ネットワーク上の頂点間特徴量としての Top- k 距離とその高速なクエリ応答

Fast Top- k Distance Retrieval for Network Structural Proximity Features

秋葉 拓哉
Takuya Akiba

国立情報学研究所, JST さきがけ
National Institute of Informatics & JST PRESTO
takiba@nii.ac.jp

林 孝紀
Takanori Hayashi

東京大学
The University of Tokyo
thayashi@is.s.u-tokyo.ac.jp

則 のぞみ
Nozomi Nori

京都大学
Kyoto University
nozomi@ml.ist.i.kyoto-u.ac.jp

岩田 陽一
Yoichi Iwata

東京大学
The University of Tokyo
y.iwata@is.s.u-tokyo.ac.jp

吉田 悠一
Yuichi Yoshida

国立情報学研究所
National Institute of Informatics
yyoshida@nii.ac.jp

keywords: graph, social network, top- k distance, proximity measure, link prediction

Summary

Estimating the relevance or proximity between vertices in a network is a fundamental building block of network analysis and is useful in a wide range of important applications such as network-aware searches and network structure prediction. In this paper, we (1) propose to use *top- k shortest-path distance* as a relevance measure, and (2) design an efficient indexing scheme for answering top- k distance queries. Although many indexing methods have been developed for standard (top-1) distance queries, no methods can be directly applied to top- k distance. Therefore, we develop a new framework for top- k distance queries based on *2-hop cover* and then present an efficient indexing algorithm based on the recently proposed *pruned landmark labeling* scheme. The scalability, efficiency and robustness of our method are demonstrated in extensive experimental results. It can construct indices from large graphs comprising millions of vertices and tens of millions of edges within a reasonable running time. Having obtained the indices, we can compute the top- k distances within a few microseconds, six orders of magnitude faster than existing methods, which require a few seconds to compute these distances. Moreover, we demonstrate the usefulness of top- k distance as a relevance measure by applying them to link prediction, the most fundamental problem in graph data mining. We emphasize that the proposed indexing method enables the first use of top- k distance for such tasks.

1. はじめに

最短経路の長さとして定義される頂点間の距離は、グラフにおける最も基礎的な概念の1つであり、様々な幅広い応用を持つ。例えば、距離は頂点間の関連性を示すと考えられるため、ソーシャルネットワークを用いユーザとの関連性を考慮する情報検索 [Vieira 07, Yahia 08, Maniu 13] や、ウェブグラフを用い現在閲覧中のページとの関連性を考慮する情報検索 [Vieira 07, Yahia 08, Maniu 13, Ukkonen 08, Potamias 09] において用いられる。

しかし、距離を関連度の指標として用いることには問題点がある。ソーシャルグラフやウェブグラフは通常重み無しのグラフであるため、距離は整数値となる。

また、それらを含む非常に幅広い種類のグラフがいわゆるスモールワールド現象を示し、小さい直径を持っている [Watts 98]。従って、距離が取り得る値の種類は実際にはかなり少なく、グラフの構造を表現する力に欠けてしまう。

また、距離は最も短い1本の経路の長さのみに依存してしまうという欠点もある。例として、図1に示される3つの異なるグラフを考えてみよう。各グラフにおいて赤く塗られた2頂点間の距離は全て5である。従って、距離を関連度の指標として用いると、これらの頂点对の関連度は全く同じということになってしまう。しかし、直感的には図1(c)の頂点对は図1(a), (b)のものに対しより強い関連度を持っていると感じられ

図 1: 2 頂点の接続関係の例.

表 1: 図 1 に示された頂点对の距離と Top- k 距離.

グラフ	(Top-1) 距離	Top- k 距離
(a)	5	[5, 7, 7, 7, 7, 7, 9, 9, ...]
(b)	5	[5, 5, 5, 5, 7, 7, 7, 7, ...]
(c)	5	[5, 5, 5, 5, 5, 5, 5, 5, ...]

る. これは, 図 1 (c) の 2 点間により多くの短い経路が存在するためである.

本論文ではまず 1 つめの貢献として, より高い表現力を持つ頂点間関連度の指標として **Top- k 距離** を用いることを提案する. **Top- k 距離** は複数回同じ頂点を通ることが許された 2 点間の経路の中で最も短い k 本の距離であり, バイオインフォマティクスなどの分野では使用されてきたものの [Eppstein 98], ソーシャルネットワークの解析には用いられてこなかった. **Top- k 距離** の正確な定義は 3 章で行う. 表 1 は図 1 に示した 3 つの頂点对の **Top- k 距離** を表している. **Top- k 距離** は 3 つの頂点对について確かに異なっており, **Top- k 距離** がこのような構造の違いを捉えることができることがわかる.

既存の関連度の指標は, 関連度の強さを 1 つの数で表現する. 一方, **Top- k 距離** は頂点間の構造を k 個の数で表現する. 従って, 既存の関連度指標の代替として即座に利用できない場面も存在するが, かえってこのような表現がより有利となる応用も多数存在する. 例えば, 機械学習に基づいた分類器等と組み合わせる場合, k 個の数の特徴量として用いることができ, 問題ごとの目的に応じた調整を分類器に任せることができる.

Top- k 距離 を計算するアルゴリズムは, 索引付けを伴わないもののみ, 先行研究が存在する. **Top- k 距離** を計算する素朴な方法は, 同一の頂点を高々 k 回訪れるように Dijkstra のアルゴリズムを拡張して得られる. n, m をグラフの頂点数及び辺数とした時, 計算量は重み無しグラフで $O((n+m)k)$ 時間, 及び重み有りグラフで $O((n \log n + m)k)$ 時間である. 一方, Eppstein のアルゴリズム [Eppstein 98] は重み無しグラフでは $O(n + m + k)$ 時間, 重み有りグラフでは $O(n \log n + m + k)$ 時間と, 理論的な計算量の改善を達成しているものの,

本論文 5 章の実験結果でも確認されるように, 実用的な実行速度は改善していない.

これらの手法のこれまでの応用先は交通グラフにおける経路計画やバイオインフォマティクスにおけるアラインメント等である. 我々の知る限り, **Top- k 距離** をソーシャルネットワークやウェブグラフの解析に用いている先行研究は存在しない. 従って, **Top- k 距離** を頂点間の関連性の特徴量とみなす考え方自体の提案が本論文の 1 つめの貢献である.

一方, **Top- k 距離** を本論文で対象とするようなネットワーク解析の応用に適用するためには, 計算時間が課題となる. 例えば, ネットワークを考慮した情報検索やネットワーク構造予測のような応用においては, 評価や訓練のため, **Top- k 距離** を非常に多くの頂点对に対し計算する必要がある. 従って, 上述のような, 1 つの頂点对に対する計算時間がグラフの大きさに比例して大きくなってしまふ既存の手法を大規模なグラフに適用した場合, 時間がかかり過ぎてしまう. 特に, ネットワークを考慮した情報検索のようなユーザとの相互作用を含むタスクでは, 検索結果を瞬時に提示したい一方, 検索結果のランク付けのために多くの頂点对に対する関連度を得る必要があり, 1 つの頂点对の **Top- k 距離** を非常に高速に計算する必要がある. 例えば, 5 章における最大のデータにおける実験結果では, 上述の既存手法は 1 つの頂点对について共に 5 秒以上の計算時間を要してしまう. 仮に 100 個の対をランキングして検索結果を提示したい場合, 少なくとも 500 秒の間ユーザを待たせることになってしまい, 非現実的である.

そこで本論文では, 2 つめの貢献として, **Top- k 距離** を効率的に計算するための索引付けの手法を提案する. 上で述べた **Top- k 距離** を計算する既存の手法は, 前計算を一切伴わない手法である. 一方, 提案手法は索引付け手法であるため, 与えられたグラフからまず索引を構築し, その索引を用いて 2 点間の **Top- k 距離** を計算する. 我々の知る限り, 提案手法は **Top- k 距離** に対する最初の索引付け手法である.

全ての結果を前計算するような自明な手法は, グラフが大規模な場合, 明らかに適用できない. 例えば, 5 章の実験で用いた最大サイズのグラフは約 3×10^6 個の頂点を持つ. $k = 8$ とした場合, 全頂点对に対する結果は, 無向グラフと仮定しても約 3.6×10^{13} 個の数となる.

提案手法は通常の 2 点間距離に対する索引付け手法である **枝刈りラベリング法** [Akiba 13] に基づいている. **Top- k 距離** への拡張のためには, ユニークな経路の数を新たに考慮する必要が生じる. そのため, 新たなアイデアに基づく同じ経路を 2 度数えないための工夫を提案する. 正当性の証明もより非自明なものとなっている. また応答性能やスケーラビリティの低下を防

ぐための高速化手法も複数提案する。

提案手法の性能は、実際のグラフを用いた比較実験により検証した。提案手法は高いスケーラビリティを持ち、数千万辺から成る大規模なグラフから索引を構築することができた。また、構築した索引を用いると、Top- k 距離の計算を平均数十マイクロ秒で行うことができ、これは既存の手法に対し最大で百万倍程度高速であった。

最後に、Top- k 距離を頂点間関連度指標として用いるアプローチ自体の有用性を実証するため、グラフに関するデータマイニングにおける最も基礎的な問題の 1 つであるリンク予測問題 [Liben-Nowell 03] を取り上げ、実験を行った。実験では、Top- k 距離を特徴量としてサポートベクターマシン (SVM) を用いることにより、複数の基礎的な比較手法に対し優位な結果を出すことができることが示された。ここで取り上げたリンク予測問題は頂点間関連度を用いる問題の 1 つの例であり、この結果は Top- k 距離の様々な応用における有用性を示唆するものと考えられる。また、このような応用で Top- k 距離を用いることは、上述の索引付け手法の提案により初めて可能になったということを強調したい。

構成. 本論文の構成は以下の通りである。2 章で距離の索引付けや頂点間関連度に関する関連研究を述べる。3 章では前提となる概念を説明する。4 章で Top- k 距離のための索引付け手法を提案する。5 章で索引付け手法の評価実験の結果を示す。6 章ではリンク予測問題への適用を通じて Top- k 距離の有用性を示す。7 章で結論を述べる。

2. 関連研究

2.1 距離に対する索引付け

通常の頂点間距離に対する索引付けの手法は数多くが提案されてきた。主要なアプローチの 1 つは、木分解に基づく手法 [Wei 10, Akiba 12] である。これは、グラフに対して木分解 [Robertson 84] をヒューリスティック的に計算し、各バッグに対する距離行列を保存するものである。2-Hop Cover に基づく手法 [Cohen 02, Cheng 09, Jin 12, Fu 13, Akiba 13] もよく研究されてきた。これは、前処理で各頂点にラベルと呼ばれる配列を計算し、クエリ時にはラベルのみから距離を計算する手法である。また、近似手法として、ランダムマークに基づく手法 [Potamias 09, Gubichev 10, Qiao 12] も提案され改善が続けられてきた。これは、数個の頂点をランダムマークとして選択して、全長頂点とランダムマークとの距離を保存することにより、三角不等式を用いて距離を推定する手法である。このように、通常の頂点間距離に対する索引付けには数多くの既存研究が存在する

ものの、Top- k 距離の応答に直接用いることができる手法は存在していない。

提案する索引付け手法は枝刈りラベリング法 [Akiba 13] に基づく。枝刈りラベリング法は、距離に対する索引付け手法の 1 つであり、2-Hop Cover [Cohen 02] に基づく厳密手法である。枝刈りラベリング法は、既存の厳密手法に対し大幅にスケーラビリティを改善している一方、巧妙な枝刈りが核となっており、アルゴリズムは全体としてより簡潔になっている。この簡潔さ及び枝刈りのアイデアの汎用性により、到達可能性への特化 [Yano 13]、交通グラフ上の最短経路への特化 [Akiba 14a]、動的グラフにおける動的更新や時系列を加味したクエリへの対応 [Akiba 14b] など、これまでも様々な拡張が行われてきた。

2.2 頂点間の関連度の指標

距離以外にも、関連度の指標として既に提案され利用されているものがある。その 1 つとして、*Random Walk with Restart (RWR)* [Haveliwala 02] が挙げられる。RWR は多くの応用において関連度の指標としての有用性が実証されてきているが、計算時間が問題となってしまう応用も少なくない。

反復法等に基づく厳密計算は大規模なグラフではかなりの実行時間がかかってしまう [Maehara 14]。精度を犠牲にした近似手法も提案されているが、上述した多数の頂点対に対する計算を要する応用を考えた場合、やはり十分な速度と言えない [Jeh 03, McSherry 05, Sun 05, Tong 06]。一方、近年提案された索引付けを伴う手法は、指定された 2 頂点間の RWR 値を非常に高速に応答できる [Fujiwara 12a, Fujiwara 12b]。しかし、索引付けが LU 分解や QR 分解のような行列分解手法に基づいているため、スケーラビリティに制限があり、百万辺程度のグラフの処理が限界となってしまう。

同様の議論が、*SimRank* [Jeh 02]、*Commuting Time* [Lovász 96] 及び *グラフカーネル* [Smola 03, Ito 05] のような他の関連度指標にも成立している。一方、提案する Top- k 距離の索引付け手法は、5 章の実験結果が示す通り、高いスケーラビリティと応答性能を兼ね備える。また、特徴量として用いる場合、既存の関連度指標と併せて利用することも可能であるため、こういった既存の関連度指標の存在は Top- k 距離の有用性を否定するものではない。

3. 前 準 備

3.1 表 記

本論文ではグラフ $G = (V, E)$ として表されるネットワークを扱う。 V は頂点集合、 E は辺集合を表す。説明を簡単にするため、主な議論はグラフ G を重み無しの無向グラフであると仮定して行う。ただし、辺の重

みや有向グラフへの対応も容易であり、4.6 節で議論する。

グラフ $G = (V, E)$ の頂点数 $|V|$ 及び辺数 $|E|$ をそれぞれ n, m と表す。頂点は整数などにより自然に表現されていると仮定する。従って、2 頂点 $u, v \in V$ に対しその表現を用いた全順序が定義されているものとし、頂点間で $u < v$ や $u \leq v$ といった自然な比較が行えるものとする。

経路の**内部頂点**とは経路に含まれる頂点であって両端点でないものとする。経路の集合 $\mathcal{P}$ に対し、 $\mathcal{P}$ の i 番目の**最短経路**とは、 $\mathcal{P}$ に含まれる経路を長さで昇順に整列し i 番目のものと定義する。同じ長さの経路の順序は任意とする。

頂点対 $s, t \in V$ に対し、 $\mathcal{P}_{st}$ を s から t への全ての (単純とは限らない) 経路の集合とする。頂点 $v \in V$ に対し、 $\mathcal{P}_{st}^{>v}$ を $\mathcal{P}_{st}$ に含まれる経路のうち全ての内部頂点が v より大きいものと定義する (ここでの頂点の比較は上で述べた頂点の表現によるものである)。同様に、 $\mathcal{P}_{st}^{\geq v}$ を $\mathcal{P}_{st}$ に含まれる経路であって少なくとも 1 つの内部頂点が v 以下であるものと定義する。 s, t 間の i 番目の**最短経路**とは $\mathcal{P}_{st}$ の i 番目の最短経路とする。また、 $d_{i\text{-th}}(s, t)$, $d_{i\text{-th}}^{>v}(s, t)$, 及び $d_{i\text{-th}}^{\geq v}(s, t)$ はそれぞれ $\mathcal{P}_{st}$, $\mathcal{P}_{st}^{>v}$, 及び $\mathcal{P}_{st}^{\geq v}$ の i 番目の最短経路の長さを表す。対応する集合の大きさが i 未満である場合、それらは ∞ と定義される。同様に $d_{i\text{-th}}^{\geq v}(s, t)$, $d_{i\text{-th}}^{\geq v}(s, t)$ も定義する。

3.2 問題設定

本論文では、以下のように定義される Top- k 距離クエリ問題に対するグラフ索引付け手法を扱う。

問題 (Top- k 距離クエリ) :

Given: 頂点対 $s, t \in V$.

Answer: 列 $(d_{1\text{st}}(s, t), d_{2\text{nd}}(s, t), \dots, d_{k\text{-th}}(s, t))$.

提案手法は索引付け手法であり、1 つのグラフに対して、索引構築は 1 回だけ行い、構築した索引を用いて質問応答を繰り返し行う。即ち、処理は以下のように索引構築と質問応答の 2 つのステップからなる。

(1) **索引構築** — グラフ G 及び正整数 k から索引を構築する。

(2) **質問応答** — 頂点の組 u, v を受け取り Top- k 距離クエリに回答する。

単純な Top- k 距離と一般の Top- k 距離. 一部の既存アルゴリズムは単純な Top- k 経路、即ち閉路を含まない Top- k 経路を計算する [Yen 71, Gao 10]。一方、上述の通り、本論文の問題設定では Top- k 経路に閉路を含むことを許している。これは、以下の理由による。

(1) 6 章の実験で示されるように、閉路を許した Top- k 距離であっても、一部の応用においては有用な

指標となっている。

(2) 閉路を許さない Top- k 距離の計算は一般の Top- k 距離よりも難しく、それ故に、閉路を許さない Top- k 距離の計算を大規模なグラフで行うことは困難であると思われる。例えば、有向グラフにおいて、一般の Top- k 距離と単純な Top- k 距離を計算する既存アルゴリズムの計算量はそれぞれ $O(n + m + k)$ 時間 [Eppstein 98] 及び $O((n + m)nk)$ 時間 [Yen 71] となっている*1。

4. 提案手法

この章では、Top- k 距離クエリに効率的に応答するための提案手法を説明する。まず、データ構造とアルゴリズムの説明を行い、その正当性を示す。次に、実際の性能を向上するための方法や辺の重み及び有向グラフに対応するための方法について議論する。

4.1 データ構造

まずは索引の枠組み、即ち索引のデータ構造と応答のアルゴリズムを提案する。Top- k 距離クエリに対する枠組みとして、通常の距離クエリに対する枠組みである 2-Hop Cover [Cohen 02] に基づいた新たな枠組みを設計する。

Top- k 距離クエリに対する枠組みを設計する上での主な課題は、同じ経路を 2 度数えないための工夫である。通常の距離クエリでは距離のみを扱い経路の本数を考慮する必要が無いのに対し、Top- k 距離クエリにおいては距離だけでなく経路の本数も新たに考慮する必要がある。従って、索引の枠組みもより進んだものが必要となる。

提案する枠組みにおける索引のデータ構造は、各頂点につき、以下の 2 種類のラベルを保存するものである。

- **距離ラベル** $L(v)$ は頂点と経路長の組 (u, δ) の集合である。ラベル $L(v)$ に含まれ頂点 u と関連付いている組が ℓ 個ある時、それらの距離を並べると、 $(d_{1\text{st}}^{\geq v}(v, u), d_{2\text{nd}}^{\geq v}(v, u), \dots, d_{\ell\text{-th}}^{\geq v}(v, u))$ となるものとする。
- **閉路ラベル** $C(v)$ は k 個の整数の列 $(\delta_1, \delta_2, \dots, \delta_k)$ であり、これは $(d_{1\text{st}}^{\geq v}(v, v), d_{2\text{nd}}^{\geq v}(v, v), \dots, d_{k\text{-th}}^{\geq v}(v, v))$ と一致する。

集合 L を距離ラベルの集合 $\{L(v)\}_{v \in V}$ 、集合 C を閉路ラベルの集合 $\{C(v)\}_{v \in V}$ とし、組 $I = (L, C)$ を索引と定義する。

*1 これは重み無しの場合であり、重みがある場合、それぞれ $O(n \log n + m + k)$ 及び $O((n \log n + m)nk)$ となる。

4.2 応答アルゴリズム

索引 $I = (L, C)$ と頂点对 (s, t) が与えられた時, s, t 間の Top- k 距離は以下のように計算する. まず, 以下のように定義される多重集合 $\Delta(I, s, t)$ を計算する.

$$\Delta(I, s, t) = \{\delta_{sv} + \delta_{vv} + \delta_{vt} \mid (v, \delta_{sv}) \in L(s), \\ \delta_{vv} \in C(v), (v, \delta_{vt}) \in L(t)\}.$$

直感的には, まず s からある頂点 v へ移動し, v を含む閉路を経て, 最後に v から t へ移動する. 距離ラベルと閉路ラベルの定義より, 各 v に対して, その v を用いて得られる経路の内部頂点は, v 以上の頂点のみから成り, v を必ず含む. これは同じ経路を二度出力しないために重要な点である. 厳密な正当性の証明は 4.4 節で行う.

列 $\text{QUERY}(I, s, t)$ を多重集合 $\Delta(I, s, t)$ の小さい方から k 個の要素とする. $|\Delta(I, s, t)| < k$ である場合, 残りの要素は ∞ とする. クエリ (s, t) に対する応答として $\text{QUERY}(I, s, t)$ を回答する. 応答の時間計算量は $O(|L(s)||L(t)|k)$ となる.

4.3 索引構築アルゴリズム

提案する索引構築アルゴリズムは通常の距離クエリに対する 2-Hop 索引を計算するアルゴリズムである **枝刈りラベリング法** [Akiba 13] に基づいている. 全体のアルゴリズムは Algorithm 1 として示されている. まず閉路ラベル $C(v)$ を各頂点 v について計算し, その後で距離ラベル L を各頂点からの **枝刈り幅優先探索** により計算する.

Algorithm 1 索引構築アルゴリズム

```

1: procedure CONSTRUCTINDEX( $G$ )
2:   for  $i = 1$  to  $n$  do  $C(v_i)$  を計算.
3:    $L(v) \leftarrow \emptyset$  for all  $v \in V$ .
4:   for  $i = 1$  to  $n$  do PRUNEDBFS( $G, v_i, L$ ).
5:   return  $(C, L)$ .
```

§1 閉路ラベルの計算

閉路ラベルの計算は以下のように行う. 各頂点 v について, v 以上の頂点のみを用いて, 幅優先探索に類似した探索を行う. ある頂点の訪問は k 回まで許され, 頂点 v に k 回訪問した時点で探索を終了する. このような探索を行うと, 頂点 v への k 回の訪問に対応する経路の長さが $d_{1\text{st}}^{\geq v}(v, v), d_{2\text{nd}}^{\geq v}(v, v), \dots, d_{k\text{-th}}^{\geq v}(v, v)$ と対応する.

1 度の探索の計算量は最悪の場合 $\Theta(mk)$ 時間である. しかし, 実際のグラフでは v 周辺のごく一部の頂点のみに訪問するため, グラフの大きさにかかわらず探索は瞬時に終了する. 従って, 索引構築の時間は距離ラベルの計算が殆どを占める.

§2 距離ラベルの計算

以下では頂点集合 V が $v_1, v_2, \dots, v_n$ として順序付けられていると仮定する. このとき, 距離ラベルを計算するため, 各 i ($1 \leq i \leq n$) について, v_i からの **枝刈り幅優先探索** を行う (Algorithm 2). この探索は, 各頂点を高々 k 回訪問する素朴な幅優先探索アルゴリズムに基づいているが, 枝刈りラベリング法の拡張によって導入される枝刈りが本質的であり, 自明な手法との大きな性能の差異を生む.

Algorithm 2 頂点 v からの枝刈り幅優先探索

```

1: procedure PRUNEDBFS( $G, v, L$ )
2:    $Q \leftarrow$  組  $(v, 0)$  のみを含むキュー.
3:   while  $Q$  が空になるまで do
4:     組  $(u, \delta)$  を  $Q$  から取り出す.
5:     if  $\delta < \max(\text{QUERY}((L, C), v, u))$  then
6:       組  $(v, \delta)$  を  $L(u)$  に追加.
7:       for all  $w \in V$  s.t.  $(u, w) \in E, w > v$  do
8:         組  $(w, \delta + 1)$  を  $Q$  に追加.
```

枝刈りの判定は, 作りかけの索引にクエリを発行することによって行われる. 頂点 u を距離 δ で訪問している際, 作りかけの索引 (L, C) を用いて v, u 間の k 番目の距離 δ' を計算する. ここで得られる δ' は正しい k 番目の距離とは限らない. そこで, δ と δ' を比較し, $\delta \geq \delta'$ の場合は枝刈りを行う. 即ち, u からの枝を走査せず, 組 (v, δ) を距離ラベル $L(u)$ に追加しない. この正当性については次節で示す.

距離ラベルの計算の時間計算量を大まかに見積もる. l を距離ラベルの大きさの平均とする. 全体の探索で, $O(nl)$ 頂点を訪問する. 各頂点では約 $O(\frac{m}{n})$ 辺を走査し, 4.5 節で紹介する手法を用いると, 枝刈りのためのクエリの評価は $O(l)$ 時間で行える. 従って, 距離ラベルの計算の時間計算量は全体で $O(ml + nl^2)$ 程度であると見積もれる. これは我々の実験結果と大まかに一致している. 実験結果では, l は数百程度である.

4.4 正当性

提案手法の正当性を以下で証明する. 以下では L_i を i 番目の頂点 v_i からの枝刈り幅優先後に得られた距離ラベルとする. 各頂点 v に対し $L_0(v) = \emptyset$ とする. 索引 I_i を組 (L_i, C) とおく. 以下の補題を証明する.

補題 1. 任意の整数 i ($0 \leq i \leq n$) 及び任意の頂点对 (s, t) について, $\text{QUERY}(I_i, s, t) = (d_{1\text{st}}^{\geq v_i}(s, t), d_{2\text{nd}}^{\geq v_i}(s, t), \dots, d_{k\text{-th}}^{\geq v_i}(s, t))$.

証明. i に関する帰納法により示す. $i = 0$ の場合, $\text{QUERY}(I_i, s, t) = (\infty, \infty, \dots, \infty)$ であり主張は成立する.

ある i に対し, 任意の $i' < i$ にて主張が成立すると仮定する. このとき, $s \neq t$ である任意の頂点对 (s, t) について i で主張が成立することを示す.

既に $\text{QUERY}(I_{i-1}, s, t) = (d_{1\text{st}}^{>v_{i-1}}(s, t), d_{2\text{nd}}^{>v_{i-1}}(s, t), \dots, d_{k\text{-th}}^{>v_{i-1}}(s, t))$ が計算できる. 集合 $\mathcal{P}$ を経路 P であって (i) P は $\mathcal{P}_{st}^{>v_{i-1}}$ に含まれ, (ii) P は v_i を含み, (iii) P の長さは $d_{k\text{-th}}^{>v_{i-1}}(s, t)$ 未満であるような経路の集合とする. $\mathcal{P}'$ を $\mathcal{P}$ のうち短いものから k 個の経路の集合とする. このとき, i 番目の枝刈り幅優先探索の後で $\mathcal{P}'$ に含まれる経路の長さが計算できることを証明すれば十分である.

経路 P を $\mathcal{P}'$ に含まれる経路とする. P を 3 つの部分 P_{sv_i} , $P_{v_i v_i}$, 及び $P_{v_i t}$ に分割する. ここで, P_{sv_i} は P のうち s から v_i の最初の出現までの部分であり, $P_{v_i v_i}$ は v_i の最初の出現から v_i の最後の出現までの部分であり, $P_{v_i t}$ は v_i の最後の出現から t までの部分である. $P_{v_i v_i}$ は $\mathcal{P}_{v_i v_i}^{>v_i}$ のうち短いものから k 個のうちに必ず含まれることに注意されたい. そうでなければ, より短い k 個の経路が得られるため, $P \in \mathcal{P}'$ という仮定に矛盾する. 従って, $C(v_i)$ は $P_{v_i v_i}$ の長さを必ず含む.

v_i からの枝刈り幅優先探索が $P_{v_i t}$, P_{sv_i} の長さを正しくラベルに追加することを示す. 即ち, v_i からの枝刈り幅優先探索が経路 $P_{v_i t}$ に沿って枝刈りされずに進むことを確かめる (P_{sv_i} に関しても同様である). 探索が $P_{v_i t}$ 上のある頂点 u に距離 δ で到達した時に枝刈りされたと仮定しよう. この場合, $\mathcal{P}_{v_i u}^{>v_{i-1}}$ に δ より短い経路が少なくとも k 本含まれることになる. この k 本の各経路について, P_{sv_i} , $P_{v_i v_i}$, 及び $P_{v_i t}$ のうち u から t への部分と連結すると, $\mathcal{P}_{st}^{>v_{i-1}}$ に含まれる k 本の経路であって, P より短い, 即ち $d_{k\text{-th}}^{>v_{i-1}}(s, t)$ より短いものが得られる. これは上の条件 (iii) に矛盾する. $\square$

$i = n$ としてこの補題を用いることにより, 提案手法の正当性が証明できる.

系 1. Algorithm 1 により得られた索引を用いて任意の頂点对に対して Top- k 距離クエリが正しく回答できる.

4.5 高速化

以下では, 高い応答性能やスケーラビリティを実現するために重要なアルゴリズムやデータ構造に対する工夫を提案する.

枝刈り幅優先探索を行う頂点の順番. 枝刈りラベリング法と同じく, 頂点を枝刈り幅優先探索の根とする順番を工夫することで実世界のグラフの性質を活用し枝刈りの効果を大幅に向上し索引構築時間やラベルサイズを小さく保つことができる. ハブと呼ばれるような中心的な頂点の存在や, 周辺部分の木に近い構造が, 頂点の順序の工夫により活用できる. より詳しくは [Akiba 13] を参照されたい.

枝刈り判定の高速化. 索引構築における枝刈り幅優先探索においては, 主な時間が枝刈り判定のためのクエリの評価に費やされる. そこで, 索引構築時に評価する

クエリの形式に注目し, 枝刈り判定を高速化する. ある頂点 v からの枝刈り幅優先探索において判定したいのは, ある頂点 u 及びある数 δ に対し, 頂点 v, u 間に長さ δ 未満の経路は k 本以上あるか, ということのみである. そこで, 幅優先探索を開始する前に, $L(v)$ に含まれる各頂点 w について, $C(w)$ を用い, v から w への経路であって長さが δ' 以下のものの数を表す $c_{w, \delta'}$ を計算しておく. すると, 頂点 v, u 間に存在する長さ δ 未満の経路の本数は, $\sum_{(w, \delta', c) \in L(u)} c \cdot c_{w, \delta - \delta'}$ として求められるため, $O(|L(u)|)$ 時間で計算できる.

キューの項目の併合. 頂点 v からの幅優先探索において頂点 u に距離 δ で到達した際, 組 (u, δ) をキューに追加するのではなく, 組 (u, δ, c) をキューに追加するようにする. ここで, c は組 (u, δ) の重複度を表す. 無向グラフでは組の重複が少なくなく, このようにすることで多数の経路を同時に処理することができるようになり, 索引付けが高速化される.

ラベルの項目の併合. 前段落の高速化と同様に, ラベルの保存に際しても, 重複度を用いた表現を行う. 即ち, 距離ラベルは 3 つ組の集合, 閉路ラベルは 2 つ組の集合となる. このようにすることで, 索引サイズが小さくなる他, クエリ応答も高速化することができる.

4.6 拡張

有向グラフ. 与えられたグラフが有向グラフの場合, 各頂点 v について 2 つの距離ラベル $L_{\text{IN}}(v)$ と $L_{\text{OUT}}(v)$ を計算し用いる. ここで, $L_{\text{IN}}(v)$ は v への経路, $L_{\text{OUT}}(v)$ は v への経路を保存する. 索引付けにおいては各頂点からの枝刈り幅優先探索を順向き及び逆向きの 2 回実行することとなる. クエリ応答においては始点の L_{OUT} と終点の L_{IN} を用いる.

重み付きグラフ. 重み付きグラフを扱う場合, 枝刈り幅優先探索を枝刈り Dijkstra アルゴリズムに置き換えればよい. 即ち, Algorithm 2 で用いたキューを順位キューに置き換える. 時間計算量の見積もりは $O(ml + nl(\log n + l))$ に変化する.

5. 評価実験

この章では, 現実世界のグラフデータを用いた比較実験により, 提案した索引付け手法のスケーラビリティ, 応答性能, 頑健性等を示す.

5.1 実験方法

計算環境. 本実験には CPU が Intel Xeon X5670 (2.93 GHz), メモリが 48GB の Linux サーバを利用した. 全てのアルゴリズムは C++ で実装され, gcc 4.8.2 を用いて最適化オプション -O2 の設定でコンパイルされた.

表 2: データセットの情報, 及び既存手法と提案手法の性能比較 ($k = 8$).

名前	データセット		$ V $	$ E $	Top- k PLL (本研究)			BFS	Eppstein [Eppstein 98]
	種類				索引構築時間	索引サイズ	応答時間		
Facebook-1	ソーシャル		334	2,218	13.7 ms	178.6 KB	1.9 μ s	227.1 μ s	378.4 μ s
Last.fm	ソーシャル		1,892	12,717	125.3 ms	1.3 MB	1.7 μ s	1.6 ms	7.5 ms
GrQc	ソーシャル		5,242	14,496	152.9 ms	2.7 MB	1.6 μ s	2.2 ms	7.3 ms
HepTh	ソーシャル		9,877	25,998	631.2 ms	7.8 MB	2.2 μ s	5.5 ms	16.5 ms
CondMat	ソーシャル		23,133	186,936	3.2 s	26.4 MB	3.1 μ s	15.2 ms	158.8 ms
Facebook-2	ソーシャル		63,732	1,545,686	239.0 s	716.8 MB	15.2 μ s	117.6 ms	2.7 s
YouTube-1	ソーシャル		1,157,828	4,945,382	624.3 s	2.3 GB	5.1 μ s	1.5 s	7.0 s
YouTube-2	ソーシャル		3,238,848	18,512,606	1627.1 s	9.6 GB	3.9 μ s	5.0 s	41.1 s
NotreDame	ウェブ		325,729	1,497,134	52.3 s	617.7 MB	2.9 μ s	249.8 ms	1.7 s
Stanford	ウェブ		281,903	2,312,497	42.5 s	230.0 MB	1.7 μ s	454.9 ms	2.9 s
BerkStan	ウェブ		685,230	7,600,595	108.7 s	1.0 GB	1.9 μ s	643.3 ms	10.8 s
Indo	ウェブ		1,382,906	16,539,644	2695.3 s	6.0 GB	12.1 μ s	1.4 s	25.4 s

(a) 索引構築時間 (ソーシャルグラフ)

(b) 索引サイズ (ソーシャルグラフ)

(c) 応答時間 (ソーシャルグラフ)

(d) 索引構築時間 (ウェブグラフ)

(e) 索引サイズ (ウェブグラフ)

(f) 応答時間 (ウェブグラフ)

図 2: パラメータ k の索引構築時間, 索引サイズ及び応答時間への影響.

データセット. 本論文ではネットワークを加味した情報検索やネットワーク構造予測問題などへの応用を目指して索引付け手法の提案を行っている. 従って, 公的に入手可能な実際のソーシャルネットワークやウェブグラフを用いて実験を行った^{*2}*3*4*5*6. 各グラフのサイズや種類は表 2 にまとめられている. 全てのグラフは重み無し無向グラフとして扱った. Facebook-1,

Facebook-2 はオンラインソーシャルネットワークサービス Facebook のあるユーザ及びその友人からなる部分グラフである. Last.fm はオンラインストリーミングラジオサービス Last.fm のユーザの友人関係の部分グラフである. GrQc, HepTh, CondMat はオンラインプレプリントサーバー ArXiv の論文より生成された共著グラフである. YouTube-1, YouTube-2 は共有サービス YouTube 上のユーザの友人関係グラフである. NotreDame, Stanford, BerkStan は大学のウェブサイト限定してクロールを行って生成されたウェブグラフである. Indo はインドのウェブサイト限定してクロールを行って生成されたウェブグラフである.

*2 <http://lovro.lpt.fri.uni-lj.si/support.jsp>

*3 <http://grouplens.org/datasets/hetrec-2011/>

*4 <http://snap.stanford.edu/>

*5 <http://socialnetworks.mpi-sws.org/datasets.html>

*6 <http://law.di.unimi.it/datasets.php> [Boldi 04, Boldi 11]

既存手法. Top- k 距離クエリに対する既存の索引付け手法は存在しないため、索引付けを伴わない以下の 2 つの手法との比較を行った。

- 幅優先探索 (BFS) に基づく素朴なアルゴリズム. FIFO キューを用いグラフの探索を行い、各頂点を高々 k 回訪問する. このアルゴリズムも著者らによって実装された.
- Eppstein のアルゴリズム [Eppstein 98]. 理論的にはほぼ線形時間を達成するアルゴリズムである. Jon Graehl 氏による C++ 実装を用いた*7.

5.2 スケーラビリティ

提案手法の高いスケーラビリティが表 2 に記された索引構築時間及び索引サイズから示唆されている. 特に、数千万辺からなる大規模なソーシャルグラフとウェブグラフ (YouTube-2 と Indo) からの索引の構築を 1 時間以内に行うことができた. また、索引のサイズは 10GB 未満に収まっており、現代の標準的な計算環境で十分利用可能である.

用いた全てのデータセットにて索引は効率的に構築されているものの、索引構築時間が依存するのはグラフのサイズのみではないことが読み取れる. これは、提案手法の索引構築の効率性は枝刈りの効果に依存しており、次数分布やクラス係数といったようなグラフの性質による影響があるからである. しかし、現実世界のソーシャルネットワーク、ウェブグラフ、コンピュータネットワーク、生物情報学のネットワークなどは類似した性質を持つことが知られており、従って提案手法は幅の広いグラフに対し効率的である. 索引サイズに対しても同様の議論が成立する. 詳しくは [Akiba 13] を参照されたい.

図 2 (d), (e) はそれぞれパラメータ k の索引構築時間及び索引サイズへの影響を表している. k の影響が緩やかであることが確認できる. 特に、 k の増加に対する索引構築時間及び索引サイズの増加は、準線形、即ち線形より緩やかであるということが分かる. これは、 k を大きくしたとしても、それまでのラベルの組み合わせでより大きな k に対する経路も表現できる場合が少なくないということ、及び 4.5 節で提案した、重複した距離の表現の工夫による.

5.3 応答時間

応答時間の計測には、提案手法は 100,000 個、既存手法は 100 個のランダムな頂点对を用いた. 提案手法は数十マイクロ秒以内での応答を実現しており、既存手法に対し大幅な高速化を実現している (表 2). 特に、最も大きいデータセットである YouTube-2 においては、

提案手法の応答時間は他の手法に対し約百万倍高速である. このような高速な応答時間は、ユーザとの相互作用を持つような実時間システムにおける Top- k 距離の利用を初めて可能にすると考えられる.

今回の実験結果では、幅優先探索に基づくアルゴリズムが一貫して Eppstein のアルゴリズムより高速であった. これは、Eppstein のアルゴリズムが複雑なデータ構造の操作を伴うため、漸近的評価では隠されていた大きな定数項が存在し、実際にはこの定数項が支配的になっているためだと考えられる.

図 2 (f) はパラメータ k に対する応答時間を表している. 応答時間は k の増加に対し遅くなっていきはするものの、 $k = 64$ でも応答が十分高速であることが確認できる. k の増加に対する応答時間の増加の速度は、索引構築時間や索引サイズの増加に対してやや早いものの、やはり k に対して準線形になっていることが分かる.

5.4 分析

提案手法の効率性をより踏み込んで分析する. 対象データには GrQc を用いた.

枝刈り幅優先探索. 図 3 (a) は各頂点からの枝刈り幅優先探索で訪れた頂点の数を表している. アルゴリズムの進行に連れて、枝刈り幅優先探索が到達する頂点の数は劇的に減少していることが分かる. また、 k の値が変化しても、訪れる頂点の数が大きく変化しないことが分かる. 一方、図 3 (b) は、各枝刈り幅優先探索で追加された組の数を表している. ここでは、4.5 節で扱ったラベルの併合を加味しないよう、延べで追加された組の数を表している. 追加される組の数もアルゴリズムの進行につれ劇的に減少するが、こちらは k の値により大きく変化することが分かる.

ラベルサイズの分布. 図 3 (c) は最終的に各頂点に付与されたラベルのサイズの分布を表している. ここでは 4.5 節で扱ったラベルの併合を加味したラベルのサイズを用いた. どの頂点のラベルのサイズも 200 以下と小さく、ラベルのサイズに大きな偏りは存在しないことが分かる. 従って、応答時間は安定していると考えられる.

6. 有用性

Top- k 距離はこれまで関連度や類似度の指標として使われてきていない. そこで、この章では Top- k 距離自体がそういった特徴量・指標として有用であることを実際の問題への適用を通じて示す.

本論文では、グラフのデータマイニングにおける最も基礎的な問題の 1 つである **リンク予測問題** [Liben-Nowell 03] を取り上げる. このような応用においては、

*7 <http://www.ics.uci.edu/~eppstein/pubs/p-kpath.html>

図 3: 枝刈り幅優先探索の探索領域やラベルサイズの分布 (GrQc データ)。

表 3: リンク予測問題における Top- k 距離に基づく手法と比較手法の予測性能 (AUC) の比較. 各データセットで平均値の最も高い手法とその他の手法の各ペアについて独立に t 検定 ($p < 0.05$) を行い, 全てのペアで有意であった場合, 平均値の最も高い手法の精度を太字で示している.

データセット	CN	Jaccard	Adamic	Preferential	Combined	SVD	RWR	Top- k	Top-1
Facebook-1	0.806	0.812	0.817	0.754	0.890	0.792	0.873	0.901	0.808
Facebook-2	0.776	0.777	0.777	0.875	0.755	0.823	0.949	0.931	0.931
Last.fm	0.596	0.597	0.603	0.831	0.861	0.644	0.844	0.876	0.802
GrQc	0.658	0.658	0.658	0.709	0.793	0.791	0.802	0.824	0.799
HepTh	0.546	0.546	0.547	0.686	0.714	0.774	0.779	0.817	0.775
CondMat	0.763	0.763	0.764	0.749	0.877	0.875	0.900	0.929	0.896

訓練や評価のため Top- k 距離を非常に多くの頂点对に対し評価する必要がある. 従って, 本論文で提案した索引付け手法によって, Top- k 距離のこういった応用への適用が初めて可能になったと考えられる.

6.1 実験方法

データセット. 前章で用いたソーシャルネットワークのうち小さいものから 6 つを用いた. これは, RWR などの比較手法のスケーラビリティの制限によるものである.

設定. 各データセットについてランダムにサンプリングを行い, 60% の辺を訓練に, 残りの 40% の辺を評価に用いた. 訓練の辺集合が与えられた時に隠された評価の辺を予測するという設定でリンク予測問題を定式化した. サンプリングを 10 回行い, 各サンプルでの実験における予測性能の平均を評価した. 各ハイパーパラメータは, 1 つのサンプルをデベロップメントデータとして用いチューニングした.

予測性能の評価指標としては AUC を用いた. 評価のために, 隠された辺の数と同数の辺の存在しない頂点对をランダムにサンプリングした. AUC は, 各評価データの中で, 実際に辺が存在した頂点对について推定されたリンク強度が, 辺が存在しない頂点对について推定されたリンク強度より高い値となる確率として計算される. AUC は境界値に依存せず精度の評価が行えるためによく用いられる評価指標である.

Top- k 距離に基づく手法. Top- k 距離に基づくリンク予測問題に対する手法として, Top- k 距離をサポートベクターマシン (SVM) の特徴量とする手法を用いた. 頂点对の Top- k 距離を $(\delta_1, \delta_2, \dots, \delta_k)$ とした時, 2 点の特徴量として, i 番目の値が $1/\sqrt{\delta_i}$ となるような k 個の値を特徴量として用いた. k の値は各データセットについて $\{2^0, 2^1, \dots, 2^6\}$ からチューニングにより選択した. 線形 SVM と非線形 SVM (RBF カーネル) のどちらを用いるかについてもチューニングにより選択した.

比較手法. まず, 最も基礎的な手法として, 以下の 4 つを用いた. (1) CN (共通近傍数), (2) Jaccard, (3) Adamic, (4) Preferential (優先的選択). 各頂点对のスコアとしてこれらの指標の値を用いた. また, (5) これら 4 つの指標を特徴量として用いた SVM とも比較を行った (Combined). さらに, より進んだ手法として, (6) SVD (singular value decomposition) [Golub 96] と (7) RWR (Random Walk with Restart) も比較対象とした. 手法 SVD では, 頂点对のスコアは潜在的意味ベクトルのコサイン類似度によって計算した. SVD の次元数は $\{2^0, 2^1, \dots, 2^8\}$ よりチューニングを行った. RWR のパラメータは既存研究 [He 04, Tong 06] に従って設定を行った. また, Top- k ($k > 1$) 距離を用いる意味を検証するため, (8) 通常の距離に基づく手法 (Top-1) の精度も記載した. これは, 上記の手法で k を 1 に固定したものである.

図 4: パラメータ k の予測性能への影響 (Last.fm データ)

表 4: リンク予測問題に用いた各データセットの average clustering coefficient (ACC) 及び global clustering coefficient (GCC).

データセット	ACC	GCC
Facebook-1	0.466	0.227
Facebook-2	0.221	0.055
Last.fm	0.187	0.049
GrQc	0.534	0.362
HepTh	0.477	0.118
CondMat	0.639	0.108

6.2 実験結果

表 3 は各データセット及び各手法についての平均 AUC を示している. 各データセットで平均値の最も高い手法とその他の手法の各ペアについて独立に t 検定 ($p < 0.05$) を行い, 全てのペアで有意であった場合, 平均値の最も高い手法の精度を太字で示した.

ほぼ全てのデータセットにおいて, **Top- k** 距離に基づく手法は他の比較手法を上回った. データセット Facebook-2 においては RWR の予測性能が最も高かったが, **Top- k** 距離に基づく手法は近い性能を達成している.

また, ほぼ全てのデータセットにおいて, **Top- k** 距離に基づく手法は通常の (**Top-1**) 距離に基づく手法より良い性能を達成している. また, 図 4 にあるように予測精度はパラメータ k に対して安定していることが確認された.

どのような構造を持つグラフで提案手法が有効であるかを完全に特徴づけることは難しいが, **Top- k** 距離が高い AUC の実現に有効であるようなグラフは, クラスタ性によりある程度特徴づけられるのではないかと考えられる. 高い AUC を実現できるのは, 正例の枝 (枝が実際に張られている頂点ペア) と負例の枝 (枝が実際には張られていない頂点ペア) を良く区別できる場合に他ならない. 従って, 距離に基づいた手法が有効となるグラフは, 基本的に, 負例頂点ペア間の距離が比較的長く, 正例頂点ペアの間の距離が短いような

グラフとなる. 特に, 現実のネットワークの多くはクラスタ性を持つことが知られている. クラスタ性が高い場合, 枝が張られている頂点ペアをランダムに取ってくると, 距離 2 のパスが複数存在することになる. さらに, それらのパスに隣接する三角形の経路により, 距離 3 のパスもより多く構成される可能性が高い.

表 4 はリンク予測問題に用いた各データセットのクラスタ係数を示している. ここでは, 2 つのクラスタ係数を用いた. 一方は **global clustering coefficient** であり, 以下のように定義される.

$$C_{GCC} = \frac{\text{三角形の個数}}{\text{連結な 3 頂点の組み合わせの数}}.$$

他方は, **average clustering coefficient** であり, **local clustering coefficient** の平均として定義される. **local clustering coefficient** の定義は以下である.

$$C_v = \frac{v \text{ を含む三角形の個数} \times 2}{d_v(d_v - 1)}.$$

ここで d_v は頂点 v の次数である. **average clustering coefficient** は以下のように表せる.

$$C_{ACC} = \frac{1}{n} \sum_{v \in V} C_v.$$

表 4 より, 確かに, 提案手法が苦手とした Facebook-2 のデータセットは相対的に低いクラスタ係数を示していることが分かる. 一方で, より低いクラスタ係数を持つ Last.fm でも, 提案手法は良好な精度を実現している. どのような構造を持つグラフで距離に基づく手法が有効であるかについてのより精密な分析を行うことは, 今後の重要な課題である.

7. ま と め

本研究では, まず **Top- k** 距離クエリに対する効率的な応答を可能とする索引付け手法を提案した. 実データを用いた実験により, 提案手法は高いスケーラビリティを持ち大規模なソーシャルネットワークやウェブグラフにおいて索引の構築が可能であること, そして **Top- k** 距離クエリが非常に高速に応答できることを示した.

次に, **Top- k** 距離をグラフにおける頂点対の関連度や類似度の特徴量として用いることを提案し, その有用性を実際の問題への適用により示した. 他の既存の関連度指標は関連度を 1 つのスカラ値により表現するのに対し, **Top- k** 距離は k 個の値により関連度を表現するため表現度が高いと考えられ, 特徴ベクトルとして用い機械学習手法と組み合わせることができる. このような応用においては, 訓練や評価のため **Top- k** 距離を非常に多くの頂点対に対し計算する必要がある. 従っ

て、本論文で提案した索引付け手法によって、Top- k 距離のこういった応用への適用が初めて可能になったと考えられる。

今後の課題として、 k の値の自動的な決定という課題が考えられる。Top- k 距離の持つ情報は、 k が大きくなるほど多くなる一方、 k を大きくすると計算に必要な時間・空間が大きくなってしまふ。与えられたデータ及び利用可能な時間・空間から、適切な k の値を自動的に調整して処理を行うことができれば、より利便性を向上できる。

ネットワーク構造が変化した時の索引の再構築手法の開発も今後の課題と言える。多くのソーシャルネットワークは時間が経過するにつれてその構造を変化させていく。グラフの頂点および辺が追加・削除されるたびに索引を高速に再計算することができれば Top- k 距離の応用の幅をより広げることが期待できる。

ソフトウェア公開. 本研究で提案を行った Top- k 距離クエリに対する索引付け手法は、容易に利用可能な形で著者らのウェブページより入手可能である*⁸。提案手法が幅広く利用され、Top- k 距離のさらなる応用が探求されることを期待する。

謝 辞

本研究は、JST、さががけの支援を受けたものである。また、本研究の一部は、日本学術振興会特別研究員奨励費 (256563, 269329) の助成によるものである。また、本研究の一部は日本学術振興会科研費 (24106003) の助成によるものである。ここに記して謝意を表す。

◇ 参 考 文 献 ◇

- [Akiba 12] Akiba, T., Sommer, C., and Kawarabayashi, K.: Shortest-path queries for complex networks: Exploiting low tree-width outside the core, in *EDBT*, pp. 144–155 (2012)
- [Akiba 13] Akiba, T., Iwata, Y., and Yoshida, Y.: Fast exact shortest-path distance queries on large networks by pruned landmark labeling, in *SIGMOD*, pp. 349–360 (2013)
- [Akiba 14a] Akiba, T., Iwata, Y., Kawarabayashi, K., and Kawata, Y.: Fast shortest-path distance queries on road networks by pruned highway labeling, in *ALENEX*, pp. 147–154 (2014)
- [Akiba 14b] Akiba, T., Iwata, Y., and Yoshida, Y.: Dynamic and historical shortest-path distance queries on large evolving networks by pruned landmark labeling, in *WWW*, pp. 237–248 (2014)
- [Boldi 04] Boldi, P. and Vigna, S.: The webgraph framework I: compression techniques, in *WWW*, pp. 595–602 (2004)
- [Boldi 11] Boldi, P., Rosa, M., Santini, M., and Vigna, S.: Layered label propagation: a multiresolution coordinate-free ordering for compressing social networks, in *WWW*, pp. 587–596 (2011)
- [Cheng 09] Cheng, J. and Yu, J. X.: On-line exact shortest distance query processing, in *EDBT*, pp. 481–492 (2009)
- [Cohen 02] Cohen, E., Halperin, E., Kaplan, H., and Zwick, U.: Reachability and distance queries via 2-hop labels, in *SODA*, pp. 937–946 (2002)
- [Eppstein 98] Eppstein, D.: Finding the k shortest paths, *SIAM J. Computing*, Vol. 28, No. 2, pp. 652–673 (1998)
- [Fu 13] Fu, A. W.-C., Wu, H., Cheng, J., Chu, S., and Wong, R. C.-W.: IS-LABEL: An independent-set based labeling scheme for point-to-point distance querying on large graphs, *PVLDB*, Vol. 6, No. 6, pp. 457–468 (2013)
- [Fujiwara 12a] Fujiwara, Y., Nakatsuji, M., Onizuka, M., and Kitsuregawa, M.: Fast and exact top- k search for random walk with restart, *PVLDB*, Vol. 5, No. 5, pp. 442–453 (2012)
- [Fujiwara 12b] Fujiwara, Y., Nakatsuji, M., Yamamuro, T., Shiokawa, H., and Onizuka, M.: Efficient personalized pagerank with accuracy assurance, in *KDD*, pp. 15–23 (2012)
- [Gao 10] Gao, J., Qiu, H., Jiang, X., Wang, T., and Yang, D.: Fast top- k simple shortest paths discovery in graphs, in *CIKM*, pp. 509–518 (2010)
- [Golub 96] Golub, G. and Loan, C.: *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences (1996)
- [Gubichev 10] Gubichev, A., Bedathur, S., Seufert, S., and Weikum, G.: Fast and accurate estimation of shortest paths in large graphs, in *CIKM*, pp. 499–508 (2010)
- [Haveliwala 02] Haveliwala, T. H.: Topic-sensitive PageRank, in *WWW*, pp. 517–526 (2002)
- [He 04] He, J., Li, M., Zhang, jiang H., Tong, H., and Zhang, C.: Manifold-ranking based image retrieval, in *MM*, pp. 9–16 (2004)
- [Ito 05] Ito, T., Shimbo, M., Kudo, T., and Matsumoto, Y.: Application of kernels to link analysis, in *KDD*, pp. 586–592 (2005)
- [Jeh 02] Jeh, G. and Widom, J.: SimRank: A measure of structural-context similarity, in *KDD*, pp. 538–543 (2002)
- [Jeh 03] Jeh, G. and Widom, J.: Scaling personalized web search, in *WWW*, pp. 271–279 (2003)
- [Jin 12] Jin, R., Ruan, N., Xiang, Y., and Lee, V.: A highway-centric labeling approach for answering distance queries on large sparse graphs, in *SIGMOD*, pp. 445–456 (2012)
- [Liben-Nowell 03] Liben-Nowell, D. and Kleinberg, J.: The link prediction problem for social networks, in *CIKM*, pp. 556–559 (2003)
- [Lovász 96] Lovász, L.: Random Walks on Graphs: A Survey, in *Combinatorics, Paul Erdős is Eighty*, Vol. 2, pp. 353–398, János Bolyai Mathematical Society (1996)
- [Maehara 14] Maehara, T., Akiba, T., Iwata, Y., and Kawarabayashi, K.: Computing personalized PageRank quickly by exploiting graph structures, *PVLDB*, Vol. 7, No. 12, pp. 1023–1034 (2014)
- [Maniu 13] Maniu, S. and Cautis, B.: Network-aware search in social tagging applications: instance optimality versus efficiency, in *CIKM*, pp. 939–948 (2013)
- [McSherry 05] McSherry, F.: A uniform approach to accelerated PageRank computation, in *WWW*, pp. 575–582 (2005)
- [Potamias 09] Potamias, M., Bonchi, F., Castillo, C., and Gionis, A.: Fast shortest path distance estimation in large networks, in *CIKM*, pp. 867–876 (2009)
- [Qiao 12] Qiao, M., Cheng, H., Chang, L., and Yu, J. X.: Approximate shortest distance computing: A query-dependent local landmark scheme, in *ICDE*, pp. 462–473 (2012)
- [Robertson 84] Robertson, N. and Seymour, P. D.: Graph minors. III. Planar tree-width, *J. Combin. Theory Ser. B*, Vol. 36, pp. 49–63 (1984)
- [Smola 03] Smola, A. J. and Kondor, : Kernels and regularization on graphs, in *COLT*, pp. 144–158 (2003)
- [Sun 05] Sun, J., Qu, H., Chakrabarti, D., and Faloutsos, C.: Neighborhood formation and anomaly detection in bipartite graphs, in *ICDM*, pp. 418–425 (2005)
- [Tong 06] Tong, H., Faloutsos, C., and Pan, J.-Y.: Fast random walk with restart and its applications, in *ICDM*, pp. 613–622 (2006)
- [Ukkonen 08] Ukkonen, A., Castillo, C., Donato, D., and Gionis, A.: Searching the wikipedia with contextual information, in *CIKM*, pp. 1351–1352 (2008)
- [Vieira 07] Vieira, M. V., Fonseca, B. M., Damazio, R., Golgher, P. B., Reis, D. d. C., and Ribeiro-Neto, B.: Efficient search ranking in social networks, in *CIKM*, pp. 563–572 (2007)
- [Watts 98] Watts, D. J. and Strogatz, S. H.: Collective dynamics of ‘small-world’ networks, *Nature*, Vol. 393, No. 6684, pp. 440–442 (1998)
- [Wei 10] Wei, F.: TED: efficient shortest path query answering on

*⁸ <http://git.io/topk-pll>

graphs, in *SIGMOD*, pp. 99–110 (2010)

[Yahia 08] Yahia, S. A., Benedikt, M., Lakshmanan, L. V. S., and Stoyanovich, J.: Efficient network aware search in collaborative tagging sites, *PVLDB*, Vol. 1, No. 1, pp. 710–721 (2008)

[Yano 13] Yano, Y., Akiba, T., Iwata, Y., and Yoshida, Y.: Fast and scalable reachability queries on graphs by pruned labeling with landmarks and paths, in *CIKM*, pp. 1601–1606 (2013)

[Yen 71] Yen, J. Y.: Finding the k shortest loopless paths in a network, *Management Science*, Vol. 17, No. 11, pp. 712–716 (1971)

〔担当委員：風間 一洋〕

2015 年 7 月 2 日 受理

—— 著 者 紹 介 ——

秋葉 拓哉

2011 年東京大学理学部情報科学科卒業。2013 年同大学院情報理工学系研究科修士課程修了。2015 年同研究科博士課程修了。同年より国立情報学研究所助教。同年より JST さきがけ研究員。データベース、情報検索、人工知能などに応用を持つ、大規模なデータを効率的に処理するための実用的なアルゴリズムとデータ構造の開発に取り組む。

林 孝紀

2014 年東京大学理学部情報科学科卒業。2014 年より同大学院情報理工学系研究科修士課程。実験的に高速なグラフアルゴリズムの開発に興味を持つ。

則 のぞみ(学生会員)

2010 年東京大学工学部システム創成学科卒業。2012 年同大学院情報理工学系研究科創造情報学専攻修士課程修了。2014 年より京都大学大学院情報学研究科知能情報学専攻博士課程。2014 年より学術振興会特別研究員(DC1)。統計的機械学習とデータマイニング、ヘルスケアへの応用に興味を持つ。

岩田 陽一

2011 年東京大学理学部情報科学科卒業。2013 年同大学院情報理工学系研究科修士課程修了。2013 年より同研究科博士課程。NP 困難問題に対するアルゴリズムの理論・応用に興味を持つ。

吉田 悠一

2007 年京都大学工学部情報科学科卒業。2009 年同大学院情報学研究科修士課程修了。2009 年 10 月～2010 年 8 月 Rutgers 大学客員研究員。2012 年京都大学大学院情報学研究科博士課程修了。同年、国立情報学研究所特任助教。2015 年より国立情報学研究所准教授。劣線形時間アルゴリズムや巨大グラフに対するアルゴリズムなど、理論的・実用的に高速なアルゴリズムの開発に従事。