D-005

空間 Web データ上の m-最近接キーワード検索問題における点データスコアの導入
Intriduction of Point Data Score in m-Closest Keyword Search Problem on Spatial Web Data

津野　貴大 *　大森　匡 *　藤田　秀之 *　新谷　隆彦 *
Takahiro Tsuno　Tadashi Omori　Hideyuki Fujita　Takahiko Shintani

1　はじめに

最近の Web データには地図上の緯度経度情報を持つ写真やテキストが増えており，これらは空間 Web と呼ばれる．近年，空間 Web を用いて地図・空間情報を抽出する研究が盛んである．その一つに，m-最近接キーワード検索問題（mCK 検索）がある．mCK 検索問題とは，個々の空間 Web データが地図上の一点を表すとみなしたとき，キーワード m 個の入力 Q を受けて，高々 m 点の集合でその m 点全体で Q を満たし，かつ，その m 点が最も相互に近く位置しているような m 点を決定する問題である．これによって，Q を満たす地図上の位置を発見することを狙っている．[1][2] しかし，従来の問題設定では，局所的な解しか求まらない問題やテキスト情報と位置情報がそぐわない問題が発生する．これを本解決するため，本稿では，mCK 検索で空間 Web 上の各オブジェクトにスコアを持たせる手法とスコア関数の有用性を検討する。

2　mCK 検索問題

2.1　mCK 検索問題の定義

本稿では個々の Web データ 1 つは地図上の 1 点を表す位置情報を持っていると仮定する．このようなデータには，ブログ，ツイッター，写真などがある．以下では，こうした点データをオブジェクトと呼ぶ．このとき，mCK 検索とは，ユーザが m 個のキーワード Q を入力したとき，

1. 個々のオブジェクト o は位置情報 $o.l$ とテキスト情報 $o.t$ をもつ．$o = [o.l, o.t]$ である．
2. 個々のオブジェクト o が持つテキスト情報は少なくとも 1 つ以上のキーワードに該当する，という制約で高々 m 個のオブジェクト集合 $O = \{o_{i1}, o_{i2}, ..., o_{il}\}$ を考える
3. Q の各キーワードは，少なくとも 1 つの O のオブジェクトによって満たされている
4. Q を満たすオブジェクト集合 O について，その直径 $diam(O)$ を 2 オブジェクト間距離 $dist$ の最大値として与える．$diam(O) = \max_{o_i, o_j \in O} dist(o_i, o_j)$ となる．このとき，Q を満たすオブジェクト集合 O のうち，$diam(O)$ が最小となる O を最適解 O_{opt} として返す．

という検索問題である．[1][2]

mCK 検索の例を示す．図 1 では，地図上にカフェ，学校，池の 3 種類の写真が存在する．図 1 の例で $Q = \{$カフェ，学校，池$\}$ で mCK 検索を行うと，3 種類の写真からなるオブジェクト集合で最も直径が小さいものを探す．すると，赤い線で結ばれたオブジェクト集合が最も直径が小さいのでこれを O_{opt} として返す．O_{opt} が Q に応じた最適な位置となる．

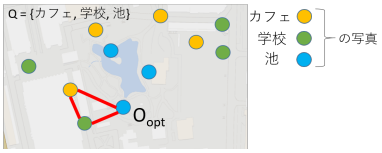

図 1　mCK 検索の例

2.2　従来解法

mCK 検索問題はキーワード数 m について NP 完全である [2]．Qiu らは mCK 検索問題の厳密解列挙法として Pairwise Expansion(PE 法) を提案している [3]．　本稿ではこの PE 法に基づいてオブジェクトにスコアを導入する．

PE 法は，まず，キーワードを一つでも満たすオブジェクト全てをデータベースから取り出して四分木に格納する．（この四分木は，各ノードあたり，そのノードに属す点データ集合について MBR 情報も補助的に持たせている．）この後，PE 法の基本戦略は，Object-Pair Generation と呼ぶ処理で，mCK 解の直径になりうる 2 オブジェクトの組 (o_1, o_2) を最近接二点探索の戦略で探していき，候補 (o_1, o_2) が見つかる度に，これを直径とした mCK の解が存在するかを検査する．この検査は，Object-Pair Check 関数 $check(o_1, o_2)$ を呼び出して行う．

PE 法自体 [3] は，Object-Pair Generation 部や Object-Pair Check 部で無駄な探索を避けるために複雑な絞り込み戦略を用いる．本稿は，これらの戦略を使わず，PE 法の骨組みだけを使ってスコア関数を導入する．以下に，Top-K 解の場合の各処理の概要を示す．

Object-Pair Generation 処理:

用語として，四分木の中間または葉ノード N_1, N_2 の組をノードペアと呼ぶ．($N_1 = N_2$ の場合も含む)．$mindist(N_1, N_2)$ はノード N_1, N_2 間の最小距離であり，MBR を使って求める．

Object-Pair Generation 部は，根のペアから始めて，相異なる 2 キーワード以上を含む全ノードペアの再帰探索を $mindist()$ の小さい順に行う．途中，葉ノードペアに到達したときは，そこに存在する (2 キーワード以上を持つ) オブジェクトのペア (o_1, o_2) ($o_1 = o_2$ の場合も含む) について，$dist(o_1, o_2)$ の昇順に $check(o_1, o_2)$ を呼び出し，(o_1, o_2) を直径とした mCK 解で上位 K 解の候補になるものを求め，全体の上位 K 解候補を更新する．また，ノードペアの再帰探索では，kNN 探索と同様にして上位 K 解の直径閾値による枝刈りを行う．

Object-Pair Check 処理 $check(o_1, o_2)$

* 電気通信大学 The University of Electro-Communications

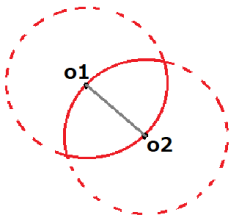

図 2　shuttle($\langle o_1, o_2 \rangle$)

用語：オブジェクトペア (o_1, o_2) が与えられた時，オブジェクト o について $dist(o_1, o_2) \geq dist(o_1, o)$ かつ $dist(o_1, o_2) \geq dist(o_2, o)$ となる o のエリアを shuttle($\langle o_1, o_2 \rangle$) と呼ぶ．(図 2 の赤実線で囲われた部分)．

このとき，$check(o_1, o_2)$ 関数は，$\{o_1, o_2\}$ の 2 オブジェクトだけでキーワード m 個を全て満たすなら true を返す．それ以外の場合，Q のうち $\{o_1, o_2\}$ が持たないキーワード集合 Q' を求め，$shuttle(o_1, o_2)$ 内のオブジェクトのペア (o_x, o_y) について $dist(o_x, o_y)$ の昇順に再帰的に $check(o_x, o_y)$ を呼び出す．すなわち，$o_x = o_y$ の場合は残る一点で Q' を満たすものを探し，$o_x \neq o_y$ なら (o_x, o_y) を含む高々 $m-2$ 個のオブジェクトによる Q' の充足を検査する．この検査が成功した時点で $check(o_1, o_2)$ は（(o_1, o_2) を直径とした mCK 解が存在するので）true を返し，失敗したら false を返す．

3　従来解法における問題点

従来の mCK 検索で直径が小さい順から上位 K 個の解を求める Top-KmCK 検索を行うと， 以下の問題が発生する．従来解法では，写真にタグをつけてアップロードする Flickr という SNS のデータをデータセットとして用いている。(写真につけられているタグをテキスト情報としている)

問題点：上位 K 解が地図上で 2-3 カ所の場所にのみ局所的に集中する問題である．原因は， 従来の Top-KmCK 検索では直径が閾値以下となるオブジェクトセットの解を求めているからである．上位の解付近にオブジェクトが複数存在すると， 上位の解の直径を使った別の解を作るので， 解が重なって出現する．

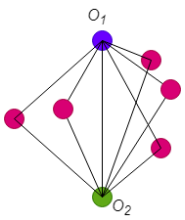

図 3　問題点

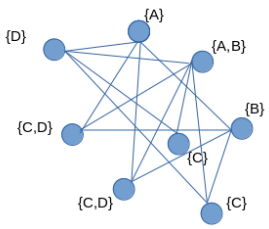

図 4　問題点

例えば，図 3 では，各オブジェクトが入力キーワード Q を満すキーワードを 1 つだけもっているとする．オブジェクト o_1, o_2 を直径とするオブジェクトの組み合わせ 5 つ全てが解となってしまう．図 4 では入力キーワード Q = {A,B,C,D} とし，各オブジェクトが Q を満すキーワードを 1 つ以上持っているとしたときの解となるオブジェクトの組み合わせである．

4　研究課題の設定

4.1　スコア付き mCK 問題の設定

従来の問題設定では上記の問題点を根本的に解決することはできないため，本研究では，各点 o_x にスコアをつけるスコア関数 $score(o_x)$ を導入し新たに問題を設計することで対応する．以下に新たに設定した Top-KmCK 検索で扱う問題を記述する．

問題設定:実効直径 $ediam$ が小さい順に入力キーワード Q を満たす mCK 解を k 個見つける．ただし，一つの $ediam$ につき,Q を満たす O のうちスコアの総和が最大のもの O' を代表解として使う．直径となるオブジェクトのペアを (o_i, o_j) とすると $ediam(O)$ は以下のように表すことができる．

$$ediam(O) = \frac{diam(O)}{score(o_i) + score(o_j)}$$

4.2　スコア付き Pairwise Expansion の提案

4.2.1　データ格納方法

スコア付き PE 法では，Q が与えられたときに必要なデータ集合をデータベースからロードして四分木を作成する．四分木の各ノードには，最小包囲直方体（MBR）とノードに格納されているデータの最大スコアを与えた．ノードが持つことができる最大データ数は 50 とし，最大データ数よりもデータ数が多いときは，各データがもっている緯度経度の平均をとり子ノードに分割した．

4.2.2　スコア付き Pairwise Expansion の計算方法

葉ノード 2 つの組み合わせ (ノードペア) を作り, 以下の規則の基づいて枝刈りする．残った葉ノードの組み合わせから直径となるデータ 2 点の組み合わせ (o_1, o_2) を ediam が昇順に列挙し，Q を満たすように $shuttle(\langle o_1, o_2 \rangle)$ 内でオブジェクト集合 O を見つける．最後に直径が同じ O について，一番スコアの合計が高いもの O' を代表解とする．

規則 1　ノードペア $(\langle N_i, N_j \rangle)$ と探索するノードを合わせて入力されたキーワードを満たさなければならない．

規則 2　k 番目の解の実効直径 $ediam(O_k)$ を δ とし，ノードペアがとりうる最大実効直径 $mindist(N_1, N_2)$ が δ より小さくならなければならない．

4.3　スコアの付け方

従来方法における問題点である解が 2，3 カ所にかたまる問題について，解がより散らばるように以下 2 つの手法を提案する．

- 葉ノードの深さによって点数をつける
- 葉ノードの面積と格納されているデータの個数によって点数をつける

以下にて二つの手法について詳しく説明する．

4.3.1　葉ノードの深さによるスコア付け法

一般に，ノードが深くなればなるほどノードがもつ MBR は小さくなり，またデータが密集しているので解が出現しやすい．そのためノードが深くなればなるほどそのノードが持つスコアを相対的に低くすることで，浅いノードにおいても実効直径が小さくなり，解が出現しやすくなると考えられる．よって，葉ノード l がもつデータのスコアを，l の深さを d として以下のように計算する．

$$score_1(o) = \frac{1}{f^d} \qquad (f = 4) \tag{1}$$

データを四分木に格納しているため f=4 とした．

4.3.2　葉ノードの面積とデータの個数によるスコア付け法

データの緯度経度の重心を取る四分木でバランス木になったとき，葉ノードの深さによるスコア関数は機能しない．そのためノードの深さではなく MBR の面積に着目する．ノードが持てる最大のデータ数は決まっているため，ノードが広いほど各データはちらばりやすい．各ノードにおける密度を計算することで，密度が低いノードにおいても解が出現しやすくなると考えられる．よって，ある葉ノードがもつデータのスコアを，根ノードの MBR の面積を 1 としたとき，葉ノード N の MBR の相対面積を a として以下のように計算する．

$$score_2(o) = \frac{a}{count(N)} \tag{2}$$

ここで相対面積を用いた理由として，スコア関数 1 における根ノードのスコアが 1 であるためである．

5　評価実験

3 章で述べた問題点が提案した手法によって改善されたか評価実験を行った．データセットとして Flickr に 2013 年 1 月 1 日から 2016 年 3 月 1 日までの期間に登録された東京周辺の写真の集合 (約 25 万件) を用いた．計算には，linux の corei7 の CPU3.20GHz，メモリ 8GB の計算機を用いた．

ここで Top-k mCK 検索の解法として用いた手法は，allans(従来の解法 2.2 節)，noscore(allans において直径 1 つ当たりの mCK 解を一つだけ求めて次の異なる直径を探すように制限した方法)，score1(4.3 節の score1 を noscore 方式に適応したもの)，score2(4.3 節の score2 を noscore 方式に適応したもの) である．

5.1　解が局所的に集まる問題について

解が局所的に固まる問題が解決できたか確認するために，Q={sakura,river,temple} としてそれぞれのスコア関数を用いて評価実験を行った．図 5-8 は Top-100mCK 検索したときの東京周辺を拡大したものである．赤点が sakura，緑点が river，青点が temple を表していて，ediam が小さい順に青い数字で解を表している．

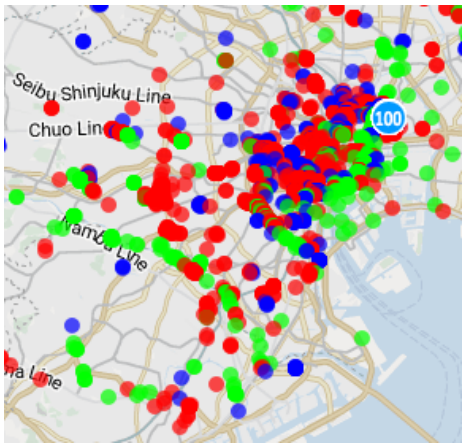

図 5　allans の上位 100 解

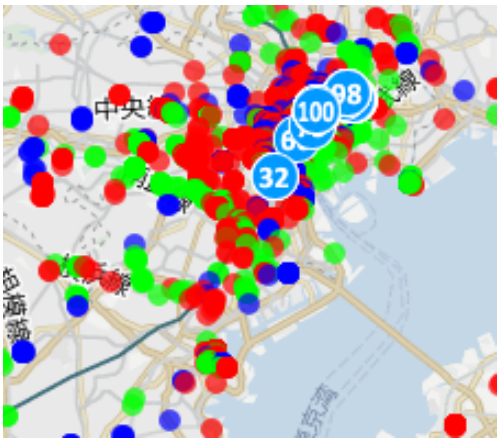

図 6　noscore の上位 100 解

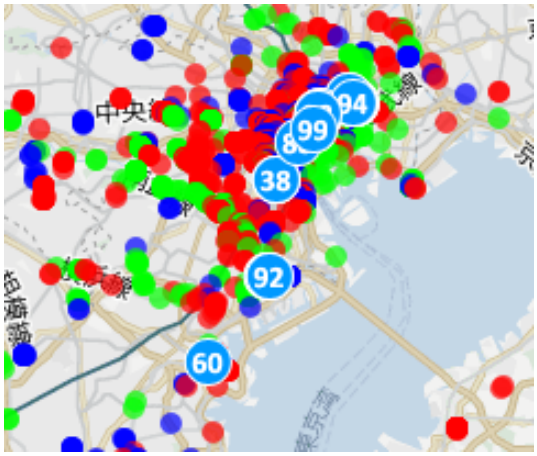

図 7　score1 の上位 100 解

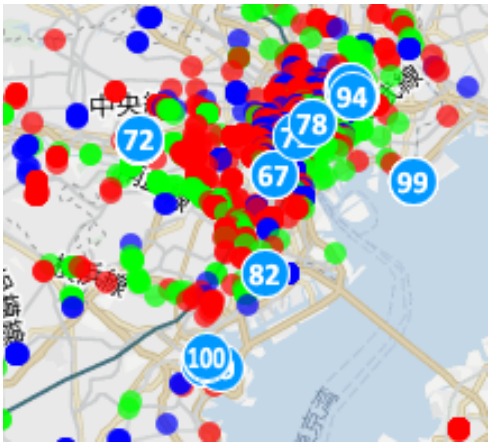

図 8　score2 の上位 100 解

これらの結果からスコア関数によって多様性の値に効果があったと考えられる．このことを数学的に確認する方法として従来方法と提案手法について，相異なる解の直径となる 2 端点の中点同士の距離の平均値 (diversity) を計算した．この数値が大きいほど解同士が離れているといえる．

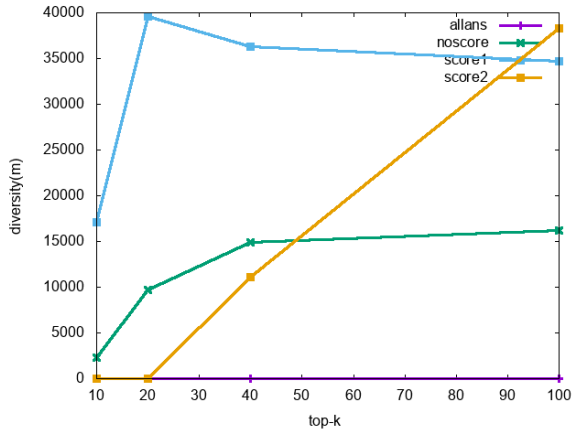

図 9　それぞれのスコア関数における diversity

図 9 は，上位 10，20，40，100 解におけるそれぞれのスコア関数を用いたときの diversity を表している．allans は従来解法における結果を表していて，解がほぼ一点に集中していることが確認できる．score1，score2 の提案は，上位 40 解を超えると noscore よりも多様性が有効であり，スコア導入の効果が一定数見られる．図 5-8 からも分かるように上位 100 解では，score2 を用いたときが一番解が散らばっている．しかし，score2 は K が 40 以下で noscore より多様性が低く，この原因を調べる必要がある．

6　まとめ

本稿では，従来の $m$ck 検索問題にみられた局所的に解が固まる問題について各データあたり空間的なデータの良さやバラつきを表すスコアを与えることによる解決を試みた．提案した問題設計に基づいて計算した結果，従来の問題設定より多様性の数値が上昇していることから，スコア関数を用いることによって解の幅を広げることができた．しかし，与えるキーワードや解の個数によってはスコア関数が期待したようにうまく働かない場合があるため，その原因を解析し改良すること，idf 値などの新しいスコア関数を作成しその効果を確かめることを次回の課題としたい．

参考文献

[1] D. X. Zhang, Y.M. Chee, A. Mondal, A.K.H. Tung, and M. Kitsuregawa, ”Keyword search in spatial databases: Towards searching by document,” IEEE ICDE, pp.688-699, 2009.
[2] T.Guo,X.Cuo,G.Cong”Efficient Algorithms for Answering the m-closest Keywords Query,” ACM SIGMOD, pp.405-418, 2015.
[3] Y.Qiu, T.Ohmori, T.Shintani,H.Fujita, ”Pairwise Expansion:A new Topdown Search for mCK Queries Problem over Spatial Web,” APweb, pp.459-463 2016
[4] Y.Qiu,X.Hei, T.Ohmori,H.Fujita, ”An Object-pair Driven Approach for Top-k mCK Query Problem by Using Hilbert R-tree,” TrustCom/IEEE Int.Conf BigDataSE 2019