6L-04

大規模グラフに対する逐次的なノードの枝刈りを用いたRankClusの高速化

山崎 耕太郎 † 塩川 浩昭‡ 北川 博之‡

† 筑波大学情報学群情報科学類 ‡ 筑波大学計算科学研究センター

1 はじめに

グラフ分析手法のひとつである RankClus [1]はランキングとクラスタリングを統合し，相互に補完することで従来の手法より正確で効率的な分析を可能にした手法である．しかし，RankClus はグラフに含まれるクラスタの数だけ部分グラフを作成し，全ての部分グラフに対してランキングとクラスタリングを実行する必要があり，大規模グラフにおいて計算時間が膨大となってしまうという問題がある．

そこで本稿では大規模グラフに対する RankClus の高速化手法を提案する．提案手法では，計算対象となるノード数を削減することで RankClus の高速化を図る．具体的には，処理の過程において計算されるランク値の変化に着目し，ランク値の変化率が著しく低いノードを検出し逐次的に計算対象から枝刈りすることでノードの数を削減する．本稿では，人工データを用いた提案手法の有効性を検証し，ノード間のリンク接続密度が低いグラフに対して，提案手法が効果的であることを確認した．

2 RankClus

2.1 入力データと初期化

RankClus [1] は，Bi-Type Information Network と呼ばれるデータモデルを対象とする．2 つのノード集合，$X=\{x_1,x_2,\dots,x_m\}, Y=\{y_1,y_2,\dots,y_n\}$が与えられた時，$V$ と E はそれぞれグラフのノード集合とエッジ集合とし，$V(G)=X\cup Y$，$E(G)=\{\langle o_i,o_j\rangle\}$ (ただし，$o_i,o_j\in X\cup Y$)とするグラフ$G(V,E)$を Bi-Type Information Network と定義する．m,nをそれぞれX,Yの要素の数とする時，各データ間のリンクの重みを$W_{(m+n)\times(m+n)}=\{w_{o_i,o_j}\}$の隣接行列で表す．また便宜的にリンクの行列 W を$W_{XX}, W_{XY}, W_{YX}, W_{YY}$ の 4 つのブロックに分解する．各ブロックの添字が型間のオブジェクトのサブネットワークを示す．

RankClus ではX,Yの中から，target type と attribute type を定める．RankClus は target type であるノード集合に対してのみクラスタリングを行い，attribute type として指定されたノード集合はランキングとクラスタリング処理の補助情報として利用する．

2.2 ランキング処理

RankClus は target type に対し任意のK個の初期クラスタを与え，各データ集合に対してクラスタに基づいたランキング処理をする．

ランキング関数は Authority Ranking [3] と呼ばれるノード集合 X，Y のオーソリティを考慮したランキング関数を用いる．ノード集合X,Yに含まれる各ノードのランク値 $\vec{r}_X,\vec{r}_Y$を次の式で定義する．

$$\vec{r}_X=\frac{W_{XY}\vec{r}_Y}{|\vec{r}_X|},\vec{r}_Y=\frac{\alpha W_{YX}\vec{r}_X+(1-\alpha)W_{YY}\vec{r}_Y}{|\vec{r}_Y|}$$

ただし，$0<\alpha<1$とする．$\vec{r}_X,\vec{r}_Y$は互いの値を入力として持つ再帰構造となっており，収束した$\vec{r}_X,\vec{r}_Y$を獲得するまで，$\vec{r}_X,\vec{r}_Y$を反復計算する必要がある．

2.3 クラスタリング処理

ランキング関数によって得たランク値に基づき，target type の各ノードに対して各クラスタへの寄与率を求める．ここではノード集合Xを target type とする．寄与率を計算するにあたって EM アルゴリズム[2]を適応する．混合分布モデルを扱う上で，あるクラスタkに対するノード集合 Y のランク値を$\mathrm{p}_k(Y)$と表し，クラスタkに対するXのランク値を$p_k(X)$と表す．また，クラスタkにXの要素x_iが属する事後確率，$\pi_{ik}=p(k|x_i)$とするとき，x_iとYに含まれる各yの間にリンクが生成される確率$p_{x_i}(Y)$は以下の式でモデル化する．

$$p_{x_i}(Y)=\sum_{k=1}^{K}\pi_{i,k}p_k(Y),\text{ただし}\sum_{k=1}^{K}\pi_{i,k}=1$$

π_{ik}のパラメータ推定にあたりリンク生成の尤度を EM アルゴリズムにより計算し，最終的に以下の式を得る．$p(k)$はグラフの総リンク数に対するクラスタkの内部に含まれるリンク数の割合である．

$$\pi_{i,k}=\frac{p(z=k|x_i)=p_k(x_i)p(z=k)}{\sum_{l-1}^{K}p_l(x_i)p(z=l)}$$

π_{ik}をk次元ベクトル$\vec{s}_{x_i}=(\pi_{1k},\pi_{2k},\dots,\pi_{ik})$で表しクラスタの中心を求める．これは各クラスタ内の全ての要素x_iに対するs_{x_i}の平均から求める．最後に各 target type に対し求めたクラスタの中心に一番近いクラスタに割り当て処理を終了する．

2.4 RankClus の問題点

RankClus は与えられた全てのデータに対して，上述のランキング処理，EM アルゴリズムによる計算およびクラスタリング処理を繰り返し行う．特にランキング処理は全てのノードに対する反復計算を必要とし，データ数の増加に伴い計算時間も増加する．

3 提案手法

本稿では RankClus の高速化手法を提案する．我々は RankClus の出力するランク値は反復計算の

Fast RankClus for Large Graphs via Incremental Node Pruning

Kotaro Yamazaki †, Hiroaki Shiokawa ‡ and Hiroyuki Kitagawa ‡

† College of Information Science, University of Tsukuba

‡ Center for Computational Sciences, University of Tsukuba

進行とともに次第に収束していく性質に着目した．そこで本研究では，ランク値の収束傾向を利用して，ランキング処理における計算対象のノード数を削減することにより RankClus の高速化を図る．

具体的には，提案手法は反復計算の過程において，ランク値の変化が著しく低いノードを特定し，計算対象から逐次的に枝刈りする．これにより，ランキング処理における計算時間の削減を図る．

枝刈り可能なノードを特定するために，ランキング処理の各反復計算において各ノードの前の反復と比べたときのランク値の変化率を求める．変化率は現在のノードのランク値とひとつ前の反復のランク値の変化の割合とし，以下の式で定義する．

[定義 1] (変化率)

t 番目の繰り返し計算におけるクラスタkに対するノード j のランク値$\vec{r}_Y$の変化率$R^{(t)}(j,k)$ は次のように計算する．

$$\boldsymbol{R}^{(t)}(j,k) = \frac{{r_Y}^{(t)}(j,k)}{{r_Y}^{(t-1)}(j,k)}$$

変化率がパラメータλ_I回の反復において連続して$R^{(t)}(j,k) \leq \lambda_R$となるノードを枝刈り対象とする．パラメータ$\lambda_I, \lambda_R$はユーザが任意に与えるものとする．

4 評価実験

提案手法の実行速度および精度を人工データセットにより評価する．本実験で用いる人工データデータセットはノード集合Xのノード数 45，ノード集合Yのノード数 2000 とする Bi-Type Information Network であり，ノード集合 Xを target type とする．ノード集合 Xは 3 つのクラスタに分割され，各クラスタ内のリンクの総数$P = [P_1, P_2, P_3]$（リンク密度）と各クラスタ間でリンクが生成される確率を指定する確率遷移行列$T = [T_{11}, T_{12}, T_{13}; T_{21}, T_{22}, T_{23}; T_{31}, T_{32}, T_{33}]$（クラスタ間の分離度）の値を変えた 5 種類のデータセットを構築した．ただし，P_iはクラスタi内のリンク数，T_{ij}はクラスタiからjへリンクが張られる確率とする．

表 1:データセットの詳細

データセット	リンクの総数P	確率遷移行列T
Dataset1 (中密度,中分離)	[1000,1500,2000]	[0.8, 0.05, 0.15; 0.1, 0.8, 0.1; 0.1, 0.05, 0.85]
Dataset2 (低密度,中分離)	[800,1300,1200]	[0.8, 0.05, 0.15; 0.1, 0.8, 0.1; 0.1, 0.05, 0.85]
Dataset3 (高密度,中分離)	[2000,3000,4000]	[0.8, 0.05, 0.15; 0.1, 0.8, 0.1; 0.1, 0.05, 0.85]
Dataset4 (中密度,大分離)	[1000,1500,2000]	[0.9, 0.05, 0.05; 0.05, 0.9, 0.05; 0.1, 0.05, 0.85]
Dataset5 (低密度,大分離)	[800,1300,1200]	[0.9, 0.05, 0.05; 0.05, 0.9, 0.05; 0.1, 0.05, 0.85]

4.2 評価指標

クラスタリングの評価指標には NMI (Normalized Mutual Information)を用いた．NMI はクラスタリングの結果と正解データを比較する．結果は 0 から 1 の値を取り，1 に近いほどクラスタリング精度は高く，0 に近いほどクラスタリング精度が低いことを示す．提案手法の RankClus に対する近似精度の評価指標には NMI の相対誤差を用いる．相対誤差は$\frac{((\text{提案手法の}NMI)-(\text{RankClus の}NMI))}{(\text{RankClus の}NMI)}$と定義する．

4.3 実験結果

実行時間を図 1 に，相対誤差を表 2 に示す．

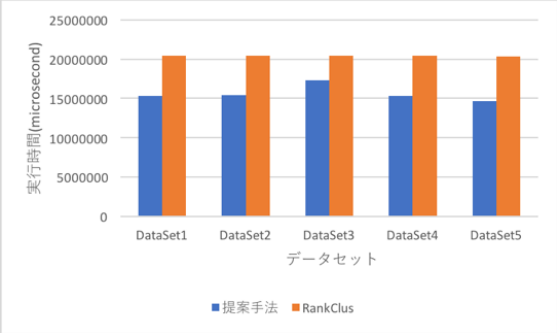

図 1: 異なるデータセットに対する実行時間

表 2: RankClus との相対誤差

データセット	相対誤差
Dataset1	0.053
Dataset2	0.026
Dataset3	0.057
Dataset4	0.126
Dataset5	0.077

実験結果より，どのデータセットに対しても提案手法は RankClus より 30%程度高速であることが確認できる．また，クラスタのリンク分離とリンク密度が低いデータセットほど誤差が小さくなる傾向にあることがわかる．以上より提案手法はリンクが適度な分離であり，密度が低いネットワークで特に有効性が高いことがわかった．

5 まとめ

本稿では大規模グラフを対象としたときの RankClus の高速化手法を提案した．提案手法では，ノードのランク値変化率が継続的に低いノードを逐次的に枝刈りすることによってノード数を削減する．提案手法は RankClus と比較して，高速かつ高精度で評価実験を行えることを確認した．特に，提案手法ではクラスタ間の接続が比較的疎であり，リンクの密度が低いネットワークに対して最も有効である．

謝辞

本研究は，JSPS 科研費(16H06650)の助成を受けたものである．

参考文献

[1] Yizhou Sun et al., RankClus: Integrating Clustering with Ranking for Heterogeneous Information Network Analysis, In Proc. EDBT ’09

[2] Jeff Bilmes. A gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, Technical report, 1998.

[3] Jon M. Kleinberg, Authoritative Sources in a Hyperlinked Environment, In Proc. ACM 1999.