

局所計算に基づく ObjectRank 推定

坂倉 悠太[†] 山口 祐人[†] 天笠 俊之^{††,†††} 北川 博之^{††}

[†] 筑波大学大学院システム情報工学研究科 〒 305-8573 茨城県つくば市天王台 1-1-1

^{††} 筑波大学システム情報系情報工学域 〒 305-8573 茨城県つくば市天王台 1-1-1

^{†††} 宇宙航空研究開発機構宇宙科学研究所 〒 252-5210 神奈川県相模原市中央区由野台 3-1-1

E-mail: [†]{yuuta,yutoymgc}@kde.cs.tsukuba.ac.jp, ^{††}{amagasa,kitagawa}@cs.tsukuba.ac.jp

あらまし データベース上のオブジェクトの重要性を評価する手法として、ObjectRank というリンク構造解析手法がある。ObjectRank は行列計算を繰り返すため、計算コストが大きい。一方で、全てのノードではなく、一部のノードの ObjectRank だけが必要とされる場合もある。本稿では、グラフ全体ではなく局所的なグラフを用いることで、計算コストを抑えつつ ObjectRank を推定する手法を提案する。提案手法では、エッジの重みを考慮し、ObjectRank の推定に適した局所的なグラフを作成する。評価実験では、提案手法は、比較手法と比較して、計算コストを抑えながら精度を維持することができた。

キーワード ObjectRank, リンク構造解析, データマイニング, 情報検索

Yuta SAKAKURA[†], Yuto YAMAGUCHI[†], Toshiyuki AMAGASA^{††,†††}, and Hiroyuki
KITAGAWA^{††}

[†] Graduate School of Systems and Information Engineering, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan

^{††} Faculty of Engineering, Information and Systems, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan

^{†††} Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency
3-1-1 Yoshinodai, Chuo-ku, Sagamihara, Kanagawa, 252-5210, Japan

E-mail: [†]{yuuta,yutoymgc}@kde.cs.tsukuba.ac.jp, ^{††}{amagasa,kitagawa}@cs.tsukuba.ac.jp

1. はじめに

論文の共著関係やソーシャルネットワークなど、様々な種類のノードから構成されるグラフが多く存在する。このようなグラフに関する研究は多く行われているが[6], [8], [16], 例えば良い論文を多く書いている著者のような、重要なノードを抽出するという問題がある。

複数種類のノードやエッジから構成されるグラフから重要なノードを抽出する手法として、ObjectRank [2] という手法がある。ObjectRank は PageRank [7] を拡張したリンク構造解析手法である。PageRank とは異なり、ObjectRank のエッジには種類ごとに重みが付けられている。

ObjectRank は、PageRank と同様に、行列計算を繰り返すため計算コストが大きい。一方で、全てのノードではなく、一部のノードの重要度だけが必要とされる場合がある。例えば、特定のノードの重要度の時間的な推移を知りたい場合、各時刻

におけるグラフ全体に対して ObjectRank を計算すると、計算コストが膨大になってしまう。また、ある特定のノード群のランキングを作成したい場合には、注目しているノード以外の重要度は不要である。このような場合、注目しているノードの重要度だけを高速に推定できると理想的である。

PageRank においては、特定のノードの PageRank を推定する手法が提案されている[4]。しかし、PageRank とは異なり、ObjectRank は複数種類のノードとエッジを持つグラフを扱っているため、エッジには重みが付けられている。この特徴により、PageRank における手法をそのまま ObjectRank に適用することは不十分である。そのため、エッジの重みを考慮して重要度を推定する。

本稿では、グラフ全体ではなく局所的なグラフを用いて、ObjectRank を推定する手法を提案する。ObjectRank において、重みが小さいエッジをたどって遷移する評価値は小さいため、そのエッジが持つ推定の精度への影響は小さい。提案手法

では、エッジの重みを考慮し、重要度の推定に適した局所的なグラフを作成する。本稿では、この局所的なグラフのことを局所グラフと呼ぶ。本研究の目的は、推定の精度を維持したまま、計算コストを抑えることである。精度の指標として、推定した重要度の精度と重要度を推定したノードのランキングの精度に注目する。計算コストの評価指標には、作成された局所グラフのサイズと、グラフデータを保存しているデータベースへのアクセス回数を用いる。

本研究の貢献を以下に示す。

- 複数種類のノードやエッジを持つグラフに対するノードの評価値を推定する手法の提案
 - DBLP^(注1)データセットを用いた提案手法の有効性の検証
- DBLP データセットを用いて評価実験を行った。提案手法は、既存手法と比較して、局所グラフサイズが約 39% 削減され、推定した重要度の精度が約 24%、ランキングの精度が約 6% 向上した。
- 本稿の構成は以下の通りである。2 節で関連研究を紹介し、3 節で予備知識を説明する。4 節で提案手法について説明する。5 節で評価実験を行い、6 節でまとめと今後の課題を述べる。

2. 関連研究

複数種類のノードを含むグラフを用いた様々な研究がされている。Neubauer ら [6] は、ソーシャルブックマークにおけるユーザ、Web ページ、タグをノードで表現したグラフを分析し、スパマー分類器を作成した。Sun ら [8] は、著者と会議をノードで表現したグラフを用いた、著者のランキング手法を提案した。山口ら [16] は、Twitter^(注2)におけるユーザと投稿されたメッセージをノードで表現したグラフに対して、ObjectRank を適用し、ユーザのランキングを行った。本研究では、DBLP データセットを、著者、論文、出版年、出版場所をノードで表現したグラフを用いる（詳細は 3.1 節で述べる）。

また、あるグラフの部分グラフ内のノードの PageRank に注目した研究も行われている。Davis ら [5] は、推定したい部分グラフ内のノードからエッジを張られているノードを部分グラフに追加し、元の部分グラフを含む部分グラフを作成した。この作成した部分グラフ内で PageRank を計算することで、ノードの PageRank を推定した。Xue ら [15] は、ユーザのアクセスパターンを用いて、推定したい Web グラフから新しいグラフを作成し、そのグラフを用いて PageRank を推定した。Wu ら [14] は、推定したい部分グラフに含まれていないノードを一つのノードで表現し、このノードと推定したい部分グラフから構成される部分グラフを用いて PageRank を推定した。Vattani ら [13] は、任意のノード集合が与えられた時に、それらのノードの Personalized PageRank を保存する誘導部分グラフを作成する手法を提案した。これらの研究は、部分グラフ内のノードの評価値に注目している点が、本研究とは異なる。

Chen ら [4] は、局所グラフを用いて PageRank を推定する

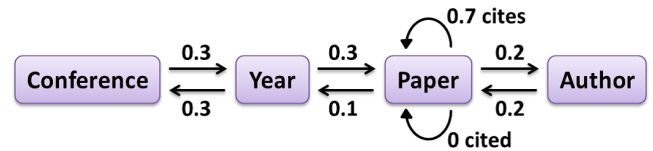


図 1 Schema Graph の例

手法を提案した。Chen らの手法は、評価値の推定に適した局所グラフを作成し、その局所グラフを用いて評価値を推定する。また、過去にクロールした Web グラフを用いて、新しくクロールした Web グラフにおける評価値を推定する手法も提案した。詳細は 3.2 節で述べる。Chen らの手法は、PageRank を推定しているため、局所グラフを作成する時にエッジの重みを考慮していない点が、本研究と異なる。

Broder ら [3] は、グラフ凝集による PageRank の推定手法を提案した。Broder らの手法は、Web ページをホストごとに凝集したホストグラフを作成し、ホスト間における計算とホスト内の計算によって、PageRank を推定した。本研究との違いは、特定のノードの評価値だけではなくグラフ全体の評価値を推定している点である。

3. 予備知識

3.1 ObjectRank

ObjectRank [2] は PageRank [7] を拡張したリンク構造解析手法であり、データベース上のオブジェクトの重要度を評価するためのアルゴリズムである。PageRank とは異なり、ObjectRank では複数種類のノードとエッジを扱うことができる。ObjectRank では、ノードとエッジにラベルを付与することで、ノードとエッジの種類を区別する。

まず、Authority Transfer Schema Graph と呼ばれる、グラフ構造とエッジの重みを定義するグラフを定義する（以下、Schema Graph と呼ぶ）。図 1 に Schema Graph の例を示す。Schema Graph は、ノードラベルの集合（図 1 における “Conference”, “Year”, “Paper”, “Author”）とノードラベル間の関係を表すエッジの集合から構成される。ノードラベル間に存在する各エッジには、エッジラベルと重みが付与されている。また、評価値は互いに遷移すべきという考え方に基づいて、ノードラベル間には双方向のエッジが存在している。エッジの重みは、そのエッジをたどって遷移する評価値の割合を表している。例えば、図 1 の例では、Paper ラベルノードの評価値は、1 割が Year ラベルノードへ、2 割が Author ラベルノードへ、7 割が他の Paper ラベルノードへ遷移する。エッジの重みは、ノードラベル間の関係を反映するように手動で設定される。この時、各ノードから出るエッジの重みはそれぞれ 0 以上、エッジの重みの総和は 1 以下になるように設定しなければならない。

次に、Schema Graph に基づいて、リンク構造解析の対象のグラフである Authority Transfer Data Graph を構築する（以下、Data Graph と呼ぶ）。図 2 に Data Graph の例を示す。Data Graph におけるノード v_i からノード v_j へのエッジ e_{ij}

(注1): <http://www.informatik.uni-trier.de/~ley/db/>

(注2): <http://twitter.com/>

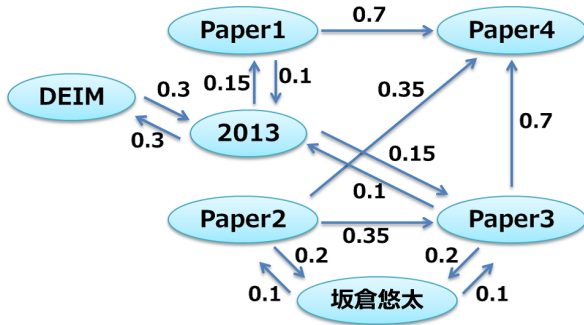


図 2 Data Graph の例

の重み w_{ij} は、エッジ e_{ij} のエッジラベルに付与されている重みを、 v_i におけるそのエッジラベルの出次数で割った値になる。例えば、図 2 の例では、Year ラベルノードである “2013” からは、Paper ラベルノードである “Paper1” と “Paper3” へのエッジが張られている。図 1 より、Year ラベルノードから Paper ラベルノードへのエッジの重みは 0.3 である。Year ラベルノードである “2013” は、Paper ラベルノードに対して 2 本のエッジを張っている。そのため、ノード “2013” がノード “Paper1” とノード “Paper3” に対して張っているエッジの重みは、それぞれ 0.15 になる。

ObjectRank は以下のように定式化される。

$$\mathbf{r} = d\mathbf{A}\mathbf{r} + \frac{(1-d)}{N}\mathbf{u}$$

ここで、 N はノードの総数、 $\mathbf{r}$ は全ノードの評価値ベクトル、 d はエッジをたどって評価値が遷移する確率（ダンピングファクター）、 $\mathbf{A}$ は遷移確率行列、 $\mathbf{u}$ は要素が全て 1 の N 次元ベクトルである。

評価値を計算するために、以下の式による計算を繰り返す。

$$\mathbf{r}^{(k+1)} \leftarrow d\mathbf{A}\mathbf{r}^{(k)} + \frac{(1-d)}{N}\mathbf{u} \quad (\leftarrow \text{は代入を意味する})$$

ここで、 $\mathbf{r}^{(k)}$ は k 回目の計算によって得られた評価値ベクトル $\mathbf{r}$ である。評価値ベクトル $\mathbf{r}$ が収束するまで、上式による計算を繰り返す。

また、この式は PageRank と同様であるが、ObjectRank は遷移確率行列 $\mathbf{A}$ の作成の仕方が異なる。ObjectRank では、ノード v_i から v_j へのエッジが存在する場合、 a_{ji} には Data Graph におけるエッジの重み w_{ij} が格納され、それ以外の場合は 0 が格納される。

3.2 Chen らの PageRank 推定手法

Chen ら [4] の手法は、局所グラフを用いて PageRank を推定する手法である。Chen らの手法では、評価値の推定に適した局所グラフを作成する。作成した局所グラフを用いて評価値を推定したいノード（対象ノード）の評価値を推定する。

まず、対象ノードからエッジを逆方向にたどっていくことで部分グラフを拡大し、局所グラフを作成する。拡大の始点となる部分グラフは、対象ノードだけを含む部分グラフである。まず、対象ノードについて注目し、対象ノードから出ているエッジとそのエッジ先のノード、対象ノードに入っているエッジと

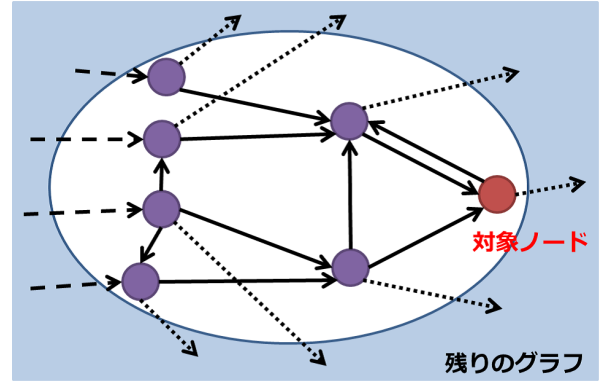


図 3 作成された局所グラフの例

そのエッジ元のノードの情報を取得する。その中で、対象ノードに張られているエッジとそのエッジ元のノードを部分グラフに追加する。あるノードに張られているエッジとそのエッジ元のノードを部分グラフに追加する操作のことをノードの展開と呼ぶ。次に、ノードの展開によって部分グラフに新しく追加されたノードについて注目し、それらのノードも同様に展開する。ある展開の基準を満たさなくなるまで再帰的にノードの展開を繰り返すことで、部分グラフを拡大し、局所グラフを作成する。作成された局所グラフの例を図 3 に示す。

局所グラフは三種類のエッジとノードを含んでいる。一つ目のエッジは入力エッジと呼ばれる局所グラフ外から局所グラフ内へ入ってくるエッジ、二つ目のエッジは出力エッジと呼ばれる局所グラフ内から局所グラフ外へ出ていくエッジ、三つ目は、内部エッジと呼ばれる局所グラフ内から局所グラフ内へのエッジである。また、入力エッジを張られているノードを境界ノードと呼ぶ。対象ノードと境界ノードを除く、局所グラフ内のノードを内部ノードと呼ぶ。図 3 の局所グラフは、7 本の出力エッジ（点線）と、4 本の入力エッジ（破線）、10 本の内部エッジ（実線）を持つ。また、対象ノードは 1 個、内部ノードは 2 個、境界ノードは 4 個である。

Chen らは、効率的に部分グラフを拡大するために、ノードの影響力を考慮した。あるノードの影響力とは、そのノードの持つ評価値のうち、ランダムジャンプを除いて、最終的に対象ノードに到達する評価値の割合のことである。影響力の大きいノードは、対象ノードの評価値の推定精度への影響が大きい。また、影響力が大きいノードに対してエッジを張っているノードも、大きな影響力を持つと考えられる。Chen らの手法では、影響力を入次数で割った値を展開の基準として用い、この値が閾値以上であるノードを展開する。

影響力の正確な計算は計算コストが大きいため、Chen らは OPIC アプローチ [1] を用いて、以下の手順で影響力の推定値を計算した。

(1) 影響力の推定値を計算したいノード v に評価値 1 を与える。

(2) ノード v の評価値を、ノード v が張っているエッジとランダムジャンプによって遷移させる。この時、出力エッジによって部分グラフ外に遷移する評価値とランダムジャンプに

よって遷移する評価値は無視する。

(3) 評価値を持っている内部ノードと境界ノードから一つのノードを選択し、そのノードの評価値を(2)と同様に遷移させる。

(4) 内部ノードと境界ノードが持っている評価値の合計が閾値未満になるまで(3)の操作を繰り返す。

(5) 対象ノードが持っている評価値をノード v の影響力の推定値とする。

局所グラフの作成後、境界ノードの評価値を推定する。境界ノードは局所グラフ内からだけではなく、局所グラフ外からもエッジ(入力エッジ)を張られている。そのため、境界ノードの評価値を直接計算することはできず、その評価値を推定する必要がある。また、対象ノードの評価値の推定精度は、境界ノードの影響力と境界ノードの評価値の推定精度に影響される。そのため、境界ノードの評価値を精度よく推定することが重要である。Chen らは、ヒューリスティックな方法を用いて、境界ノードの評価値を推定した。具体的には、境界ノードに張られている入力エッジから評価値 d/E が遷移してくると推定した。ここで、 E はグラフ全体におけるエッジの総数である。ランダムジャンプと内部ノード、対象ノードから境界ノードに遷移してくる評価値は計算することができる。これにより、境界ノードの評価値を推定することができる。

最後に、局所グラフ内で PageRank を計算することで対象ノードの評価値を計算する。PageRank の計算の繰り返しごとに以下の操作を行う。

- 推定した評価値 (d/E) が入力エッジから境界ノードに遷移する。
- ランダムジャンプと出力エッジによって局所グラフ外に遷移する評価値を無視する。

PageRank の計算終了後の対象ノードの評価値を、推定した対象ノードの評価値とする。

4. 提案手法

本節では、まず提案手法の概要について説明する。その後、3種類の提案手法の詳細について説明する。提案した手法の中で、最も精度が良かった手法は提案手法1であった(詳細は5.節で述べる)。

4.1 概要

重みが小さいエッジをたどって遷移する評価値は小さいため、そのエッジが持つ対象ノードの評価値の推定精度への影響は小さい。そこで、エッジの重みを考慮しながら対象ノードから部分グラフを拡大し、局所グラフを作成する。その後、局所グラフの境界ノードの評価値を推定する。最後に、局所グラフを用いて ObjectRank を計算し、対象ノードの評価値を計算する。

Chen らの手法は PageRank を対象にしているため、全てのエッジを平等に扱っている。しかし、ObjectRank で扱うエッジは複数種類あり、重み付きエッジである。そのため、提案手法では、エッジの重みを考慮して部分グラフを拡大する。例えば、図2において、ノード“2013”は、ノード“DEIM”と“Paper1”からエッジを張られている。これらのエッジは重み

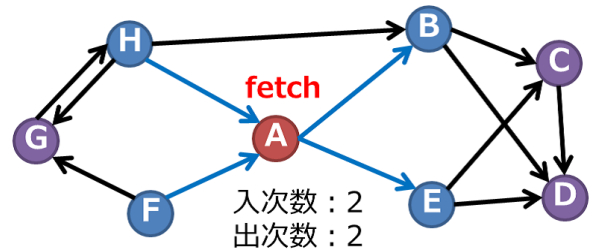


図4 fetch 操作の例

が異なるため、その重みの違いを考慮する。

本稿では、グラフデータがリレーショナルデータベースなどに保存されている環境を想定する。そのため、ノードの情報を取得するために、“fetch”操作を用いる。fetch 操作では、一つのノードの ID が与えられた時、以下の情報をデータベースから取得する。

- 与えられたノードの入次数と出次数
- 与えられたノードが張っているエッジとそのエッジのラベル、そのエッジ先のノードの ID
- 与えられたノードに張られているエッジとそのエッジのラベル、そのエッジ元のノードの ID

fetch 操作の例を図4に示す。図4において、ノード A に fetch 操作を行った時、ノード B, E, F, H, これらのノードとノード A をつなぐエッジ、これらのエッジのラベルを取得できる。また、ノード A の入次数と出次数を取得することができる。5.節の評価実験では、部分グラフの拡大の時に行った fetch 操作の回数を計算コストの指標として用いる。

4.2 提案手法1: Data Graph におけるエッジの重みを考慮した部分グラフの拡大手法

提案手法1は、Data Graph におけるエッジの重みを考慮して部分グラフを拡大する。重みが小さいエッジが持つ対象ノードの評価値の推定精度への影響は小さい。そこで、提案手法1では、境界ノードを展開する時に、境界ノードに張られている全てのエッジとエッジ元のノードを部分グラフに追加するのではなく、Data Graph におけるエッジの重みを考慮し、追加するエッジとノードを選択する。これにより、次に展開する境界ノードを減らし、fetch 操作の回数を減らすことができる。そのため、計算コストを小さくすることができる。

提案手法1では、影響力を入次数で割った値が閾値以上である境界ノードに張られているエッジのうち、Data Graph における重みが閾値以上であるエッジとエッジ元のノードだけを部分グラフに追加する。図5に例を示す。この例では、ノード“2012”とノード“Paper1”は境界ノードである。影響力を入次数で割った値の閾値として0.1、Data Graph におけるエッジの重みの閾値として0.02が設定されている。まず、それぞれのノードの影響力の推定値を計算すると、ノード“2012”の影響力は0.4、ノード“Paper1”の影響力は0.33であった。影響力を入次数で割った値を計算すると、ノード“2012”は0.08、ノード“Paper1”は0.11である。閾値は0.1であるため、ノード“2012”は展開しない。次に、ノード“Paper1”に張られて

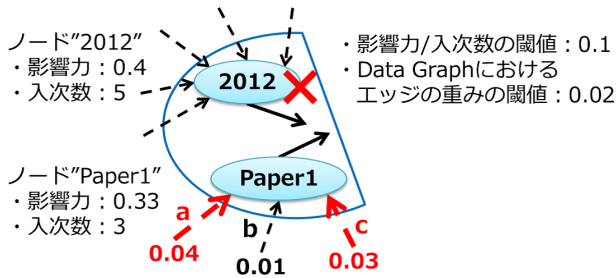


図5 部分グラフの拡大例1

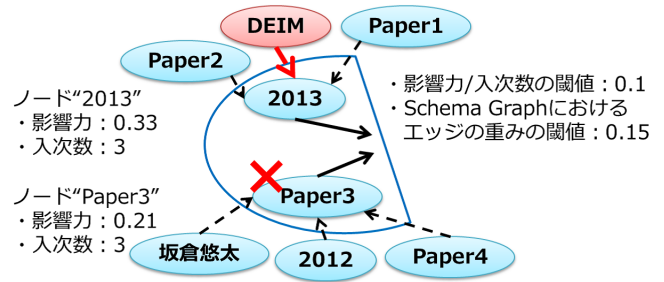


図6 部分グラフの拡大例2

いるエッジ a, b, c の Data Graph における重みを計算すると、順に 0.04, 0.01, 0.03 であった。エッジの重みの閾値は 0.02 であるため、部分グラフに追加されるのはエッジ a とエッジ c、これらのエッジ元のノードだけである。

4.3 提案手法2：Schema Graph におけるエッジの重みを考慮した部分グラフの拡大手法

提案手法2は、Schema Graph におけるエッジの重みに対して閾値を設定し、追加するエッジのラベルを選択する。例えば、図1の Schema Graph に対して、エッジの重みの閾値として 0.15 が設定されたとする。この時、Paper ラベルノードから Year ラベルノードへのエッジと、cited ラベルエッジの重みは閾値未満であるため、これらのラベルのエッジは部分グラフに追加されない。他の6種類のラベルのエッジだけが追加される。

提案手法1は Data Graph におけるエッジの重みを考慮するのに対して、提案手法2は Schema Graph におけるエッジの重みを考慮して部分グラフを拡大する。提案手法1において、Data Graph におけるエッジの重みを計算するためには、実際に境界ノードを展開する前に、部分グラフ外にある、境界ノードにエッジを張っているノードに対して fetch 操作を行う必要がある。一方、提案手法2では、Schema Graph とエッジの重みに対する閾値が決定されると、追加するエッジのラベルが決定される。境界ノードに対して fetch 操作を行えば、境界ノードに張られているエッジのラベルを取得することができる。そのため、提案手法2は、提案手法1と比較して、fetch 操作の回数を小さくすることができる。

図6に例を示す。この例では、図1の Schema Graph を用いており、ノード“2013”とノード“Paper3”は境界ノードである。影響力を入次数で割った値の閾値として 0.1、Schema Graph におけるエッジの重みの閾値として 0.15 が設定されている。まず、それぞれのノードの影響力の推定値を計算すると、ノード“2013”の影響力は 0.33、ノード“Paper3”の影響力は 0.21 であった。影響力を入次数で割った値を計算すると、ノード“2013”は 0.11、ノード“Paper3”は 0.07 である。閾値は 0.1 であるため、ノード“Paper3”は展開しない。次に、ノード“2013”に張られているエッジのラベルから、Schema Graph におけるエッジの重みを取得する。Conference ラベルノードから Year ラベルノードへのエッジの重みは 0.3、Paper ラベルノードから Year ラベルノードへのエッジの重みは 0.1 である。エッジの重みの閾値は 0.15 であるため、部分グラフに追加され

るのは、Conference ラベルノード“DEIM”とノード“DEIM”が張っているエッジだけである。

4.4 提案手法3：Schema Graph におけるエッジの重みを考慮した確率的な部分グラフの拡大手法

Schema Graph におけるエッジの重みが大きいほど、対象ノードの評価値の推定精度に与える影響は大きいと考えられる。そのため、提案手法3では、Schema Graph におけるエッジの重みが大きいほど、大きな確率をエッジに与え、その確率に基づいて部分グラフを拡大する。

提案手法2と提案手法3は Schema Graph におけるエッジの重みを考慮している。提案手法2はエッジの重みに対して閾値が設定されると、重みが閾値未満であるエッジラベルを全て無視してしまう。一方、提案手法3では、エッジの重みに基づいた確率を与え、その確率に基づいてエッジとエッジ元のノードを部分グラフに追加する。そのため、必ず無視されるエッジラベルはなく、全てのエッジラベルが追加される可能性を持っている。

提案手法3は、提案手法2と同様の手法である。ただし、Schema Graph におけるエッジの重みに基づいて、エッジとエッジ元のノードを確率的に部分グラフに追加する。図7に例を示す。この例は、4.3節と同様に、図1の Schema Graph を用いている。ノード“2013”とノード“Paper3”は境界ノードであり、影響力を入次数で割った値の閾値として 0.1 を設定している。まず、影響力を入次数で割った値について判定すると、閾値未満であるため、ノード“Paper3”は展開しない。次に、ノード“2013”に張られている各エッジのラベルから、Schema Graph におけるエッジの重みを取得する。Schema Graph におけるエッジの重みを、そのエッジとエッジ元のノードを追加する確率として与える。この例では、0.3の確率でノード“DEIM”とノード“DEIM”から張られているエッジを、0.1の確率でノード“Paper1”とノード“Paper2”、これらのノードから張られているエッジを部分グラフに追加する。

5. 評価実験

本節では、まず実験で用いるデータセットについて説明する。次に、提案手法と比較する手法について説明し、実験の方法を述べる。その後、評価指標について述べ、実験結果を示し、結果に対する考察を述べる。

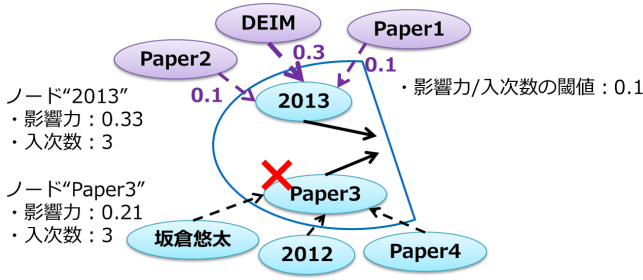


図 7 部分グラフの拡大例 3

表 1 Data Graph の詳細

Paper ラベルノード数	1,572,278
Author ラベルノード数	916,973
Conference ラベルノード数	6,711
Year ラベルノード数	76
Conference-Year 間のエッジ数	29,427
Year-Paper 間のエッジ数	1,572,134
cites/cited ラベルエッジ数	2,083,947
Paper-Author 間のエッジ数	4,121,353

5.1 データセット

本実験では、Arnetminer^(注3)が提供している DBLP のデータセット [9] ~ [12] を用いた。このデータセットには、文献ごとのタイトル、著者、出版場所、出版年、引用している文献、引用されている数が含まれている。図 1 の Schema Graph と上記のデータセットに基づいて Data Graph を作成した。作成された Data Graph の詳細を表 1^(注4)に示す。

5.2 比較手法

本実験では、3 種類の提案手法を Chen らの手法および次の二つの手法と比較した。

比較手法 1 は、対象ノードから k ステップだけエッジを逆方向にたどることで部分グラフを拡大する。すなわち、対象ノードまでの距離が k 以下である全てのノードを含む局所グラフを作成する。例えば、図 3 のグラフは $k = 2$ の時に作成される局所グラフの例である。

比較手法 2 は、提案手法 1 と同様の部分グラフ拡大手法である。ただし、Data Graph におけるエッジの重みを計算する時に、Schema Graph におけるエッジの重みを使用しない。具体的には、部分グラフ外にあるノード v から境界ノードに張られているエッジの重みを $1/\text{outdegree}(v)$ とする。ここで、 $\text{outdegree}(v)$ はノード v の出次数である。

5.3 実験方法

まず、グラフ全体に対して ObjectRank を計算し、各ノードの正確な評価値を求めた。次に、評価値を推定する対象ノードを選択するために、5.1 節で示したデータセットから対象ノードをサンプリングした。本実験では、Author ラベルノードの上位 100 件を選択した。そして、提案手法と比較手法を用いて、

サンプリングした対象ノードの評価値を推定した。Chen らの手法を参考にして、入力エッジから評価値 d/E が遷移してくるとした。また、ダンピングファクター d の値を 0.85 に設定した。

各手法の閾値を以下のように設定して、評価実験を行った。提案手法 1 と比較手法 2 は、以下の閾値の組合せ (15 通り) で評価実験を行った。

- ・ 影響力/入次数: 0.1, 0.01, 0.001
- ・ エッジの重み (Data Graph): 0.1, 0.01, 0.001, 0.0001, 0.00001

提案手法 2 は、以下の閾値の組合せ (6 通り) で評価実験を行った。

- ・ 影響力/入次数: 0.1, 0.01, 0.001
- ・ エッジの重み (Schema Graph): 0.1, 0.2

提案手法 3 と Chen らの手法は、影響力を入次数で割った値の閾値を 0.1, 0.01, 0.001 に設定して評価実験を行い、比較手法 1 は、ステップ数 k を 1, 2 に設定して評価実験を行った。

5.4 評価指標

本実験では、推定の精度と計算コストの観点から手法を評価した。

各手法の評価値の推定精度の評価指標として、相対誤差を用いた。対象ノード t の評価値の相対誤差 $\text{err}(t)$ は以下のように定義される。

$$\text{err}(t) = \frac{|r(t) - r'(t)|}{r(t)}$$

ここで、 $r(t)$ は対象ノード t の正確な評価値、 $r'(t)$ は対象ノード t の推定した評価値である。相対誤差 $\text{err}(t)$ の値が小さいほど、対象ノード t の評価値を高い精度で推定できていることを示している。サンプリングした対象ノードの相対誤差の平均を計算して、各手法を評価した。

サンプリングした対象ノードのランキングの精度を評価するために、スピアマンの順位相関係数を用いた。スピアマンの順位相関係数 ρ は以下のように定義される。

$$\rho = 1 - \frac{6\sum D^2}{N^3 - N}$$

ここで、 D は正確な評価値に基づくサンプリングした 100 個の対象ノードのランキングと推定した評価値に基づくサンプリングした 100 個の対象ノードのランキングにおける同一ノードの順位差、 N はノードの数 (本実験では 100) である。順位相関係数 ρ の値が 1 に近いほど、推定した評価値に基づくランキングは精度よく推定できていることを示している。

計算コストを評価するために、fetch 操作の回数と作成された局所グラフのサイズを用いた。大規模なグラフデータはリレーショナルデータベースなどに保存されているため、fetch 操作を行うにはデータベースにアクセスする必要がある。fetch 操作の回数を減らすことで、データベースへのアクセスを減らせるため、部分グラフの拡大における計算コストを小さくすることができる。また、局所グラフのサイズが小さいほど、評価値の推定の時に作成する遷移確率行列が小さくなる。そのため、

(注3): <http://arnetminer.org/>

(注4): データセットに出版年が不明な文献が含まれていたため、作成された Data Graph において、“Paper ラベルノード数”と“Year-Paper 間のエッジ数”が一致していない。

局所グラフのサイズが小さいほど、評価値の推定における計算コストを小さくすることができる。

5.5 実験結果

手法ごとの順位相関係数と平均 fetch 回数および平均局所グラフサイズの比較結果を、それぞれ図 8 と図 9 に示す。提案手法 1、提案手法 2、比較手法 2 については、順位相関係数が最大の時におけるエッジの重みの閾値の結果を用いて比較した。凡例の括弧の中の値はエッジの重みの閾値である。

また、手法ごとの平均相対誤差と平均 fetch 回数の比較結果を図 10 に示す。相対誤差は、サンプリングした 100 個のノードの正確な評価値と推定した評価値について、それぞれの総和が 1 になるように正規化してから計算した。

図 8 と図 10 より、3 種類の提案手法の中では、提案手法 1 の順位相関係数が最も大きく、平均相対誤差が最も小さいことが分かった。また、図 9 より、提案手法 1 は順位相関係数を維持しながら、平均局所グラフサイズを小さくできることが分かった。提案手法 2 では、Schema Graph のエッジラベルを考慮しているため、エッジの重みの閾値が決定すると部分グラフに追加するエッジラベルが決定される。提案手法 3 では、Schema Graph のエッジラベルに基づいて、確率的に部分グラフを拡大している。一方、提案手法 1 では、Data Graph におけるエッジの重みを考慮しており、重みが閾値以上であるエッジだけを部分グラフに追加する。そのため、閾値を適切に設定することで、評価値の推定に適した局所グラフを作成することができたと考えられる。

また、提案手法 1 は、比較手法 1 と比較手法 2 と比較して、順位相関係数の大きさと平均 fetch 回数が同程度であるが、平均局所グラフサイズが小さいことが分かった。提案手法 1 は、Data Graph におけるエッジの重みが閾値以上のエッジだけを追加していくことで部分グラフを拡大していく。すなわち、この結果は、提案手法 1 は評価値の推定に不要なエッジとノードを追加せずに、局所グラフを作成することができることを示していると考えられる。比較手法 2 も Data Graph におけるエッジの重みを考慮しているが、出次数の逆数をエッジの重みとして用いているだけで、エッジラベルを考慮していない。そのため、正確なエッジの重みを考慮できず、評価値の推定に不要なノードとエッジも部分グラフに追加してしまったと考えられる。

さらに、提案手法 1 は、Chen らの手法と比較して、ほぼ同じ平均 fetch 回数であるが、順位相関係数と平均相対誤差、平均局所グラフの点で優れていた。Chen らの手法は、影響力を入次数で割った値が閾値以上である境界ノードにエッジを張っている全てのノードを部分グラフに追加する。一方、提案手法 1 は、Data Graph におけるエッジの重みを考慮し、評価値の推定に不要なエッジとノードを部分グラフに追加せずに、局所グラフを作成できる。そのため、提案手法 1 の局所グラフサイズは、Chen らの手法よりも小さくなったと考えられる。また、提案手法 1 は、Chen らの手法より局所グラフサイズが小さいにもかかわらず精度が向上していた。これより、境界ノードの展開時に全てのエッジを追加するのではなく、追加するエッジを適切に選択することが重要であると考えられる。

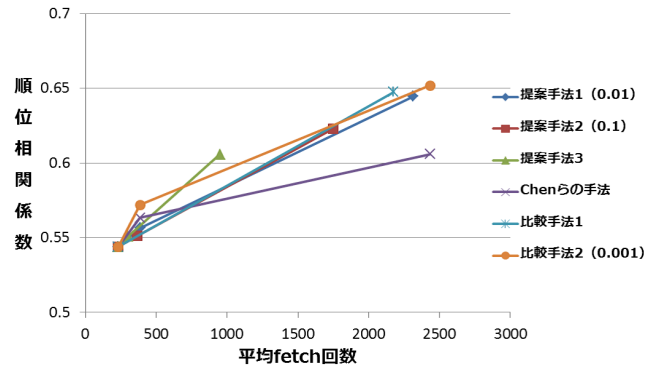


図 8 順位相関係数と平均 fetch 回数の比較

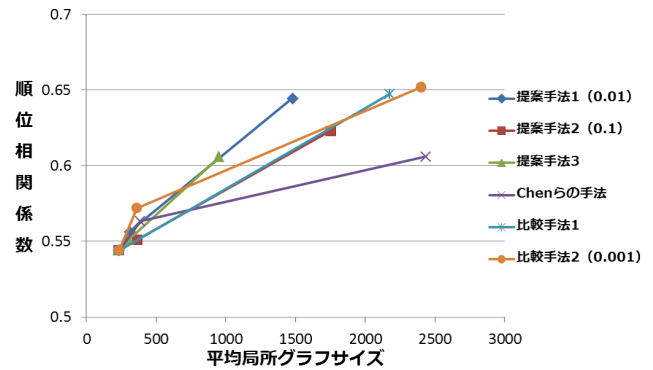


図 9 順位相関係数と平均局所グラフサイズの比較

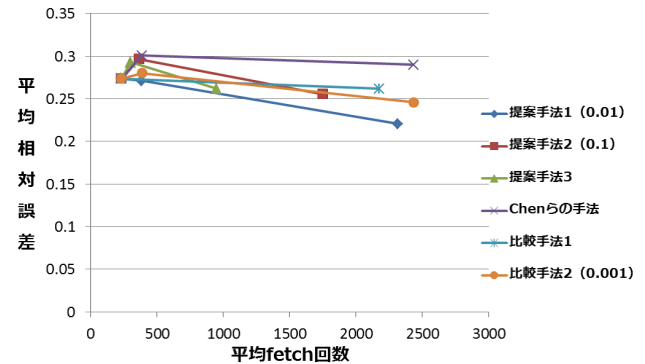


図 10 平均相対誤差と平均 fetch 回数の比較

6. まとめと今後の課題

本稿では、局所グラフを用いた ObjectRank の推定手法を提案した。提案手法では、対象ノードから、エッジの重みを考慮して部分グラフを拡大し、局所グラフを作成した。作成した局所グラフを用いて ObjectRank を計算することで、対象ノードの評価値を推定した。評価実験では、提案手法は比較手法と比較して、計算コストを抑えつつ、精度を維持することができた。

今後の課題として、境界ノードの評価値の推定方法の改善が挙げられる。今回の評価実験では、Chen らの手法を参考にした方法で評価値を推定した。そのため、ObjectRank に適した評価値の推定手法を提案することで、推定の精度を向上させることができると考えられる。

謝辞 本研究の一部は、科学研究費補助金・若手 B(23700102)による。

文 献

- [1] Serge Abiteboul, Mihai Preda, and Gregory Cobena. Adaptive on-line page importance computation. In *World Wide Web Conference Series*, pages 280–290, 2003.
- [2] Andrey Balmin, Vagelis Hristidis, and Yannis Papakonstantinou. ObjectRank: Authority-Based Keyword Search in Databases. In *Very Large Data Bases*, pages 564–575, 2004.
- [3] Andrei Z. Broder, Ronny Lempel, Farzin Maghoul, and Jan O. Pedersen. Efficient PageRank approximation via graph aggregation. *Information Retrieval*, 9:123–138, 2006.
- [4] Yen-Yu Chen, Qingqing Gan, and Torsten Suel. Local methods for estimating pagerank values. In *International Conference on Information and Knowledge Management*, pages 381–389, 2004.
- [5] Jason V. Davis and Inderjit S. Dhillon. Estimating the global pagerank of web communities. In *Knowledge Discovery and Data Mining*, pages 116–125, 2006.
- [6] Nicolas Neubauer, Robert Wetzker, and Klaus Obermayer. Tag spam creates large non-giant connected components. In *Adversarial Information Retrieval on the Web*, pages 49–52, 2009.
- [7] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.
- [8] Yizhou Sun, Jiawei Han, Peixiang Zhao, Zhijun Yin, Hong Cheng, and Tianyi Wu. RankClus: integrating clustering with ranking for heterogeneous information network analysis. In *Extending Database Technology*, pages 565–576, 2009.
- [9] Jie Tang, Limin Yao, Duo Zhang, and Jing Zhang. A combination approach to web user profiling. *ACM TKDD*, 5(1):1–44, 2010.
- [10] Jie Tang, Duo Zhang, and Limin Yao. Social network extraction of academic researchers. In *ICDM’07*, pages 292–301, 2007.
- [11] Jie Tang, Jing Zhang, Ruoming Jin, Zi Yang, Keke Cai, Li Zhang, and Zhong Su. Topic level expertise search over heterogeneous networks. *Machine Learning Journal*, 82(2):211–237, 2011.
- [12] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: Extraction and mining of academic social networks. In *KDD’08*, pages 990–998, 2008.
- [13] Andrea Vattani, Deepayan Chakrabarti, and Maxim Gurevich. Preserving personalized pagerank in subgraphs. In *International Conference on Machine Learning*, pages 793–800, 2011.
- [14] Yao Wu and Louiqa Raschid. ApproxRank: Estimating Rank for a Subgraph,. In *International Conference on Data Engineering*, pages 54–65, 2009.
- [15] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Wei-Ying Ma, Hong-Jiang Zhang, and Chao-Jun Lu. Implicit link analysis for small web search. In *Research and Development in Information Retrieval*, pages 56–63, 2003.
- [16] 山口祐人, 天笠俊之, 高橋翼, 北川博之. 情報伝搬を考慮したグラフ分析による Twitter ユーザランキング手法. 情報処理学会論文誌データベース, 4:142–157, 2011.