DBSJ Japanese Journal
Vol. 15-J, Article No. 3, March 2017

和文タイトル：距離・キーワード・ソーシャル関係に基づくスカイライン検索

English Title: Skyline Query Processing based on Distance, Keyword, and Social relationship

田口 直弥♡ 天方 大地♢ 原 隆浩♢ 西尾 章治郎♠

Naoya TAGUCHI Daichi AMAGATA
Takahiro HARA Shojiro NISHIO

ソーシャル型位置情報サービスの出現により，地理的距離，キーワード，およびソーシャル関係を利用した情報検索に多くの注目が集まっている．本稿では，これら 3 つの指標に基づくスカイライン検索問題に初めて取り組む．スカイライン検索は，あるデータ集合の中から，どのデータにも支配されていないデータのみを検索するものであり，多くのアプリケーションを持つ．本稿では，**SKR** 木と呼ぶインデックスを用いたスカイライン計算アルゴリズムを提案し，高速なスカイライン検索を実現する．また，**Gowalla** および **Brightkite** の実データを用いた実験により，提案手法の有効性を示す．

Due to the emergence of geo-social services, information retrieval based on spatial distance, keyword, and social relationship has been receiving much attention. In this paper, we address a novel problem of skyline query processing based on the three criteria. Given a multi-dimensional dataset, a skyline query retrieves a set of non-dominated data, and has many real-life applications. We develop an index structure, namely SKR-tree, and propose an efficient algorithm for skyline query processing. The result of experiments on Gowalla and Brightkite verifies the efficiency of our algorithm.

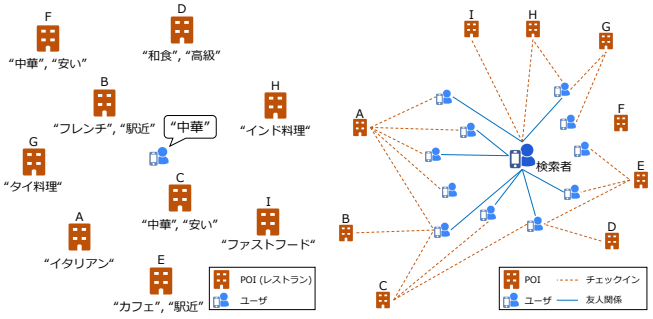
1. はじめに

近年，スマートフォンやタブレットなどのモバイル端末の普及により，位置情報サービスやソーシャルネットワーキングサービスが多くの人に利用されている [7, 12]．位置情報サービスの一例である Google Maps[1]では，ユーザが指定する位置およびキーワードに基づいて，距離が近く，検索キーワードに関連する POI (Point Of Interest) を検索できる．これに伴い，位置情報サービス上でソーシャル関係を考慮する，ソーシャル型位置情報サービスが現れ，多くの注目を集めている [6]．ソーシャル型位置情報サービスの一例である Yelp[2]では，位置およびキーワードに基づいたサービスに加え，チェックインした POI を友人と共有するなど，ソーシャル関係を考慮した様々なサービスを受けることができる．このようなサービス上で，距離が近く，キーワードの関連度が高く，さらにソーシャル関係が深い POI を選出することは，多くのアプリケーションにとって有用である [1]．この選出を効率的に行うための既存研究として，これら 3 つの指標を数値化し，その重み付き和によってスコア付けしたときの上位 k 個の POI を検索する Top-k 検索が提案されている [1]．しかし，この Top-k クエリでは，スケール普遍性がなく[3]，重みの組合せも無限にあるため，ユーザにとって困難な入力を求めている．

そこで本稿では，これら 3 つの指標に基づくスカイライン検索 [3] を提案する．あるデータ集合におけるスカイラインとは，その要素のうち，どのデータにも支配されていないものの集合であり，あるデータ o が別のデータ o' に支配されるとは，o がその全ての属性値について o' のものに劣ることをいう．ユーザにとって，ある POI に支配される POI は選出候補外とみなすことができるため，スカイライン検索により選出候補を絞り込め，容易に POI を選出できる．また，スカイラインクエリにはスケール普遍性があり，スコアリング関数の入力を必要としないといった利点がある．

以下に地理的距離，キーワード，およびソーシャル関係に基づくスカイライン検索の実用例を示す．以降，このスカイライン検索を GSK (Geo Social Keyword) スカイライン検索と呼び，GSK スカイライン検索の解を GSK スカイラインと呼ぶ．

例 1. あるユーザがレストランを選出することを考える．ユーザは，検索点を自身の現在位置，検索キーワードを “中華” とする GSK スカイライン検索により候補となるレストランを選出する (図 1(a))．また，POI の各属性値として，検索点からの距離，検索キーワードとその POI が保持するキーワードの一致数，およびその POI にチェックインした検索者の友人の数を用いる．ここで，図 1(b) は，図 1(a) の検索者に関するソーシャルグラフを表しており，グラフの辺は，ユーザ間の友人関係またはユーザの POI へのチェックイン履歴である．図 1(a) および図 1(b) より，レストラン B は，検索点からの距離，キーワードの一致数，およびチェックインした検索者の友人の数全てにおいて C のものに劣るため，C に支配されており，GSK スカイラインに含まれない．同様に D，E，F，G，H，および I も C に支配されているため，GSK スカイラインに含まれない．A は検索点からの距離およびキーワードの一致数においては C に劣るが，チェックインした検索者の友人の数は C のものよりも多いため，C に支配されない．よって A および C が GSK スカイラインとなり，ユーザはこの 2 つのレストランのみを選出候補とできる．



(a) 検索者と POI 間の地理的距離お (b) 検索者に関するソーシャルグラフよび POI が持つキーワードの例

図 1: GSK スカイライン検索の実用例

本稿では，この GSK スカイライン検索問題に初めて取り組む．

♡ 非会員 東京大学大学院工学系研究科
guchio@logos.t.u-tokyo.ac.jp,
♢ 正会員 大阪大学大学院情報科学研究科
{amagata.daichi,hara}@ist.osaka-u.ac.jp
♠ 正会員 大阪大学
nishio@ist.osaka-u.ac.jp

[1]https://www.google.co.jp/maps

[2]http://www.yelp.com/

[3]各指標の値を正規化する必要がある．

この問題の最も単純な解決法は，全ての POI について各属性値を算出し，それらそれぞれについて，他の全ての POI との支配関係を評価するというものである．しかし，この方法では POI 数が多い場合に，高速に検索結果を得ることができない [3]．また地理的距離，キーワード，およびソーシャル関係に関する属性値は，それぞれ検索位置，検索キーワード，および検索者に依存するため，検索結果を再利用することも困難である．そこで本稿では，SKR 木という階層型インデックスを事前に構築することにより，任意の検索点，検索キーワード，および検索者に対して高速に GSK スカイライン検索を行うことを試みた．SKR 木は POI の位置，保持するキーワード，およびソーシャル関係に関する属性値の上界値を基に構築する aR 木 [8] であり，GSK スカイラインに含まれない POI に対応するノードの枝刈りを効率的に行うことができる．また，SKR 木において探索するノード数を最適化し，ソーシャル関係に関する属性値の性質を考慮したアルゴリズムを提案することにより，高速な GSK スカイライン検索を実現する．ソーシャル型位置情報サービスである，Gowalla および Brightkite の実データを用いた実験から，提案手法は単純手法よりも高速に GSK スカイライン検索を行えることを確認した．

以降，2 章で本稿の問題を定義し，3 章で提案手法について説明する．4 章で評価実験の結果を示し，5 章で関連研究を紹介する．最後に，6 章で本稿をまとめる．

2. 事前準備

2.1 データモデル

本稿で検索対象となるデータは POI であり，POI の集合を P とする．ある $p_i \in P$ は，$p_i = \langle loc, key \rangle$ で表される．i は POI の識別子，loc は p_i の位置，key は p_i が保持するキーワードの集合である．

2.2 問題定義

あるユーザ u が GSK スカイライン検索を実行するとき，u は位置およびキーワードの集合を指定した GSK スカイラインクエリ q_u を発行する．つまり，$q_u = \langle loc, key \rangle$ である．q_u が与えられたとき，ある $p_i \in P$ には，地理的距離，キーワード，およびソーシャル関係に関する属性値 $p_i.G(q_u)$，$p_i.K(q_u)$，および $p_i.S(q_u)$ が与えられる．$p_i.G(q_u)$ は $p_i.loc$ と $q_u.loc$ とのユークリッド距離であり，

$$p_i.G(q_u) = dist(p_i.loc, q_u.loc) \tag{1}$$

で求められる．$p_i.K(q_u)$ は，$p_i.key$ の要素と $q_u.key$ の要素の一致数であり，

$$p_i.K(q_u) = |\, p_i.key \cap q_u.key \,| \tag{2}$$

である．また，$p_i.S(q_u)$ はユーザ u の友人のうち，p_i にチェックインしている人の数であり，$p_i.checkin$ を p_i にチェックインしているユーザの集合，$u.friend$ を u と友人関係にあるユーザの集合とすると，

$$p_i.S(q_u) = |\, p_i.checkin \cap u.friend \,| \tag{3}$$

である．ただし，u 自身のチェックインは考慮しない．例えば図 1(b) において，$A.S(q_u)$ は 4 であり，$C.S(q_u)$ は 3，また $F.S(q_u)$ および $I.S(q_u)$ は 0 となる．検索点からの距離は近い方が良いため，$p_i.G(q_u)$ は小さい方が優れており，キーワードの一致数およびチェックインしている友人の数は多い方が良いため，$p_i.K(q_u)$ および $p_i.S(q_u)$ は大きい方が優れている．ここで，本稿における支配という概念を定義する．

定義 1 (支配). POI の集合 P および GSK スカイラインクエリ q_u が与えられ，$p_i, p_j \in P$ について，$(p_i.G(q_u) \geq p_j.G(q_u)) \wedge (p_i.K(q_u) \leq p_j.K(q_u)) \wedge (p_i.S(q_u) \leq p_j.S(q_u))$ が満たされる場合，これを

$$p_i \preceq p_j \tag{4}$$

と表す．また，$(p_i \preceq p_j) \wedge [(p_i.G(q_u) > p_j.G(q_u)) \vee (p_i.K(q_u) < p_j.K(q_u)) \vee (p_i.S(q_u) < p_j.S(q_u))]$ が満たされる場合，p_i は p_j に支配されるとし，これを

$$p_i \prec p_j \tag{5}$$

と表す．

また，GSK スカイラインを次のように定義する．

定義 2 (GSK スカイライン). POI の集合 P および GSK スカイラインクエリ q_u が与えられ，P の部分集合 $P' \subseteq P$ が $\forall p_i \in P'$ に対して

$$p_i \prec \nexists p_j \in P \tag{6}$$

を満たすとき，P' を P の GSK スカイラインとする．

本稿では，GSK スカイライン検索を高速に行うことを目的とする．

2.3 単純手法

ベースラインとなる単純な GSK スカイライン検索アルゴリズムをアルゴリズム 1 に示す．アルゴリズム 1 では，まず全ての POI の各属性値を算出し (1-2 行)，その後，各 POI について，他の全ての POI との支配関係を評価することにより (3-9 行)，GSK スカイラインを求める (10 行)．しかし，アルゴリズム 1 では 2 つの理由により計算時間が増加してしまう．1 つ目は，全ての POI について各属性値を算出しなければならないことである．特に，ソーシャル関係に関する属性値の算出には，検索者の全ての友人に対して各 POI にチェックインしているか否かを判定するため，計算時間が増加してしまう．2 つ目は，支配関係の評価回数が多いことである．アルゴリズム 1 では各 POI について，他の全ての POI と支配関係の評価を行うため，計算時間が増加してしまう．そこで，これらの問題を解決するアルゴリズムを提案し，GSK スカイライン検索の効率化を図る．

Algorithm 1: BASELINE ALGORITHM

```
1  for ∀p_i ∈ P do
2      Calculate each attribute value of p_i
3  for ∀p_i ∈ P do
4      for ∀p_j ∈ P do
5          if p_i ≻ p_j then
6              P ← P\{p_j}
7          if p_i ≺ p_j then
8              P ← P\{p_i}
9              break
10 return P
```

3. 提案手法

提案手法では，階層型インデックスである SKR 木を事前に構築する．このインデックスを用いることにより，各属性値を算出する POI 数および支配関係の評価回数を共に削減できる．また SKR 木において，探索するノード数を最適化し，ソーシャル関係に関する属性値の性質を考慮することにより，効率的な GSK スカイライン検索アルゴリズムを設計する．

3.1 SKR 木

SKR 木は POI の位置，保持するキーワード，およびソーシャル関係に関する属性値の上界値を基に構築する aR 木である．SKR 木の葉ノードは POI と 1 対 1 対応している．また，SKR 木の葉ノードおよび内部ノードは，それぞれ以下の要素を保持している．

- 葉ノード
  - 対応する POI の ID
  - 対応する POI の位置
  - 対応する POI が保持するキーワードの集合

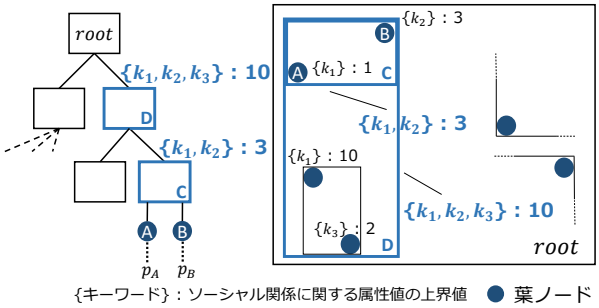


図 2: SKR 木の構造

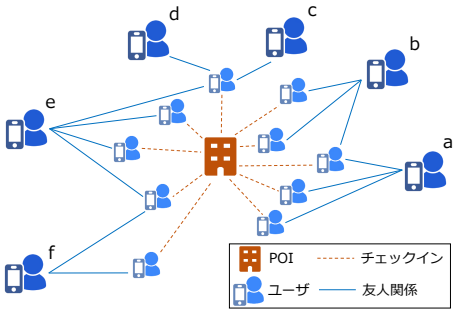


図 3: POI に関するソーシャルグラフ

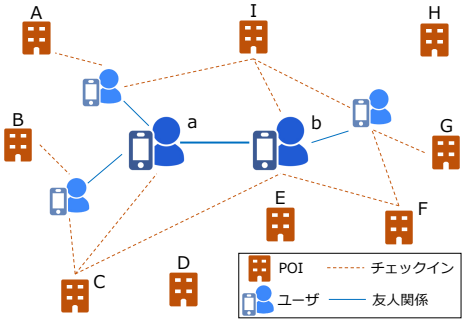


図 4: ユーザ a とユーザ b が新たに友人になった場合

  - 対応する POI のソーシャル関係に関する属性値の上界値
- 内部ノード (MBR)
  - その全ての子ノードが保持するキーワードの和集合
  - その全ての子ノードのソーシャル関係に関する属性値の上界値の最大値

図 2 は SKR 木の例を示しており，葉ノード A(B) は対応する POI の ID $p_A(p_B)$，位置，保持するキーワードの集合 $\{k_1\}(\{k_2\})$，そしてソーシャル関係に関する属性値の上界値 1(3) を持つ．ソーシャル関係に関する属性値の上界値については 3.2 節で詳しく説明する．また内部ノード C はその子ノードであるノード A および B から構成される最小外接矩形 (MBR) であり，A および B が持つキーワードの和集合 $\{k_1, k_2\}$ を持ち，A および B が持つソーシャル関係に関する属性値の上界値の最大値である 3 を持つ．同様に内部ノード D もその全ての子ノードから構成される MBR であり，キーワードの集合 $\{k_1, k_2, k_3\}$ およびソーシャル関係に関する属性値 10 を持つ．

SKR 木のノード n_i は GSK スカイラインクエリ q_u に対し，POI と同様に地理的距離，キーワード，およびソーシャル関係に関する属性値を計算でき，これらをそれぞれ $n_i.G(q_u)$，$n_i.K(q_u)$，および $n_i.S$ とする．n_i が葉ノードの場合，$n_i.G(q_u)$ は，それに対応する POI と $q_u.loc$ のユークリッド距離である．n_i が内部ノードの場合，$n_i.G(q_u)$ は，MBR の持つ領域内の点のうち，最も $q_u.loc$ に近い点と $q_u.loc$ とのユークリッド距離である．また，$n_i.K(q_u)$ は $q_u.key$ の要素と n_i が保持するキーワードの一致数であり，$n_i.S$ は n_i が保持するソーシャル関係に関する属性値である．これらの属性値を用いることにより，n_i と $p_j \in P$ の支配関係の評価が可能となる．ノードがある POI に支配される場合，そのノードの全ての子孫ノードに対応する POI は GSK スカイラインに含まれず，枝刈りが可能である．枝刈りにより，各属性値を算出する POI 数が減り，さらに支配関係の評価回数も削減されるため，高速に GSK スカイライン検索を行うことができる．

3.2 ソーシャル関係に関する属性値の上界値

前節より，SKR 木を構築するには，各 POI のソーシャル関係に関する属性値の上界値を求める必要がある．本稿では各 POI について，ソーシャルグラフ上の全てのユーザに対するソーシャル関係に関する属性値を求め，それらの中の最大値をそれぞれの POI のソーシャル関係に関する属性値の上界値とした．例えば，図 3 はある POI に関するソーシャルグラフを示しており，ユーザ a に対するこの POI のソーシャル関係に関する属性値は 3 であり，同様に b の場合も 3，c および d は 1，e は 4，そして f は 2 である．よってこの POI のソーシャル関係に関する属性値の上界値は，これらの中の最大値である 4 となる．上界値の求め方には他にも様々な方法があり，例えば POI への総チェックイン数を用いる方法などが挙げられるが，その求め方により上界値は異なったものとなる．

一方，ソーシャル関係に関する属性値の上界値が小さければ枝刈りできる POI 数が多くなり，高速に GSK スカイライン検索を行うことができる．本稿の方法で求まる上界値は，事前に求められる上界値として最小となる反面，算出にかかる計算時間は長くなる．しかし，一度この方法で上界値を算出すると，友人関係，またはチェックインに変化があった場合においても上界値を容易に更新できる．

例えば図 4 は，元々友人関係になかったユーザ a とユーザ b が新たに友人になった場合を示している．この友人関係の更新によりソーシャル関係に関する属性値の上界値が変わる可能性がある POI は，友人関係の更新を行ったユーザがチェックインしている POI のみであり，ユーザ a および b 自身がチェックインしている C，F，および I のみとなる．友人関係の更新の場合，これらの POI のソーシャル関係に関する属性値を新たに友人になった 2 人のユーザに対して算出し，これが元の上界値よりも大きくなれば上界値を更新すれば良い．この例の場合，F の上界値は変わらず 1 であり，C および I の上界値は 1 から 2 になる．

3.3 提案アルゴリズム

提案手法では，SKR 木を用いて GSK スカイライン検索を行う．本節では，これを実現する提案アルゴリズム (アルゴリズム 2) について述べる．

アルゴリズム 2 では，アクセスした各ノードをスコアリング関数 F によりスコア付けし，優先度付きキュー Q_F に格納する (スコアが大きいほど優先度は高い)．また，Q_F は SKR 木の根ノード n_{root} で初期化されている (1 行)．Q_F に格納されたノードは先頭から順にポップされていき，GSK スカイラインの途中結果に含まれる POI (以降このような POI の集合を P_{sky}^{temp} と表す) に支配されるか否かを調べられる (3-8 行)．ポップされたノードが P_{sky}^{temp} の要素に支配される場合，Q_F の次のノードをポップする．ポップされたノードが P_{sky}^{temp} のどの要素にも支配されない場合，その子ノードが内部ノードならば，その全ての子ノードの各属性値を算出し，F によりスコアを付与して，Q_F に格納する (9-13 行)．一方，その子ノードが葉ノードならば，それらの葉ノードに対応する POI である $n_i.p$ について各属性値を算出し，P_{sky}^{temp} の全ての要素との支配関係を調べる．$n_i.p$ に支配される $p_j \in P_{sky}^{temp}$ が存在する場合，p_j を P_{sky}^{temp} から除く．$n_i.p$ が P_{sky}^{temp} のどの要素にも支配さ

DBSJ Japanese Journal
Vol. 15-J, Article No. 3, March 2017

Algorithm 2: PROPOSED ALGORITHM

```
1  Q_F.push(n_root)
2  while | Q_F | ≠ 0 do
3      n_temp = Q_F.top
4      Q_F.pop
5      for ∀p_i ∈ P_sky^temp do
6          if n_temp ≺ p_i then
7              n_temp.children = ∅
8              break
9      if n_temp.children are not leaf then
10         for ∀n_i ∈ n_temp.children do
11             Calculate each attribute value of n_i
12             n_i.score = F(n_i, q_u)
13             Q_F.push(n_i)
14     else
15         for ∀n_i ∈ n_temp.children do
16             Calculate each attribute value of n_i.p
17             P_sky^temp ← P_sky^temp \ {∀p_j ∈ P_sky^temp | p_j ≺ n_i.p}
18             if n_i.p ≺ ∄p_j ∈ P_sky^temp then
19                 P_sky^temp ← P_sky^temp ∪ {n_i.p}
20 return P_sky^temp
```

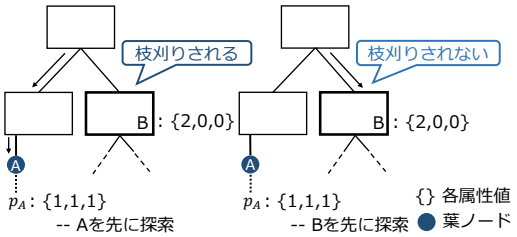


図 5: 探索順により枝刈りされるノードが変化する例

れない場合，それを P_{sky}^{temp} に加える．(14-19 行)．この操作を Q_F が空になるまで続け，Q_F が空になったときの P_{sky}^{temp} が GSK スカイラインとなる (20 行)．

3.4 スコアリング関数の設計

アルゴリズム 2 により効率的な GSK スカイライン検索を行うためには，検索終了時までに探索するノード数をできる限り少なくする必要がある．ここで，枝刈りされるノードは，ノードの探索順 (F) により変化する．図 5 では，葉ノード A に対応する POI である p_A の地理的距離，キーワード，およびソーシャル関係に関する属性値が全て 1 であり，内部ノード B の各属性値が 2，0，および 0 であるため，p_A が B を支配する．よって，A を B より先に探索すると，B の全ての子孫ノードが枝刈りされる．ところが，B が A より先に探索された場合，A によって B の全ての子孫ノードが枝刈りされることはない．そこで本稿では，アルゴリズム 2 において成り立つ定理により，探索するノード数を最適化する F を設計する．以下に，その定理を示す．

定理. GSK スカイラインクエリ q_u が与えられ，アルゴリズム 2 により GSK スカイラインを計算するとき，SKR 木中の 2 つのノード n_i および n_j に対し，F が

$$n_i \prec n_j \Rightarrow F(n_i, q_u) < F(n_j, q_u) \tag{7}$$

を満たすならば，探索するノード数は最適である．

証明. N_A および N_B をそれぞれ，式 (7) を満たす関数 F_A および F_B を用いた場合に，アルゴリズム 2 において，検索終了時までに探索される SKR 木中のノードの集合とする．このとき，$N_A \neq N_B$ と仮定すると，(i) $\exists n_A \in N_A, n_A \notin N_B$ または (ii) $\exists n_B \in N_B, n_B \notin N_A$ のいずれかが満たされるはずである．また，場合 (i) および場合 (ii) それぞれについて以下のことが成り立つ．

場合 (i)．$n_A \notin N_B$ より，F_B を用いた場合，n_A のいずれかの祖先ノード n_A^{ans} の全ての子孫ノードが枝刈りされており，$n_A^{ans} \prec \exists p_x \in P_{sky}^{temp}$ が成り立つ．ここで，p_x に対応する葉ノードを n_x とすると，SKR 木の構造上，n_x の親ノード n_x^{par} について $p_x \preceq n_x^{par}$ であるため，$n_A^{ans} \prec n_x^{par}$ である．一方，関数 F_A を用いた場合，F_A が式 (7) を満たすため，$n_A^{ans} \prec n_x^{par}$ より $F_A(n_A^{ans}, q_u) < F_A(n_x^{par}, q_u)$ となり，n_A^{ans} が探索された際，n_x^{par} あるいは n_x^{par} を支配する $p_{x'}$ に対応する葉ノードである $n_{x'}$ がすでに探索済みである．よってこのとき，p_x あるいは $p_{x'}$ が P_{sky}^{temp} に含まれており，$n_A^{ans} \prec p_x$ または $n_A^{ans} \prec p_{x'}$ より，n_A^{ans} の全ての子孫ノードは枝刈りされ，検索終了時までに探索されることはない．これは $n_A \in N_A$ に矛盾するため，場合 (i) は起こり得ない．

場合 (ii)．N_A, N_B の対称性により，場合 (i) と同様に起こり得ない．

以上により，$N_A = N_B$ となる．よって，関数 F が，式 (7) を満たすならば，アルゴリズム 2 で探索されるノードの集合は，一定 (N_{const}) となる．

また，任意の関数 F' を用いて検索を行う場合，$n \in N_{const}$ が探索されなかったとする．このとき，n のある祖先ノード n^{ans} の全ての子孫ノードが枝刈りされており，$n^{ans} \prec \exists p_y \in P_{sky}^{temp}$ が成り立つ．ところが，$n^{ans} \prec p_y$ となるとき，F' が式 (7) を満たす場合にも n は探索されない．これは N_{const} が一定のノードの集合であることに反する．従って，いかなる関数を用いた場合にも n は探索されることになり，$| N_{const} |$ はアルゴリズム 2 で探索するノード数のうち，最小 (最適) なものとなる． □

以上の議論より，アルゴリズム 2 を用いた GSK スカイライン検索を高速化するためには，式 (7) を満たすスコアリング関数 F を設計すれば良い．また，GSK スカイラインの中でも支配するデータ空間が大きい POI を優先的に検索することにより，支配関係の評価回数を削減できる．これは，ノードや POI が P_{sky}^{temp} に含まれる POI に支配されやすくなるためである．よって，データ空間が大きい POI に対応する葉ノードを子孫に持つノードへと優先的にアクセスするため，このようなノードのスコアを大きくする F を設計する．ここで，ノードをスコアリングするために，ノードが持つ地理的距離，キーワード，およびソーシャル関係に関する属性値のスコアリング関数である f_G，f_K，および f_S を設計する．F ではスコアの大きい方が優れるため，地理的距離に関する属性値のスコアリング関数は次のように設計できる．

$$f_G(n_i, q_u) = dist_{max} - n_i.G(q_u) \tag{8}$$

ただし，$dist_{max}$ は $q_u.loc$ と，n_{root} の MBR の四隅との距離のうち，最大のものとする．また，大きい方が優れるキーワードおよびソーシャル関係に関する属性値のスコアリング関数はそれぞれ，$f_K(n_i, q_u) = n_i.K(q_u)$ および $f_S(n_i) = n_i.S$ である．これらの設計により，f_G，f_K，および f_S を軸とする 3 次元空間において，支配する空間が大きいほど多くのノードおよび POI を支配することが期待できるため，スコアリング関数 F を次のように設計できる．

$$F(n_i, q_u) = f_G(n_i, q_u) \cdot f_K(n_i, q_u) \cdot f_S(n_i) \tag{9}$$

ここで，式 (9) の F は f_G，f_K，および f_S のうちどれか 1 つでも 0 となった場合，それ自身も 0 となってしまうため式 (7) を満たす F とならない．例えば GSK スカイラインクエリ q_u が与えられ，SKR 木のノード n_i および n_j の f_G，f_K，および f_S をそれぞれ $n_i.(f_G, f_K, f_S) = n_i.(1, 0, 0)$，$n_j.(f_G, f_K, f_S) = n_j.(1, 1, 0)$ とすると $n_i \prec n_j$ であるが，$F(n_i, q_u) = F(n_j, q_u) = 0$ となり，F は式 (7) を満たさない．そこで f_K および f_S をそれぞれ以下のように設計し直す．

$$f_K(n_i, q_u) = \begin{cases} n_i.K(q_u) & (n_i.K(q_u) \neq 0) \\ \alpha & (n_i.K(q_u) = 0) \end{cases} \tag{10}$$

$$f_S(n_i) = \begin{cases} n_i.S & (n_i.S \neq 0) \\ \alpha & (n_i.S = 0) \end{cases} \quad (11)$$

ただし，α は $0 < \alpha < 1$ を満たす実数とする．これにより，f_K および f_S が 0 となる場合に対応できる．また，f_G が 0 となる場合に対応するため，F を次のように設計し直す．

$$F(n_i, q_u) = \begin{cases} f_G(n_i, q_u) \cdot f_K(n_i, q_u) \cdot f_S(n_i) & (f_G(n_i, q_u) \neq 0) \\ -\frac{1}{f_K(n_i, q_u) \cdot f_S(n_i)} & (f_G(n_i, q_u) = 0) \end{cases} \quad (12)$$

式 (12) の F は，$f_G(n_i, q_u) = 0$ となる場合のスコアを負の値にすることにより，式 (7) を満たす．また，$f_G(n_i, q_u) = 0$ の場合にスコアを負の値にすると，f_G，f_K，および f_S を軸とする 3 次元空間において，支配する空間が大きい POI に対応する葉ノードを子孫に持つことが期待できるノードほどスコアが小さくなるため，逆数をとっている．

さらに本稿では，位置情報とソーシャルグラフに関するヒューリスティックを F に取り入れる．あるユーザが自身の生活区域で GSK スカイライン検索を行うとする．このとき，ユーザとその友人は，同じ生活区域で生活していることが多いため，検索者の友人は，検索点の周辺の POI にチェックインしていることが多いと考えられる．よって，検索者の周辺の POI はソーシャル関係に関する属性値が高くなりやすい．一方，SKR 木の各ノードは，各 POI のソーシャル関係に関する属性値の上界値を保持しており，上界値は検索位置に関係していない．よって F によりノードをスコアリングする際，ソーシャル関係に関する属性値が大きい POI に対応するノードを優先的に探索するためには，ソーシャル関係に関する属性値のスコアリング関数において検索位置を考慮する必要がある．従って，ソーシャル関係に関する属性値のスコアリング関数 f'_S は，式 (11) の f_S と，f'_S の検索位置依存度を表す実数 $d(> 0)$ を用いて次のように設計される．

$$f'_S(n_i, q_u) = \begin{cases} f_S(n_i) \cdot f_G(n_i, q_u)^d & (n_i.G(q_u) \neq 0) \\ f_S(n_i) & (n_i.G(q_u) = 0) \end{cases} \quad (13)$$

以上の議論に基づいて，本稿で用いるスコアリング関数を以下のように設計する．各属性値に関するスコアリング関数は，式 (8)，(10)，および (11) を用いる．最終的に，スコアリング関数は式 (12) の f_S を式 (13) の f'_S に置き換えたものとなり，次のように表される．

$$F(n_i, q_u) = \begin{cases} f_G(n_i, q_u)^{d+1} \cdot f_K(n_i, q_u) \cdot f_S(n_i) & (f_G(n_i, q_u) \neq 0) \\ -\frac{1}{f_K(n_i, q_u) \cdot f_S(n_i)} & (f_G(n_i, q_u) = 0) \end{cases} \quad (14)$$

4. 評価実験

本章では，提案手法の性能評価のために行った実験の結果を紹介する．比較手法として，2.3 節で紹介したアルゴリズム (単純手法と称する) を用いた．全てのアルゴリズムは C++で実装されており，実験は 3.47GHz Intel(R)Xeon(R)CPU および 192GB RAM で構成される PC 上で行った．

4.1 設定

本実験は，Gowalla[4] および Brightkite[5] の 2 つの実データを用いて行った．Gowala におけるユーザ数，POI 数，総友人関係数，および総チェックイン数は，それぞれ 196,591，1,280,9691，1,900,665，および 3,981,334 である．一方，Brightkite は，58,228，772,965，428,157，および 1,072,965 である．これらのデータセットは，どちらもソーシャル型位置情報サービスのものであり，ID および位置情報を保持する POI データ，ユーザの POI へのチェックインデータ，そしてユーザ間の友人関係のデータにより構成される．

[4]http://snap.stanford.edu/data/loc-gowalla.html
[5]http://snap.stanford.edu/data/loc-brightkite.html

また，各 POI に 10,000 種類の人工キーワードを 5 個ずつ付与した．これらの 5 個の人工キーワードに相関性 (例えば，“レストラン” と “ランチ” というキーワードは同じ POI に付与されている可能性が高い) を持たせるため，ランダムで 1 つの自然数 n を選び n，$n+1$，$n+2$，$n+3$，$n+4$ をそれぞれ 10,000 で割った余りを用いた．

実験は，ランダムに選ばれたユーザがある検索点で，1 から 5 個のキーワードを用いて GSK スカイライン検索を行う，というシナリオで行った．検索点には，検索ユーザ自身がチェックインしている POI により構成される MBR をその検索ユーザの生活区域とみなし，その MBR 内のランダムな地点を用いた．また，検索キーワードには，POI に付与したものと同様の方法で選んだ人工キーワードを用いた．

4.2 実験結果

本節では，評価実験の結果を示す．なお，式 (10) および (11) における α は，準備実験の結果に基づいて $\alpha = 0.001$ とした．

インデックスの構築時間． 表 1 に各データセットにおけるソーシャル関係に関する属性値の上界値の計算時間を示す．本稿におけるソーシャル関係に関する属性値の上界値の計算は，上界値として最小となる反面，算出にかかる計算時間は長くなるが，Gowalla で 67.7[sec]，Brightkite で 35.2[sec] で計算できている．ソーシャル関係に関する属性値の上界値の計算時間は，データセットのユーザ数と，その平均友人数および平均チェックイン数に依存する．Brightkite に比べて，ユーザ数，平均友人数，および平均チェックイン数が全て多い Gowalla の方が，表 1 のようにソーシャル関係に関する属性値の上界値の計算に多く時間がかかる．また，表 2 に各データセットにおける SKR 木の構築時間を示す．SKR 木の大きさは各データセットにおける POI 数に依存する．Brightkite に比べて POI 数が多い Gowalla の方が，表 2 のように SKR 木の構築に時間がかかる．

表 1: ソーシャル関係に関する属性値の上界値の計算時間

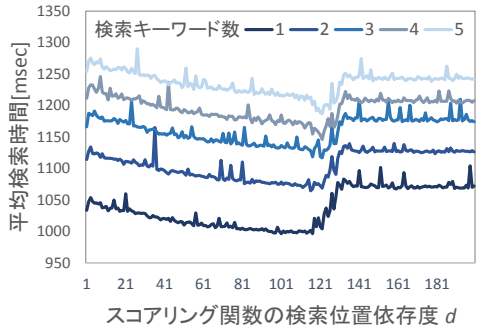
データセット	計算時間 [sec]
Gowalla	67.7
Brightkite	35.2

表 2: SKR 木の構築時間

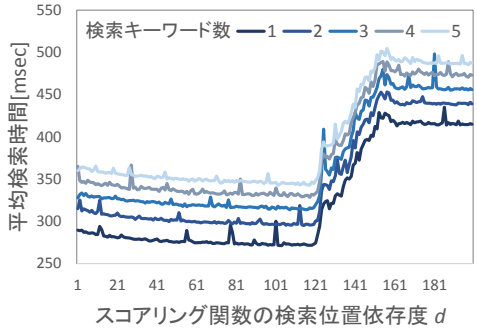
データセット	構築時間 [sec]
Gowalla	4,376
Brightkite	2,376

d の影響． 図 6 に，検索キーワード数 1 から 5 について，式 (13) の検索位置依存度 d を変えたときの提案手法の平均検索時間を示す．横軸は d の値，縦軸はランダムに選んだ 100 人のユーザが GSK スカイライン検索を行った際の平均検索時間を表す．また，それぞれのデータセットについて，各検索キーワード数毎に最も平均検索時間が短くなった d の値を表 3 に示す．どちらのデータセットでも，各検索キーワード数について，ある一定のところまでは平均検索時間が短くなり続け，その後は急激に長くなっている．また，最適な d の値が 2 つのデータセット間で類似しているのは，この 2 つのデータセットの POI の位置の分布が類似しているためである．

検索キーワード数の影響． 図 7 に，検索キーワードの数を変えたときの提案手法および単純手法の平均検索時間の結果を示す．横軸は検索キーワード数，縦軸はランダムに選んだ 100 人のユーザが GSK スカイラインを行った際の平均検索時間を表す．提案手法では，各データセットについて，各検索キーワード数毎に表 3 に示される d を用いた．検索キーワード数の増加に伴い，どちらのデータセットでも，それぞれの手法で少しずつ平均検索時間が



(a) Gowalla



(b) Brightkite

図 6: d の影響

表 3: 検索キーワード数毎の最適な d

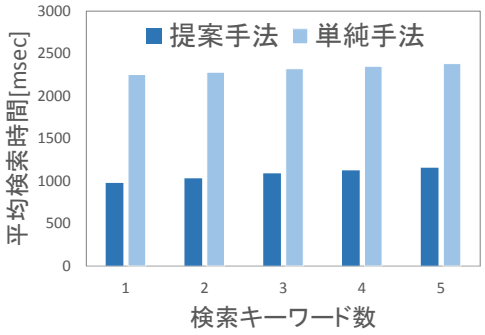
データセット	検索キーワード数				
	1	2	3	4	5
Gowalla	101	116	122	122	122
Brightkite	119	119	119	120	120

増加している．検索キーワード数が増加することで，キーワードの一致数のみが多い POI や，キーワードの一致数がある程度多くて他の属性値も良い POI など，様々な POI が現れる．よって，各 POI が支配する POI 数が少なくなり，支配関係の計算回数が増加してしまう．しかし，提案手法は単純手法に対し，Gowalla では 43%から 49%の時間で，そして Brightkite では 12%から 16%の時間で GSK スカイラインを検索している．

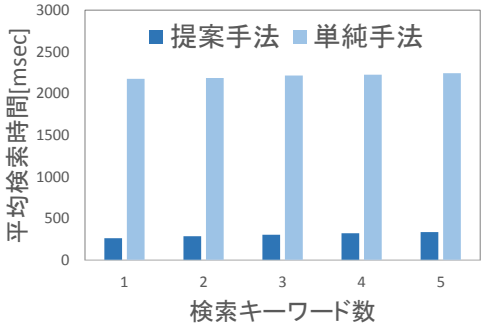
ここで，図 8 に，各データセットにおける，全 POI のうち枝刈りされた POI の割合を示す．図 7 および図 8 から，枝刈りにより検索時間が削減されたことが分かる．また，Brightkite において提案手法を用いた場合，Gowalla における場合より，平均検索時間はより短く，枝刈りされた POI の割合はより多くなっている．これは，Brightkite では Gowalla に比べ，ソーシャル関係に関する属性値の上界値と実際に検索時に求めるソーシャル関係に関する属性値の差が小さく，POI が SKR 木のノードを支配する可能性が高いためである．

5. 関連研究

位置情報サービスの普及により，地理的距離およびキーワードに基づく検索についての研究が多く行われている [4, 9]．文献 [11] では，Top-k 検索によく用いられている IR(Inverted R) 木 [5] をスカイライン検索に用いており，地理的距離とキーワードに基づくスカイライン検索に R 木に基づくインデックスを用いることの有効性が示されている．IR 木では，各ノードがその子ノードの持つ全てのキーワードを転置インデックスにより管理しているが，



(a) Gowalla



(b) Brightkite

図 7: 検索キーワード数の影響

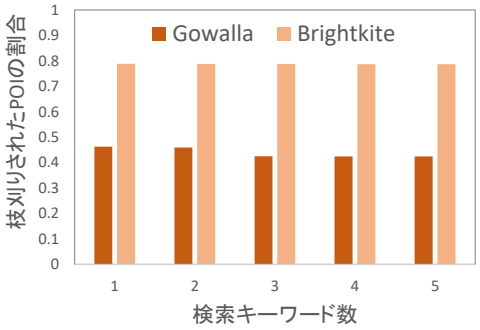


図 8: 枝刈りされた POI の割合

SKR 木では，各ノードの管理するキーワードがどの子ノードに由来するものであるかを特定する必要がないため，転置インデックスを使用していない．また，文献 [9] では S2I(Spatial Inverted Index) というインデックスを用いることにより，地理的距離およびキーワードに基づく Top-k 検索を効率的に行っている．S2I は転置インデックスによりキーワードを管理しており，同時に，aR 木を用いてデータの管理を行っている．この文献から，地理的距離，キーワードに加え，ある値を保持するデータを扱う場合，転置インデックスによりキーワードを管理し，同時に aR 木を用いることが有効であると分かる．本稿で用いた SKR 木が aR 木であるのは，この文献に着想を得ている．

地理的距離およびソーシャル関係に基づく検索についての研究も多く行われている [2, 13]．ソーシャル関係の定義には様々なものがあり，これらの文献ではその定義に基づく様々なインデックスが用いられている．文献 [6] では，地理的距離およびソーシャル関係に基づくスカイライン検索を提案しており，地理的距離についてのみ考慮するインデックスを用いた手法を提案している．し

かし，これらの文献で扱われている方法では，地理的距離，キーワード，およびソーシャル関係の全ては考慮していない．

一方，地理的距離，キーワード，およびソーシャル関係全てに基づく検索についての研究も行われ始めている．文献 [1] はその一例であり，地理的距離，キーワード，およびソーシャル関係に基づく Top-k 検索のフレームワークを示し，階層型インデックス GSKI(Geo-Social Keyword Index) を用いて効率的な検索を行っている．GSKI はグリッドに基づく木構造であり，各ノードは転置インデックスによりキーワードを管理している．また，GSKI の各ノードは，対応するグリッド内の POI へチェックインしたユーザを，ブルームフィルタにより管理している．しかし，実世界において，POI の存在位置には偏りが存在する (例えば，郊外における POI 数よりも都市部における POI 数の方が多いなど)．よって存在位置を考慮しないグリッドを用いると，グリッドの粒度が細かい場合，POI が存在しない領域に対応するノードが多くできてしまい，計算効率およびメモリの利用効率が悪くなる．また，グリッドの粒度が粗い場合，POI が多く存在する領域に対応するノードの各属性値が良いものとなりやすく，枝刈りができないため，計算効率が悪くなる．そこで本稿では，R 木に基づく SKR 木を用いた．その他にも地理的距離，キーワード，およびソーシャル関係全てに基づく検索についての文献は存在する [10] が，スカイライン検索を扱っているものは，筆者らの知る限り存在しない．

6. おわりに

本稿では，地理的距離，キーワード，およびソーシャル関係に基づくスカイライン検索問題に取り組んだ．単純な手法による検索では検索時間がかかりすぎるため，本稿では SKR 木というインデックスを用いたアルゴリズムを提案した．本稿で提案したアルゴリズムは，SKR 木において探索するノード数を最適化でき，またソーシャル関係に関する属性値の性質を考慮することにより，高速に GSK スカイラインを検索できる．また，実データを用いた実験から，提案手法は単純手法に比べて検索時間を削減できることを確認した．

本稿では，検索点を静的なものと想定したが，移動しながらスカイラインとなる POI をモニタリングすることも考えられる．今後は，検索点が移動する場合でも効率的にスカイラインを更新する問題に取り組む予定である．

謝辞．本研究の一部は，文部科学省科学研究費補助金 · 基盤研究 (A)(JP26240013) および JST 国際科学技術共同研究推進事業 (戦略的国際共同研究プログラム) の研究助成によるものである．ここに記して謝意を表す．

[文献]

[1] R. Ahuja, N. Armenatzoglou, D. Papadias, and G. J. Fakas, “Geo-Social Keyword Search”, In *SSTD*, pp.431–450, 2015.

[2] N. Armenatzoglou, S. Papadopoulos, and D. Papadias, “A general framework for geo-social query processing”, *PVLDB*, 6(10):pp.913–924, 2013.

[3] S. Borzsony, D. Kossmann, and K. Stocker, “The skyline operator”, In *ICDE*, pp.421–430, 2001.

[4] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, “Collective spatial keyword querying”, In *SIGMOD*, pp.373–384, 2011.

[5] G. Cong, C. S. Jensen, and D. Wu, “Efficient retrieval of the top-k most relevant spatial web objects”, *PVLDB*, 2(1):pp.337–348, 2009.

[6] T. Emrich, M. Franzke, N. Mamoulis, M. Renz, and A. Züfle, “Geo-Social Skyline Queries”, In *DASFAA*, pp.77–91, 2014.

[7] K. Mouratidis, J. Li, Y. Tang, and N. Mamoulis, “Joint Search by Social and Spatial Proximity”, *IEEE TKDE*, 27(3):pp.781–793, 2015.

[8] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, “Efficient OLAP operations in spatial data warehouses’, In *SSTD*, pp.443–459, 2001.

[9] J. B. Rocha-Junior, O. Gkorgkas, S. Jonassen, and K. Nørvg, “Efficient processing of top-k spatial keyword queries’, In *SSTD*, pp.205–222, 2011.

[10] P. Shankar, Y.-W. Huang, P. Castro, B. Nath, and L. Iftode, “Crowds replace experts: Building better location-based services using mobile social network interactions’, In *PerCom*, pp.20–29, 2012.

[11] J. Shi, D. Wu, and N. Mamoulis, “Textually Relevant Spatial Skylines”, *IEEE TKDE*, 28(1):pp.224–237, 2016.

[12] Y. Tao and C. Sheng, “Fast Nearest Neighbor Search with Keywords”, *IEEE TKDE*, 26(4):pp.878–888, 2014.

[13] D.-N. Yang, C.-Y. Shen, W.-C. Lee, and M.-S. Chen, “On socio-spatial group query for location-based social networks”, In *KDD*, pp.949–957, 2012.

田口 直弥 Naoya TAGUCHI
2016 年大阪大学工学部卒業．東京大学大学院工学系研究科在学中．

天方 大地 Daichi AMAGATA
大阪大学大学院情報科学研究科助教．2015 年大阪大学大学院情報科学研究科博士後期課程修了，情報科学博士．データベースシステムの研究に従事．IEEE，ACM，情報処理学会各会員．

原 隆浩 Takahiro HARA
大阪大学大学院情報科学研究科教授．1997 年大阪大学大学院工学研究科博士前期課程修了，工学博士．データベースシステム，分散処理の研究に従事．IEEE，ACM，電子情報通信学会各会員．

西尾 章治郎 Shojiro NISHIO
大阪大学総長．1975 年京都大学工学部卒業．1980 年同大学院工学研究科博士後期課程修了，工学博士．京都大学工学部助手等を経て，1992 年大阪大学工学部教授，2015 年大阪大学総長となり，現職に至る．文部科学省科学官，大阪大学理事・副学長等を歴任．データ工学の研究に従事．本会理事，監事，会長を歴任．紫綬褒章を受章し，本会より功労賞，論文賞を受賞．