.86	0.93	**1.00**	0.93	**1.00**	0.93	**1.00**	0.93
（1-3）	0.93	0.93	0.90	**1.00**	0.93	**1.00**	0.97	**1.00**	0.97
（1-4）	0.96	0.96	0.97	**1.00**	0.97	**1.00**	0.97	**1.00**	0.97
平均	0.89	0.90	0.87	0.95	0.89	**1.00**	0.93	**1.00**	0.93

表 5　漫画家での実験結果（P@20）

クエリ番号	HITS	PR	FN	（1）	（1’）	（2）	（2’）	（3）	（3’）
（2-1）	0.85	0.90	0.85	0.90	0.85	**1.00**	0.85	**1.00**	0.85
（2-2）	0.80	0.80	0.75	0.90	0.80	**1.00**	0.90	**1.00**	0.90
（2-3）	0.80	0.85	0.80	0.90	0.80	**0.95**	0.75	**0.95**	0.75
（2-4）	0.85	0.85	0.85	0.90	0.85	**1.00**	0.85	**1.00**	0.85
平均	0.83	0.85	0.81	0.90	0.83	**0.99**	0.84	**0.99**	0.84

表 6　漫画家での実験結果（P@30）

クエリ番号	HITS	PR	FN	（1）	（1’）	（2）	（2’）	（3）	（3’）
（2-1）	0.87	0.90	0.83	0.93	0.87	0.93	0.80	**0.97**	0.80
（2-2）	0.80	0.83	0.73	0.83	0.80	**0.93**	0.77	**0.93**	0.77
（2-3）	0.77	0.83	0.73	0.87	0.77	0.87	0.73	**0.90**	0.73
（2-4）	0.83	0.83	0.83	0.87	0.83	**0.90**	0.73	**0.90**	0.73
平均	0.82	0.85	0.78	0.88	0.82	0.91	0.76	**0.93**	0.76

表 7　お笑い芸人での実験結果（P@20）

クエリ番号	HITS	PR	FN	（1）	（1’）	（2）	（2’）	（3）	（3’）
（3-1）	0.70	**0.85**	0.57	0.70	0.70	0.80	0.75	0.80	0.75
（3-2）	**1.00**	**1.00**	**1.00**	0.95	**1.00**	0.95	0.95	0.95	0.95
（3-3）	0.75	**0.85**	0.80	0.75	0.80	0.80	0.75	0.80	0.75
（3-4）	0.80	0.80	0.80	0.80	0.75	**0.90**	0.70	**0.90**	0.70
平均	0.81	**0.88**	0.79	0.80	0.81	0.86	0.79	0.86	0.79

表 8　お笑い芸人での実験結果（P@30）

クエリ番号	HITS	PR	FN	（1）	（1’）	（2）	（2’）	（3）	（3’）
（3-1）	0.60	0.60	0.57	0.63	0.60	**0.77**	0.63	**0.77**	0.63
（3-2）	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**
（3-3）	**0.80**	0.70	0.77	0.73	**0.80**	0.73	0.73	0.73	0.73
（3-4）	0.73	0.73	0.67	0.73	0.73	**0.80**	0.70	0.77	0.70
平均	0.78	0.75	0.75	0.77	0.78	**0.82**	0.76	0.81	0.76

表 9　作家での実験結果（P@20）

クエリ番号	HITS	PR	FN	（1）	（1’）	（2）	（2’）	（3）	（3’）
（4-1）	0.45	**0.60**	0.50	**0.60**	0.45	0.10	0.35	0.10	0.35
（4-2）	0.25	**0.45**	0.40	0.25	0.25	0.00	0.15	0.00	0.15
（4-3）	0.35	**0.45**	0.40	0.40	0.35	0.00	0.30	0.00	0.30
（4-4）	0.40	**0.50**	**0.50**	0.35	0.35	0.00	0.25	0.00	0.25
平均	0.36	**0.50**	0.45	0.40	0.35	0.03	0.26	0.03	0.26

表 10　作家での実験結果（P@30）

クエリ番号	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
(4-1)	0.43	**0.60**	0.47	0.53	0.43	0.23	0.33	0.27	0.33
(4-2)	0.30	**0.47**	0.37	0.20	0.30	0.00	0.17	0.00	0.17
(4-3)	0.33	**0.43**	0.40	0.33	0.33	0.00	0.27	0.00	0.27
(4-4)	0.37	**0.40**	**0.40**	0.33	0.37	0.03	0.20	0.03	0.20
平均	0.36	**0.48**	0.41	0.35	0.36	0.07	0.24	0.08	0.24

表 11　広いトピックでの平均実験結果（P@20）

	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
二人	0.71	**0.77**	0.71	0.76	0.72	0.72	0.70	0.72	0.70
三人	0.75	**0.79**	0.78	0.76	0.74	0.73	0.69	0.73	0.69
全クエリ	0.72	**0.76**	0.73	0.76	0.72	0.72	0.70	0.72	0.70

表 12　広いトピックでの平均実験結果（P@30）

	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
二人	0.70	**0.75**	0.70	0.74	0.71	0.70	0.68	0.71	0.68
三人	0.72	**0.73**	0.72	**0.73**	**0.73**	0.68	0.65	0.68	0.65
全クエリ	0.71	**0.74**	0.70	0.73	0.71	0.70	0.67	0.70	0.67

表 13　ゲームサウンドのクリエイターでの実験結果（P@20）

クエリ番号	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
(1-1)	0.55	0.65	0.50	0.65	0.55	**1.00**	0.60	**1.00**	0.60
(1-2)	0.80	0.70	0.75	0.95	0.80	**1.00**	0.95	**1.00**	0.95
(1-3)	0.85	0.80	0.75	0.95	0.85	**1.00**	0.90	**1.00**	0.90
(1-4)	0.90	0.95	0.90	0.95	0.95	**1.00**	**1.00**	**1.00**	**1.00**
平均	0.78	0.78	0.73	0.88	0.79	**1.00**	0.86	**1.00**	0.86

表 14　ゲームサウンドのクリエイターでの実験結果（P@30）

クエリ番号	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
(1-1)	0.60	0.70	0.53	0.70	0.60	**0.97**	0.70	**0.97**	0.70
(1-2)	0.83	0.73	0.76	0.97	0.83	**1.00**	0.90	**1.00**	0.90
(1-3)	0.83	0.83	0.77	0.97	0.87	**1.00**	0.93	**1.00**	0.93
(1-4)	0.93	0.93	0.90	0.97	0.93	**1.00**	0.93	**1.00**	0.93
平均	0.80	0.80	0.74	0.90	0.81	**0.99**	0.87	**0.99**	0.87

表 15　ジャンプ系列誌の漫画家での実験結果（P@20）

クエリ番号	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
(2-1)	0.70	**0.90**	0.75	0.80	0.70	0.80	0.65	0.80	0.65
(2-2)	0.75	0.80	0.70	0.85	0.80	**0.95**	0.80	**0.95**	0.80
(2-3)	0.55	0.65	0.55	0.75	0.55	**0.90**	0.50	**0.90**	0.50
(2-4)	0.80	0.85	0.75	0.85	0.80	**1.00**	0.70	**1.00**	0.70
平均	0.70	0.80	0.69	0.81	0.71	**0.91**	0.66	**0.91**	0.66

4.5.2　サブトピック

表 13 と表 14 に「ゲームサウンドのクリエイター」での評価結果を，表 15 と表 16 に「ジャンプ系列誌の漫画家」での評価結果を，表 17 と表 18 に「松竹芸能のお笑い芸人」での評価結果を，表 19 と表 20 に「ミステリ・推理作家」での評価結果を示す．また，表 11 と表 12 に，全トピックにおける，ソースユーザが二人の場合の平均，三人の場合の平均，全クエリの平均の評価結果を示す．

結果から，トピック（1），（2），（3）では，平均的に，またソースユーザが三人の場合に，提案手法（2）の性能が最も良くなっている．しかし，広いトピック同様，トピック（4）は，全体的に性能が低く，提案手法（2）もほとんど不正解となってしまっている．そのため，全トピックの平均で見ると，提案手法（2）の性能が，PageRank よりも低くなっている．

表 16　ジャンプ系列誌の漫画家での実験結果（P@30）

クエリ番号	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
(2-1)	0.60	**0.73**	0.57	0.67	0.60	0.70	0.57	**0.73**	0.57
(2-2)	0.67	0.73	0.63	0.73	0.70	**0.83**	0.67	**0.83**	0.67
(2-3)	0.53	0.60	0.50	0.60	0.53	**0.80**	0.53	**0.80**	0.53
(2-4)	0.70	0.80	0.70	0.73	0.70	**0.87**	0.60	**0.87**	0.60
平均	0.63	0.72	0.60	0.68	0.63	0.80	0.59	**0.81**	0.59

表 17　松竹芸能のお笑い芸人での実験結果（P@20）

クエリ番号	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
(3-1)	0.50	**0.65**	0.40	0.50	0.50	**0.65**	0.55	**0.65**	0.55
(3-2)	0.35	**0.45**	0.35	0.25	0.35	0.25	0.25	0.25	0.25
(3-3)	0.20	0.40	0.25	0.30	0.20	**0.45**	0.30	**0.45**	0.30
(3-4)	0.65	0.75	0.65	0.70	0.65	**0.85**	0.65	**0.85**	0.70
平均	0.43	**0.56**	0.41	0.44	0.43	0.55	0.44	0.55	0.44

表 18　松竹芸能のお笑い芸人での実験結果（P@30）

クエリ番号	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
(3-1)	0.40	0.43	0.40	0.43	0.40	**0.67**	0.43	**0.67**	0.43
(3-2)	0.30	**0.43**	0.33	0.27	0.30	0.30	0.30	0.30	0.30
(3-3)	0.23	0.30	0.23	0.27	0.23	**0.37**	0.23	0.33	0.23
(3-4)	0.63	0.70	0.57	0.63	0.63	**0.77**	0.60	0.73	0.63
平均	0.39	0.47	0.38	0.40	0.39	**0.53**	0.39	0.51	0.40

表 19　ミステリ・推理作家での実験結果（P@20）

クエリ番号	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
(4-1)	0.45	**0.60**	0.50	**0.60**	0.45	0.10	0.35	0.10	0.35
(4-2)	0.25	**0.45**	0.40	0.25	0.25	0.00	0.15	0.00	0.15
(4-3)	0.35	**0.45**	0.40	0.40	0.35	0.00	0.30	0.00	0.30
(4-4)	0.40	**0.50**	**0.50**	0.35	0.35	0.00	0.25	0.00	0.25
平均	0.36	**0.50**	0.45	0.40	0.35	0.03	0.26	0.03	0.26

表 20　ミステリ・推理作家での実験結果（P@30）

クエリ番号	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
(4-1)	0.43	**0.60**	0.47	0.53	0.43	0.17	0.33	0.20	0.33
(4-2)	0.30	**0.47**	0.37	0.20	0.30	0.00	0.17	0.00	0.17
(4-3)	0.33	**0.43**	0.40	0.33	0.33	0.00	0.27	0.00	0.27
(4-4)	0.37	**0.40**	**0.40**	0.33	0.37	0.00	0.20	0.00	0.20
平均	0.36	0.48	0.41	0.35	0.36	0.04	0.24	0.05	0.24

表 21　サブトピックでの平均実験結果（P@20）

	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
二人	0.52	**0.63**	0.53	0.60	0.53	0.59	0.53	0.59	0.53
三人	0.69	**0.76**	0.70	0.71	0.69	0.71	0.65	0.71	0.66
全クエリ	0.57	**0.66**	0.57	0.63	0.57	0.62	0.57	0.62	0.56

5　考　　察

この章では実験結果をもとに考察を述べる．結果を見て分か

表 22　サブトピックでの平均実験結果（P@30）

	HITS	PR	FN	(1)	(1')	(2)	(2')	(3)	(3')
二人	0.50	**0.58**	0.50	0.56	0.51	0.57	0.50	0.57	0.50
三人	0.66	**0.71**	0.64	0.67	0.66	0.66	0.58	0.65	0.59
全クエリ	0.54	**0.61**	0.53	0.58	0.55	0.59	0.52	0.59	0.52

るとおり，トピック（1），（2），（3）では提案手法の性能は高いが，トピック（4）ではかなり低くなっている．よって，まず最初にトピック（1），（2），（3）の結果を用いて考察を行い，それをもとに，なぜトピック（4）でうまくいかなかったかを考察する．以下の節で，アルゴリズムやクエリについて複数の観点から考察を行う．

5.1　共通フォロワー数

この節では，アルゴリズムにおける共通フォロワー数の有用性について考える．共通フォロワー数はその二人のユーザのある種の類似度を表していると考えられ，個人のフォロワー数を用いるよりも良いと予想される．HITS では最適化の式に，個人のフォロワー数の項と共通フォロワー数の項があり，また，手法（1'）では，HITS の式から個人のフォロワー数の項を外した式を用いている．よって HITS と手法（1'）とフォロワー数の実験結果を比較してみると，HITS と手法（1'）では大差がないが，フォロワー数は，これらよりも少し性能が低いことが分かる．このことから個人のフォロワー数だけよりも，共通フォロワー数だけを用いた方が良く，またこれらを合わせた場合はあまり変わらないと考えられる．

しかし，広いトピックでの実験では，個人のフォロワー数だけでも，およそ８割程度の適合率を達成しており，広いトピック内でオーソリティを知りたいのであれば，フォロワー数のような個人の人気度だけでも，ある程度は分かると考えられる．

5.2　相互フォロー

この節では，アルゴリズムにおける相互フォローの制約について考える．相互フォロー制約は，手法（1），（2），（3）に含まれており，最大化の式におけるユーザ i と j を相互フォローのユーザ同士のみに制限するというものである．この制約を外したものが手法（1'），（2'），（3'）であり，これらと手法（1），（2），（3）をそれぞれ比較してみると，ほとんどのクエリで，制約ありの方が性能が良いことが分かる．このことから，制約を入れることで，異なるトピックの人気ユーザのオーソリティ度を下げ，意図したトピックのオーソリティのオーソリティ度を上げることが出来ており，相互フォローをアルゴリズムに取り入れることが大切であると考えられる．

5.3　共通フレンド

この節では，アルゴリズムにおいて共通フレンドを考える有用性について考察する．共通フレンドは提案手法（2）において考えたものであり，最大化の式において，ユーザ i と j のすべての共通フレンド k に対して，i と k，j と k の共通フォロワー数を足し合わせたものである．特に，手法（2）では相互フォロー制約を入れているので，これは互いに相互フォローであるユーザ三人組を考えていることになる．異なるトピックのユーザよりも，探しているトピックのオーソリティの方が，そのトピックの他のオーソリティをより多くフォローしていると考えられるため，共通フォロワー数だけでは区別できない，異なるトピックの人気ユーザのオーソリティ度を低くするように意図している．これは特に，サブトピックのオーソリティ検索のように，異なるトピックの人気ユーザであっても，探しているトピックのオーソリティともつながりがある場合に有効だと考えている．

まず手法（1）と（2）と比べると，ほとんどのクエリで手法（2）の方が性能が良いことが分かる．また広いトピックでの評価結果より，サブトピックでの評価結果の方が性能差が大きい．このことから，上で述べた意図通りに，共通フォロワー数だけでなく，共通フレンドを考慮する方が良いことが分かる．

次に PageRank と手法（2）を比較する．この時，個々のクエリで見ると，勝ったり負けたりしているが，平均的にはほぼ同じか，手法（2）の方が性能が良いことが分かる．クエリによって PageRank よりも手法（2）の方が性能が悪いことがあるのは，手法（2）の密なつながりを評価するという性質のためであると考えられる．つまり，手法（2）によるランキングの上位のユーザたちは，何らかのコミュニティは形成しているが，それが意図したトピックと異なる場合に性能が低くなってしまうと考えられる．例えば，クエリ（3-1），（3-2）のサブトピックでの評価結果を見ると，同じトピックでソースユーザが一人共通しているにもかかわらず，手法（2）での性能が大きく異なっている．これは，クエリ（3-1）では，選んだ二人のソースユーザによって，「松竹芸能のお笑い芸人」というトピックを指定し切れてないと考えられる．この，選んだソースユーザによってトピックが指定しきれない問題は，ソースユーザを増やすことで解決できると考えられる．実際，三人のソースユーザで実験を行ったクエリ（1-4），（2-4），（3-4）ではすべて PageRank よりも性能が良く，また二人のソースユーザで行った場合と比較してももっとも性能が良くなっている．

最後に手法（2）と（3）を比較する．手法（3）は手法（1）と（2）の式を合わせたものを最大化している．結果を見ると，手法（2）と（3）ではほとんど性能が変わらないことが分かる．これは，式を見ると，手法（1）では二人の共通フォロワー数だけを考えているのに対し，手法（2）では共通フレンドの数だけ共通フォロワー数を足し合わせており，（2）の式の方が明らかに大きくなるため，手法（2）の場合とあまり変わらないと考えられる．

5.4　トピックによる違い

この節ではクエリによる実験結果の違いをトピックの観点から考察する．トピックごとに見た場合，トピック（1），（2）と比較してトピック（3）の性能が，広いトピックでもサブトピックでも低くなっており，また広いトピックからサブトピックへの性能の下がり方も大きくなっている．これには二つの理由があると考えられる．まず最初に，広いトピックでも性能が低いが、これは「お笑い芸人」のユーザは，他のトピックのユーザ

ともつながりが多いからだと考えられる．実際，ランキング上位の不正解のユーザは，ほとんどがタレントや放送作家であった．よって，広いトピックでの結果の差は，そのトピックがそれ自体でどれだけ閉じているかを示していると予想される．次に，サブトピックで性能がかなり低くなっているが，これは「お笑い芸人」の中では，「松竹芸能」のような事務所ごとのつながり以外にも，同期や共演者，仲の良さと言った他のつながりの可能性があるからだと考えられる．よってサブトピックの結果が広いトピックでの結果よりもかなり低い場合は，意図したサブトピック以外のつながりが多いことを示していると予想される．これらの問題は，こちらの意図するトピックとのずれが原因であり，前節で考察したように，ソースユーザを増やすことで解決されると考えられる．

5.5 トピック（4）の性能が低い理由

この節では，トピック（4）の「作家，ミステリ・推理」で，提案手法（2）の性能が低い理由を考察する．まず実際のランキング結果を見てみると，どの手法でも，上位にランキングされた不正解のユーザのほとんどが，出版社のアカウントであった．これまでの考察より，提案手法（2）はつながりを強く評価することから，提案手法（2）では，出版社のコミュニティを，間違えて上位にランク付けしてしまったと考えられる．提案手法（2）では，共通フレンドが多いだけでなく，それらの共通フォロワーも多くなければオーソリティ度は大きくならない．よって，ソースユーザから作成したグラフ内では，意図しているトピックのオーソリティのフォロワー数が多くなるため，意図したトピックとは異なるトピックのコミュニティが存在しても，意図しているトピックのオーソリティよりオーソリティ度が大きくなることはないと想定していた．しかし，今回の出版社のコミュニティは，フォロワー数のランキングでも上位に来ていることから，作成したグラフ内での人気度が高くなっている．従って，提案手法（2）では，よりつながりの強いコミュニティとして，出版社のアカウントを上位にランクづけてしまったと考えられる．

6 結論と展望

本研究では，Twitter 上のあるトピックのオーソリティを，ネットワーク構造だけを用いて見つけることを目的とし，HITS の最適化の式にオーソリティ間の関係を取り入れた手法を提案した．HITS では，ハブとオーソリティという関係からオーソリティ度を計算しているが，オーソリティ同士のつながりは考慮していないため，検索したいトピックに含まれない人気ユーザを除外することが出来ない．提案手法の式では，まず最初に相互フォローの制約を入れ，これにより，単純なフォロワー数や HITS よりも性能が上がることが分かった．また共通フォロワーだけでなく共通フレンドも考慮することによって，オーソリティ間のつながりをさらに考慮することができ，HITS や PageRank よりも性能が上がることが分かった．この提案手法は，オーソリティ間のつながりを強く評価するため，ソースユーザの選び方や，トピックによるつながり方の違いに対処するためにも，ソースユーザは増やした方が良いことも実験から示された．また，現在の提案手法では，意図したトピックとは異なるコミュニティを，上位ユーザとして評価してしまう場合があることも分かった．

今後の展望としては，まず実験に用いたネットワークの解析を行うことが挙げられる．現在はどのトピックでも，トピックのオーソリティが存在し，強いつながりを持っているという仮定の下，実験を行っていた．しかし，現実にそのような構造があるかどうかの解析は出来ておらず，また，トピックごとにネットワークの構造に違いがあることも考えられる．特に，トピック（4）のような，別のコミュニティが存在する場合にもうまくいくような手法を考える必要がある．従って，例えばネットワークを可視化してみて，どのような構造になっているかを調べたり，ネットワークのクラスタ係数や次数分布のような指標を調べて，それらと今回の実験結果を照らし合わせて考察を行う必要があると考えられる．また，実験結果から，PageRank が HITS よりも性能が良く，クエリによっては提案手法よりも良くなっていることが分かるが，PageRank がある程度うまくいく理由は分かっていない．よって，PageRank を詳しく解析し，PageRank の良さも取り入れた手法も考えていきたい．

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] Kleinberg, Jon M. ”Authoritative sources in a hyperlinked environment.” Journal of the ACM (JACM) 46.5 (1999): 604-632.
[2] 武吉朋也, et al. ”クラスタ係数に基づく HITS アルゴリズムの特性解析と改善.” 北海道大学大学院情報科学研究科 DEWS2006 論文集 (2006).
[3] 津田宏治, et al. ”共起行列の固有ベクトルを用いる単語クラスタリング法 文書データベースの概要を表す単語クラスタの抽出 .” 情報処理学会研究報告自然言語処理 (NL) 1994.77 (1994-NL-103) (1994): 41-48.
[4] Weng, Jianshu, et al. ”Twitterrank: finding topic-sensitive influential twitterers.” Proceedings of the third ACM international conference on Web search and data mining. 2010.
[5] Brin, Sergey, and Lawrence Page. ”The anatomy of a large-scale hypertextual web search engine.” Computer networks and ISDN systems 30.1-7 (1998): 107-117.
[6] Jeh, Glen, and Jennifer Widom. ”Simrank: a measure of structural-context similarity.” Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. 2002.
[7] Scarselli, Franco, et al. ”The graph neural network model.” IEEE transactions on neural networks 20.1 (2008): 61-80.
[8] Kipf, Thomas N., and Max Welling. ”Semi-supervised classification with graph convolutional networks.” arXiv preprint arXiv:1609.02907 (2016).
[9] Yuan, Hao, et al. ”Explainability in graph neural networks: A taxonomic survey.” IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).
[10] Garcia, Victor, and Joan Bruna. ”Few-shot learning with graph neural networks.” arXiv preprint arXiv:1711.04043 (2017).
[11] Manning, Christopher D. Introduction to information retrieval. Syngress Publishing,, 2008.