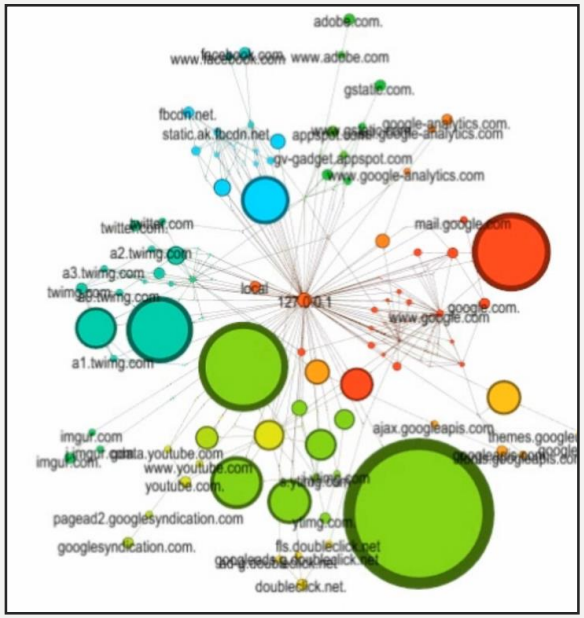
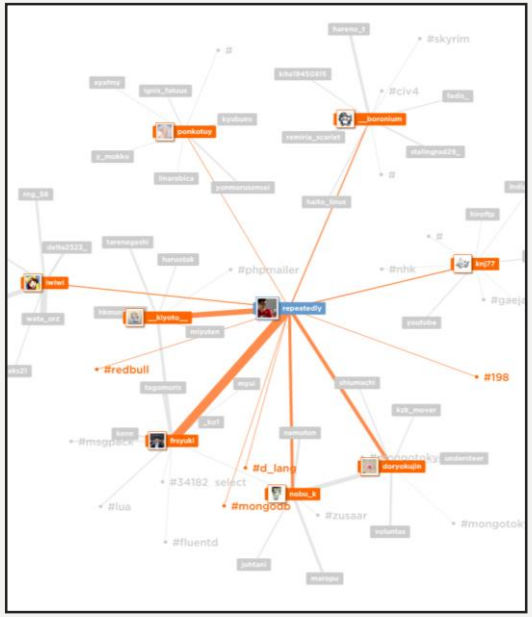
ネットワーク上の頂点間特徴量としての Top-k 距離とその高速なクエリ応答 (AAAI 2015)

秋葉 拓哉 (NII)
林 孝紀 (東京大学)
則 のぞみ (京都大学)
岩田 陽一 (東京大学)
吉田 悠一 (NII)

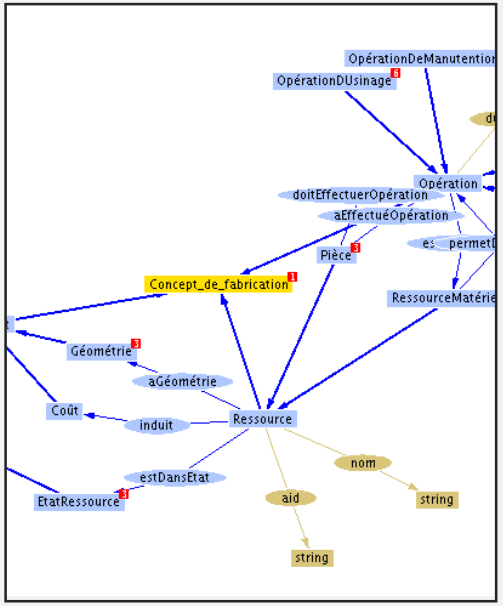
現代の大規模グラフ



ウェブグラフ



ソーシャルグラフ



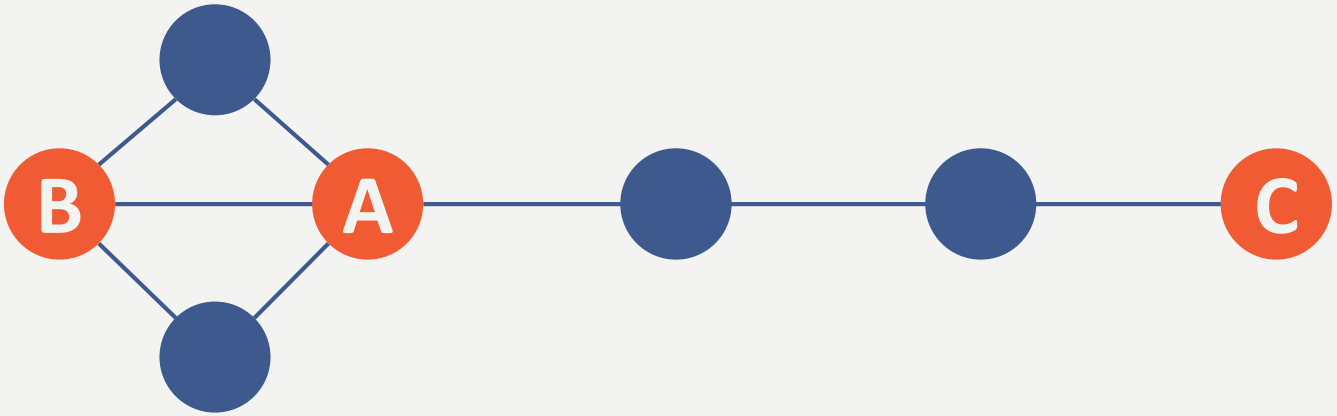
オントロジー

テキストのみからは得にくい情報がある
→グラフ解析

頂点間関連度

関連の強さ・深さを測る

グラフ解析の最も重要な部品の１つ



(A, B) > (A, C)

頂点間距離による頂点間関連度

応用

- Context-Aware Search [CIKM'08][CIKM'09]
- Socially-Sensitive Search [CIKM'07][VLDB'08][CIKM'13]

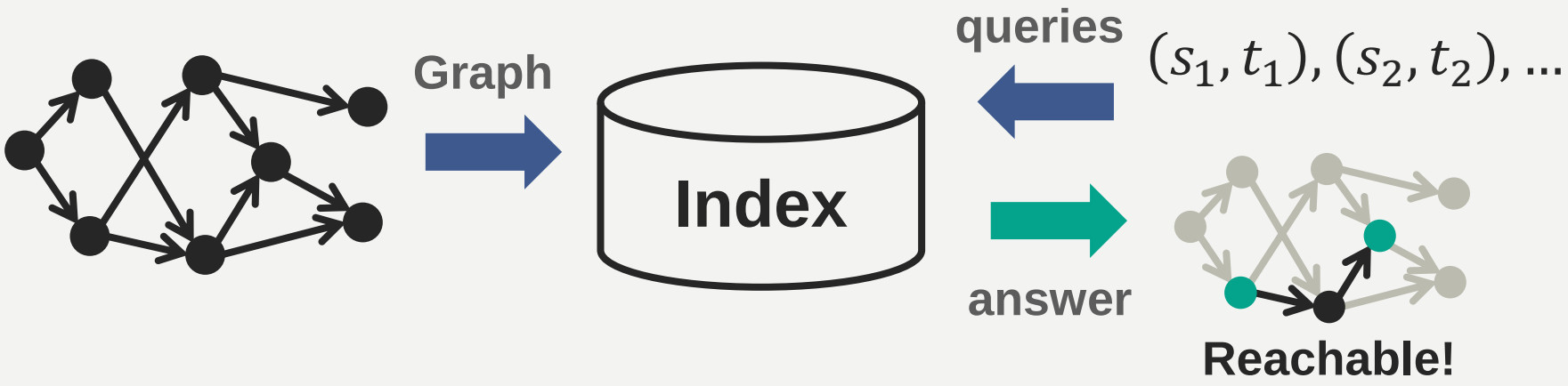
効率的な索引付け手法

- Landmark [CIKM'09] [WSDM'10] [CIKM'10] [ICDE'12]
- 木分解 [SIGMOD'10] [EDBT'12]
- 2-Hop Labeling [SIGMOD'12] [ESA'12] [SIGMOD'13]

グラフの索引付け手法

グラフ $G = (V, E)$ が与えられた時,

1. データ構造を前計算し (索引),
2. それを用いてクエリに応答する,



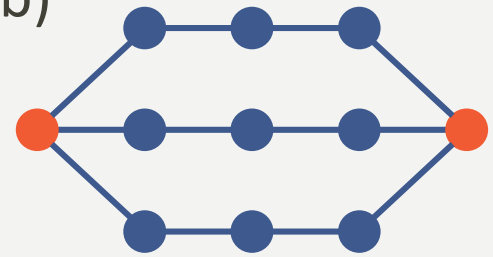
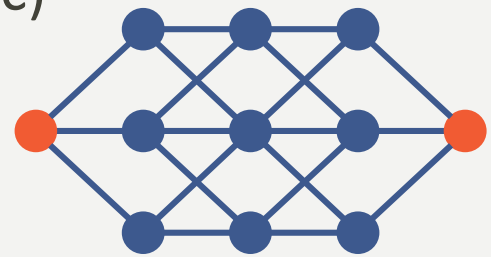
頂点間距離 を関連度として用いる

利点: スケーラブルで高速

欠点: 表現力が低い

- 重みなしグラフ
- スモールワールド性

距離は小さな整数

グラフ	距離
(a) 	4
(b) 	4
(c) 	4

Top-k 距離

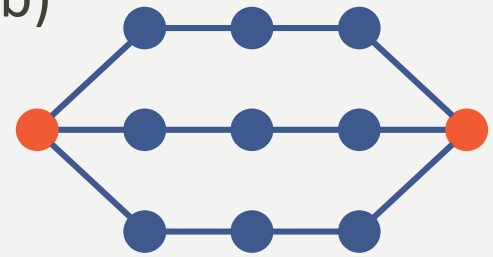
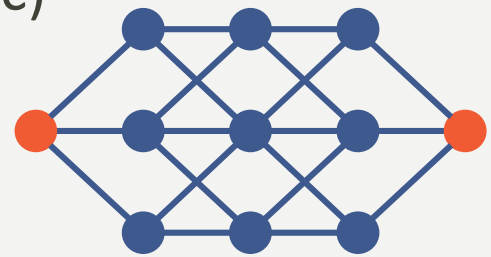
提案：頂点間関連度としての

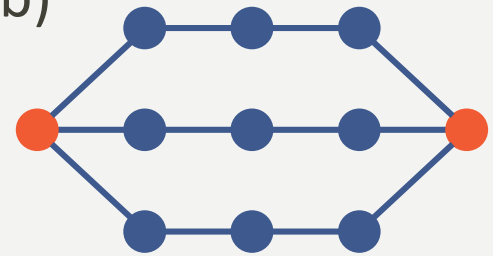
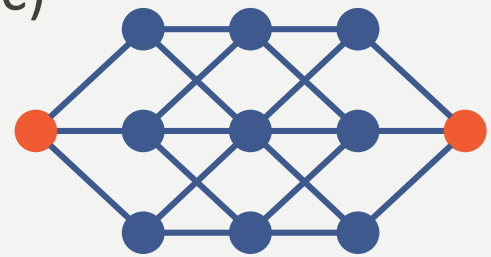
Top-k 距離

定義

k 本の最短路の長さ

- $k = 1$ → 通常の距離に一致
- 経路中に閉路を許す

グラフ	距離	Top-k 距離
(a) 	4	[4 6 6 6 6 8 8 …]
(b) 	4	[4 4 4 6 6 6 6 …]
(c) 	4	[4 4 4 4 4 4 4 …]

グラフ	距離	Top-k 距離
(a) 	4	[4 6 6 6 6 8 8 …]
(b) 	4	[4 4 4 6 6 6 6 …]
(c) 	4	[4 4 4 4 4 4 4 …]

Top-k 距離 を頂点間関連度として用いる？

Top-k 距離は k 次元のベクトル……

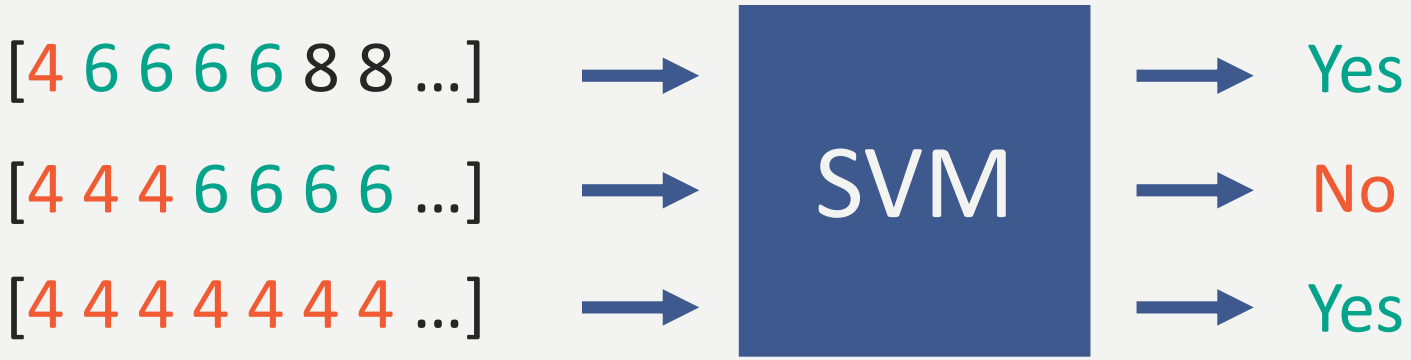
スカラ値でない

既存のものと同様に用いることができない

Top-k 距離 を頂点間関連度として用いる？

Top-k 距離は k 次元のベクトル……

特徴ベクトルとすることで
機械学習と相性が良い



既存手法 Top-k 距離の計算

幅優先探索に基づく素朴な手法 [folklore]

- $O\big((n+m)k\big)$ 時間

Eppstein のアルゴリズム [Eppstein, 1998]

- $O(n+m+k)$ 時間 だが実用的には低速

通常の距離には多くの索引付け手法が提案されてきたのに対し，

索引付け手法が存在しない

- 多くの頂点対の関連度を評価する
- グラフのサイズに比例 → スケーラブルでない

本研究の貢献

貢献 1

Top-k 距離に対する**索引付け手法**

- 枝刈りラベリング法に基づく [Akiba+,SIGMOD'13] [Akiba+,WWW'14]
- シンプルながらスケーラブルかつ高速

可能に

貢献 2

Top-k 距離を**頂点間関連度**として用いる

- 機械学習と相性が良い
- リンク予測問題における実験的評価

目次

Part 1: イントロダクション

Part 2: アルゴリズム

- 索引付け手法

Part 3: 実験結果

- 索引付け手法の評価
- Top-k 距離の有用性の実証

Top-k 距離に対する

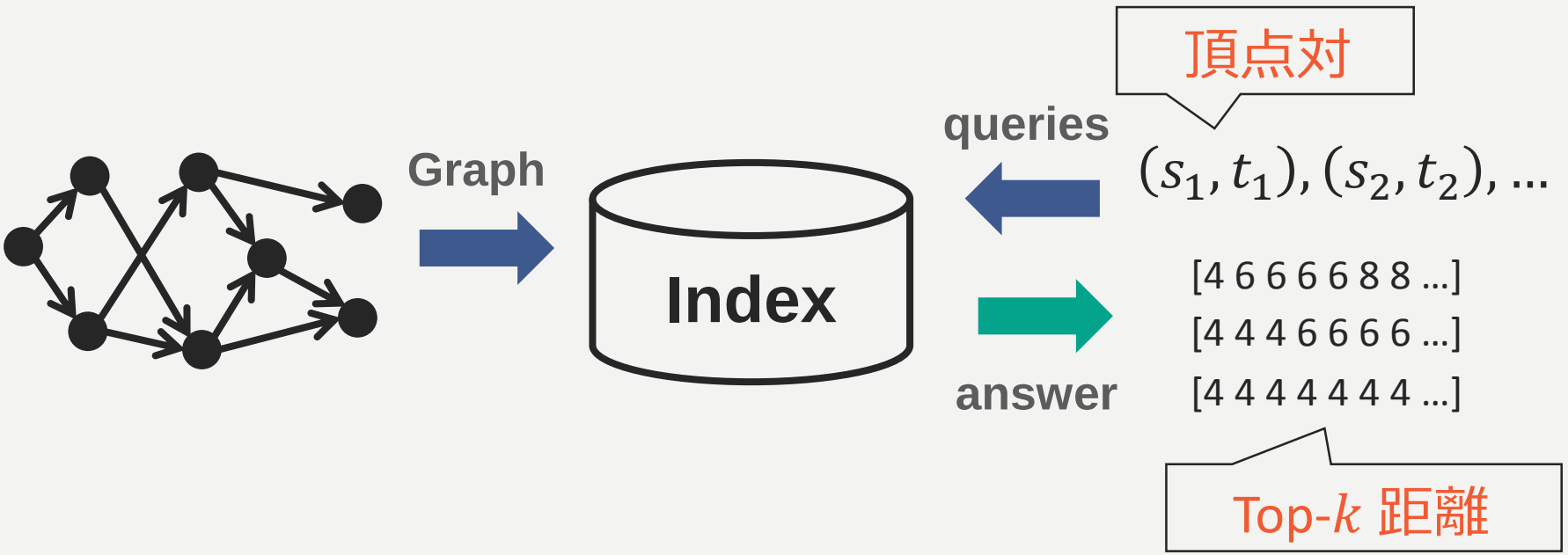
索引付け手法

Part 2: アルゴリズム

問題 Top-k 距離に対する索引付け

グラフ $G = (V, E)$ が与えられた時,

1. データ構造を前計算し (索引),
2. それを用いてクエリに応答する,



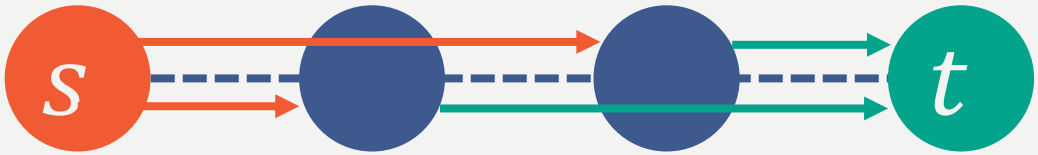
通常の距離に対する索引付け手法からの

自明な拡張か? → No! ☹

主な技術的挑戦

- 注意深く**重複計上**を避ける
  同じ経路を複数回加味してしまわないようにする

例：以下のように同じパスを 2 回表現してしまう



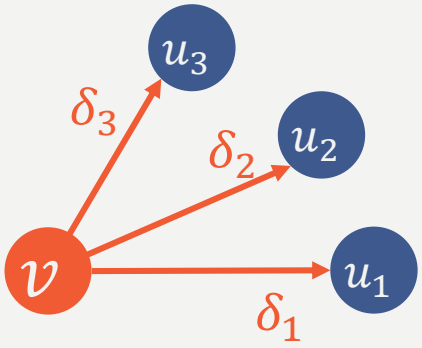
2-Hop Cover 通常の距離の索引 [Cohen+, 2002]

データ構造: ラベル を各頂点に用意

ラベル $L(v)$

- $L(v) = [(u_1, \delta_1), (u_2, \delta_2), \dots]$
- $\delta_i = d(v, u_i)$

$L(v)$



全頂点ではなく
一部の頂点のみ

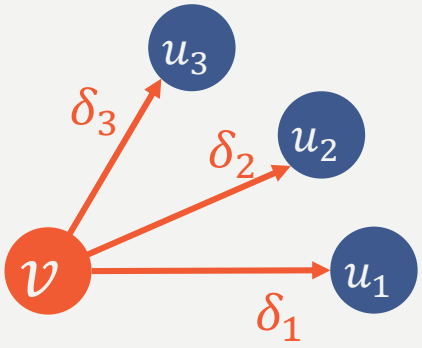
2-Hop Cover 通常の距離の索引 [Cohen+, 2002]

データ構造: ラベル を各頂点に用意

ラベル $L(v)$

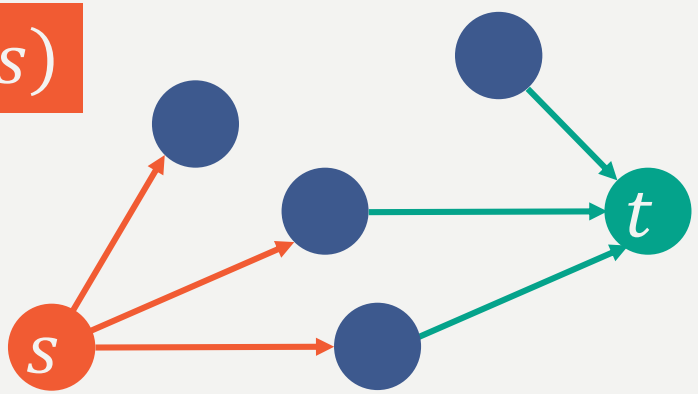
- $L(v) = [(u_1, \delta_1), (u_2, \delta_2), \dots]$
- $\delta_i = d(v, u_i)$

$L(v)$



クエリ応答: 2-hop の経路 ラベルを用いて

$L(s)$



$L(t)$

$$\min_{u \in L(s) \cap L(t)} d(s,u) + d_G(u,t)$$

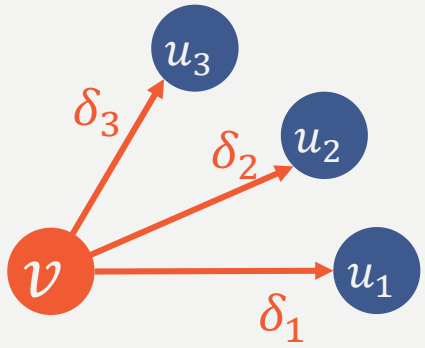
Top-k 2-Hop Cover (本研究)

データ構造

距離ラベル $L(v)$

- $L(v) = [(u_1, \delta_1), (u_2, \delta_2), \dots]$
- $\delta_i = d^{>v}_{j-\mathrm{th}}(v, u_i)$

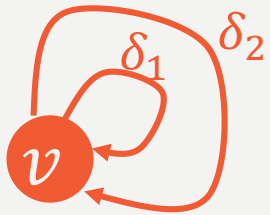
$L(v)$



閉路ラベル $C(v)$

- $C(v) = [\delta_1, \delta_2, \delta_3, \dots]$
- $\delta_i = d^{\geq v}_{i-\mathrm{th}}(v, v)$

$C(v)$



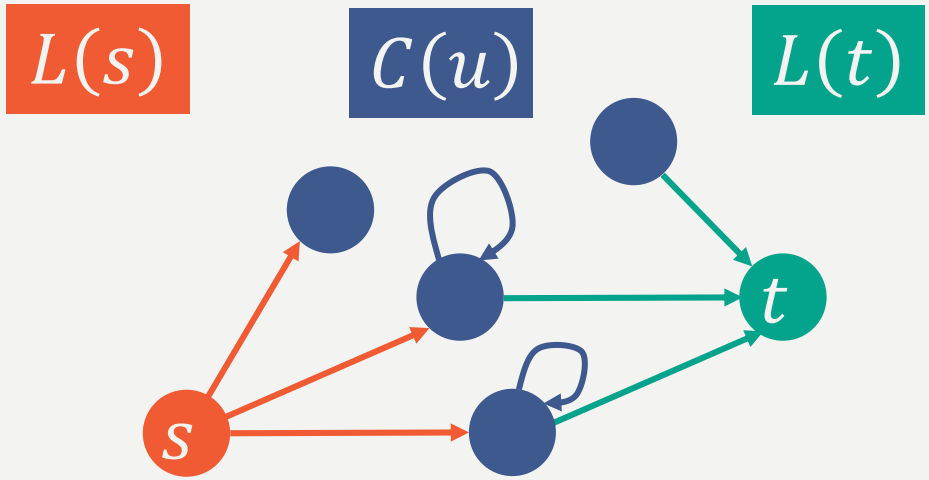
- $d^{>v}_{j-\mathrm{th}}(v, u_i)$ は**制限距離**

Top-k 2-Hop Cover (本研究)

データ構造

距離ラベル $L(v)$ + **閉路ラベル $C(v)$**

クエリ応答: 3-hop の経路



索引付け手法

技術的挑戦

- 厳密性 (正確性)
- ラベルのサイズ (索引構築時間 & クエリ時間)
- 効率性 (スケーラビリティ)

枝刈りラベリング法に基づくアルゴリズム
[Akiba-Iwata-Yoshida, SIGMOD'13][Akiba-Iwata-Yoshida, WWW'14]

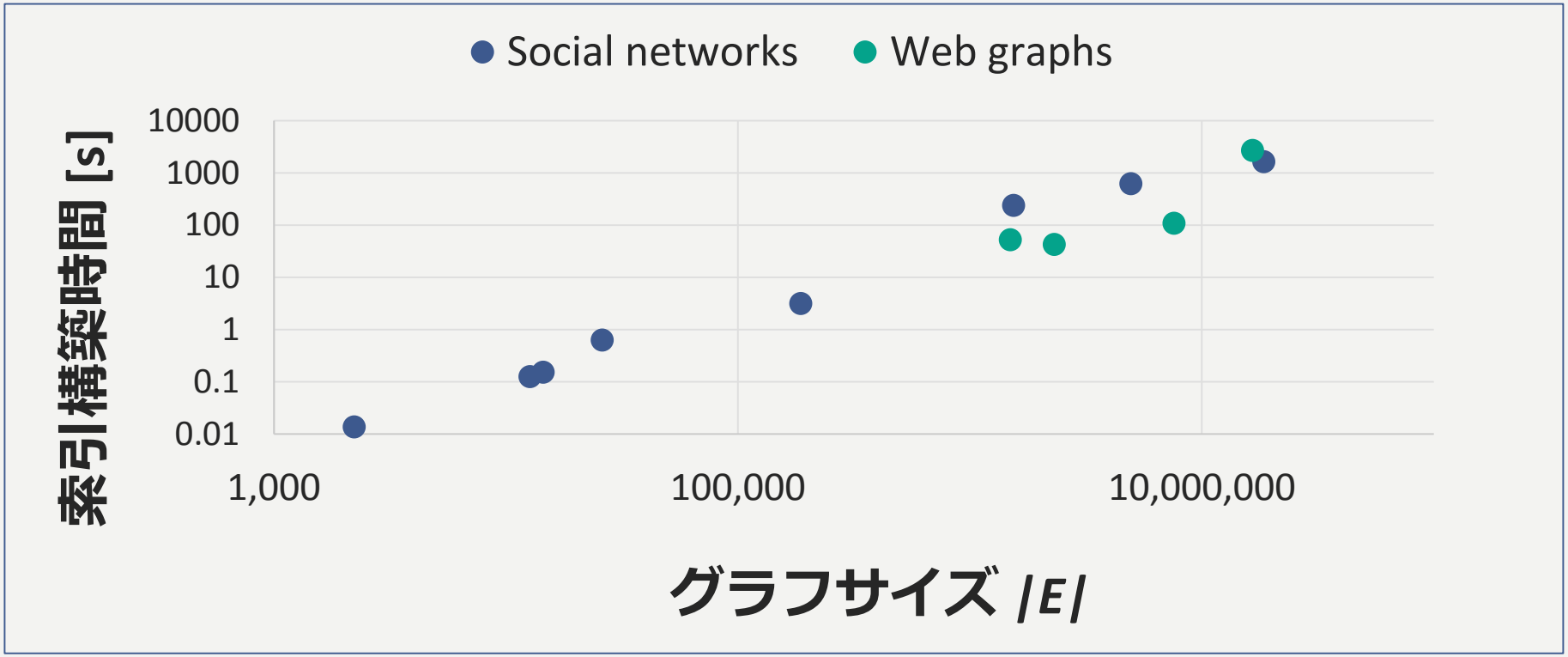
+

性能改善のための手法

評価実験

Part 3: 実験結果

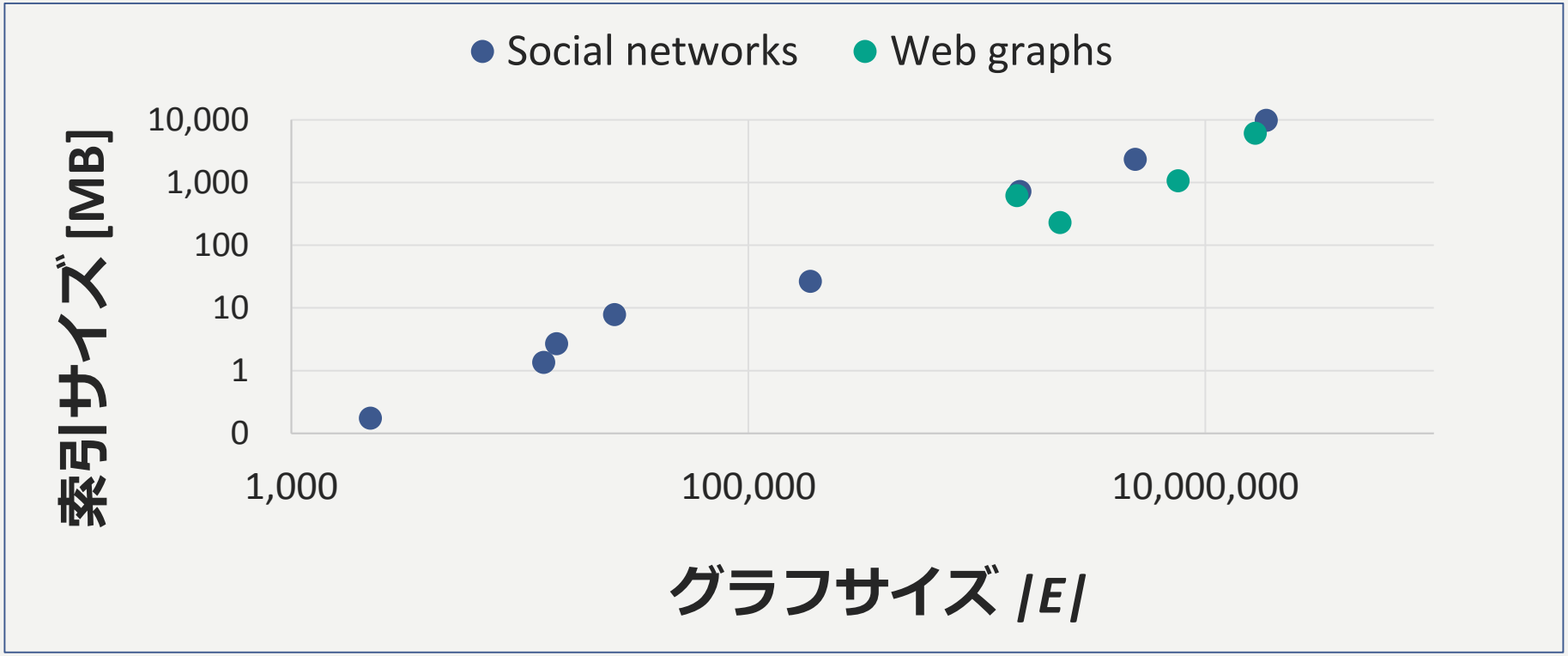
索引構築時間 実験結果



一千万辺からなる大規模グラフでも < **1 時間**

$k = 8$, Intel Xeon X5670 (2.93 GHz), 48 GB Memory, C++

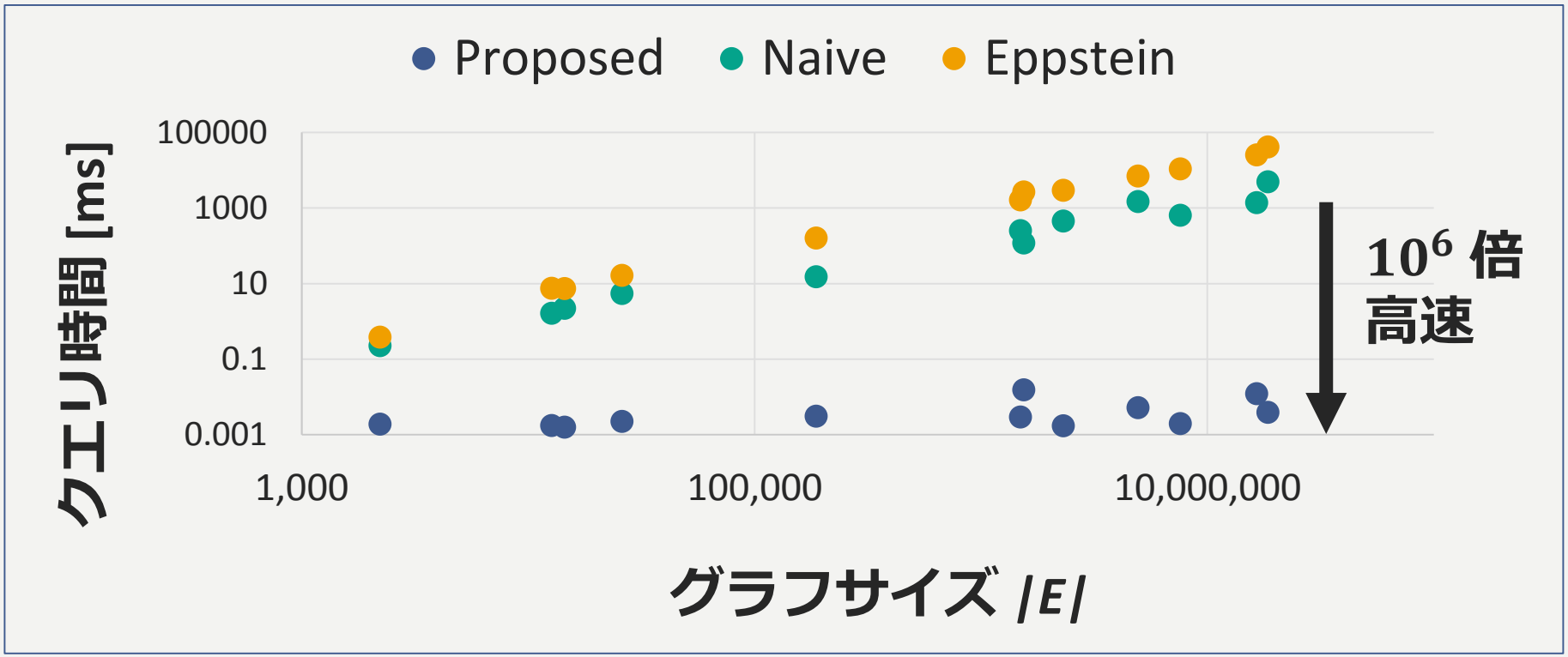
索引サイズ 実験結果



一千万辺からなる大規模グラフでも **< 10GB**

$k = 8$, Intel Xeon X5670 (2.93 GHz), 48 GB Memory, C++

クエリ時間 実験結果



一貫して <0.1ms

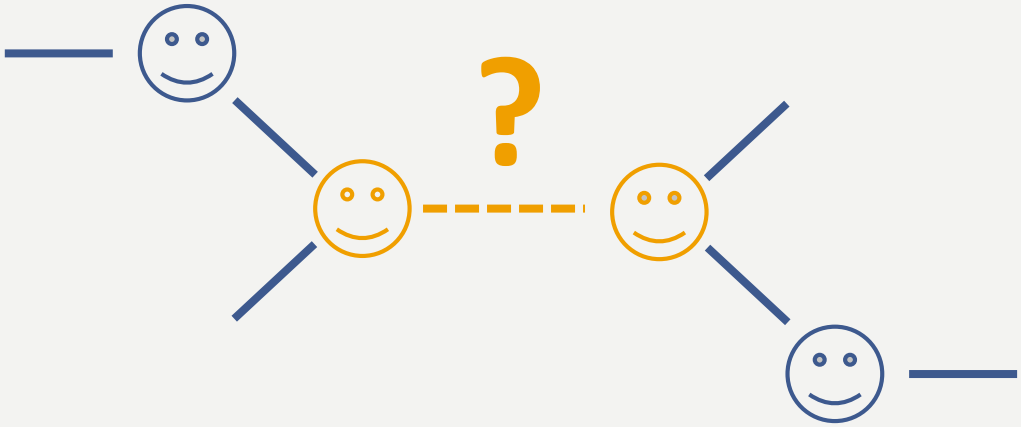
$k = 8$, Intel Xeon X5670 (2.93 GHz), 48 GB Memory, C++

リンク予測問題への
応用事例

Part 3: 実験結果

応用事例 リンク予測問題

リンク予測問題 [Liben-Nowell+, 2003]



- オンライン SNS
- タンパク質反応予測

手法

- Top-k 距離 + SVM
- 他の頂点間関連度に基づく 7 つの基礎的手法

応用事例 リンク予測問題

本研究

Dataset	CN	Jaccard	Adamic	Pref.	Comb.	SVD	RWR	Top-k	Top-1
Facebook-1	0.806	0.812	0.817	0.754	0.89	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

Precision: AUC (Area under the ROC curve)
Highlighted: statistically significant winners (by paired t-test with p < 0:05)
Setting: training (60% edges) → evaluation (40% edges), 10 times

まとめ

貢献 1

Top-k 距離に対する**索引付け手法**の提案と評価

↓

貢献 2

頂点間関連度としての Top-k 距離の提案と実証

ソフトウェア公開中! http://git.io/topk-pll

Top-k 距離を是非ご活用下さい！