モバイルセンサデータベースにおける効率的なTop-k検索結果の多様化について

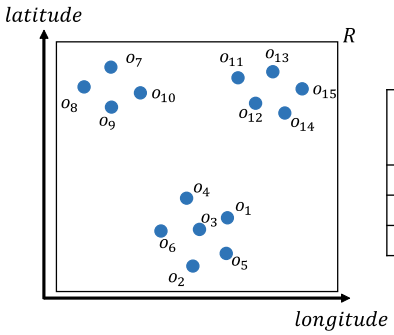
横山 正浩[1,a)]　原 隆浩[1,b)]　西尾 章治郎[1,c)]

概要：近年のセンサ技術の発展に伴い，スマートフォンなどのモバイルセンサ端末が普及しており，これらの端末から収集したモバイルセンサデータを活用する研究に注目が集まっている．時空間上に存在するモバイルセンサデータの中から，ユーザが興味を持つデータを地理的に偏りなく取得するために，Top-k検索結果の地理的多様性を考慮することが有効である．しかし，興味はユーザごとに異なるため，クエリ毎にデータのスコアを計算しなければならない．このような単純な手法における，大量のモバイルセンサデータに対するTop-k検索結果の多様化処理は，計算コストが大きい．本研究では，事前にモバイルセンサデータに対しクラスタリング処理を施すことにより，Top-k検索結果の多様化処理を高速化する手法を提案する．単純な手法では，検索結果として最適なデータを一つずつ探索するために，その度に検索範囲の全てのデータを走査する必要があるが，提案手法ではクラスタリング情報に基づき走査するデータを削減することで，高速に検索結果が得られる．さらに本稿では，提案手法の性能をシミュレーション実験によって検証する．

1. はじめに

近年，スマートフォンを始めとして，様々なセンサデバイスを搭載したモバイル端末が広く普及している．これに伴い，モバイル端末の取得したセンサデータを利活用する研究に注目が集まっている．このようなセンサデータは，時刻や位置情報のほか，気温や音といったセンシングした物理現象に関する属性値を有する．センサデータを蓄積したデータベースから，ユーザが調査したい時空間範囲のデータを取得することで，環境モニタリングなどのアプリケーションに利用できると考えられる [4], [9].

単純な時空間範囲検索では，ユーザにとって必要のないデータが検索結果として多く含まれ，多くの場合利便性に欠ける．そのため，分析の目的に応じたユーザの注目する属性（気温や湿度，騒音指数と大気汚染指数など）によりデータを順位付けし，上位 k 個のデータを取得する Top-k 検索が有用であると考えられる [11]．これにより，ユーザはその時の分析目的に応じた，有用であると考えられるデータを取得できる．しかし，一般的に環境情報は，時空間的に近くのデータは似たような属性値を取る確率が高い傾向（時空間的相関性）を有している．そのため，広域かつ細粒度に収集されたモバイルセンサデータは似たようなデータが時空間的に集中する傾向がある．例えば，図 1 に示すようなデータ分布で，騒音指数と大気汚染指数の属性値から算出されるスコアが高いデータを範囲 R 内から検索する場合，上位 6 個のデータは特定のグループ（$o_1 \sim o_6$）からしか取得できない．図中の他のデータも，$o_1 \sim o_6$ とほとんど変わらない高い属性値をとっており分析対象となりうるが，これらのデータを取得するためには十分に k を大きく設定する必要がある．しかし，ユーザはモバイルセンサデータの時空間的な分布を事前に知り得ないため，適切な k を設定するのは困難である．



Data	Air pollution level + Noise level
$o_1 \sim o_6$	[0.90, 0.9)
$o_7 \sim o_{10}$	[0.89, 0.90)
$o_{11} \sim o_{15}$	[0.88, 0.89)

図 1　データの空間分布およびデータの属性値

そこで，地理的に偏りのない結果を取得するために，検索結果のデータがなるべく類似しないようにスコア上位のデータを取得する，Top-k 検索結果の多様化処理が有効で

[1] 大阪大学大学院情報科学研究科マルチメディア工学専攻
[a)] yokoyama.masahiro@ist.osaka-u.ac.jp
[b)] hara@ist.osaka-u.ac.jp
[c)] nishio@ist.osaka-u.ac.jp

あり，多くの研究がなされている [3], [5], [7]．多様化処理では，グリーディアルゴリズムによって，初期化された正解集合に対し逐次最適なデータを追加し，大きさが k になるまで繰り返す．最適なデータは，ユーザが注目する属性値から算出されるデータのスコアと，データ間の空間距離を考慮して選択される．しかし，単純なグリーディアルゴリズムでは，最適なデータを選択するために正解集合に追加されたデータ以外のすべてのデータを走査する必要があり，全体データセットや k が大きい場合に計算コストが大きくなる．

本稿では，モバイルセンサデータの時空間的相関性に着目した事前クラスタリング処理によって，Top-k 検索結果の多様化処理を高速化する手法を提案する．提案手法では，時空間および注目する属性値に関して近接しているデータをクラスタ化し，クラスタごとに代表データを決める．データ走査フェーズでは，各クラスタの代表データのみを走査し，代表データとクラスタ半径の情報から，クラスタ内のデータに関して取りうる評価値の上界を推定する．時空間および属性値の両面から近接性を考慮してクラスタリングすることで，クラスタ内のデータが取りうる評価値の上界を出来る限り小さく，短時間で計算できる．推定された評価値が十分に小さいクラスタ内のデータを走査対象から除外することで，最適なデータを短時間で探索できる．結果として，単純なグリーディアルゴリズムよりも走査するデータ数を削減しつつ，全く同じ正解集合を取得できる．シミュレーション実験の結果から，提案手法の有効性を確認した．

以下では，2 章で想定環境を紹介し，本稿の問題を定義する．3 章で単純なグリーディアルゴリズムによるベースライン手法を紹介し，4 章で提案手法について説明する．5 章でシミュレーション実験の結果を示し，6 章で関連研究について述べる．最後に 7 章で本稿をまとめる．

2. 想定と問題定義

データモデル． データ $o \in O$ は，データ ID $o.id$，観測時刻 $o.t$，位置情報 $o.\boldsymbol{loc}$，属性値 $o.\boldsymbol{att}$ を保持している．$o.\boldsymbol{loc}$ は，緯度と経度によって表される 2 次元平面内の点とし，$o.\boldsymbol{att}$ は d 次元のベクトル $o.att_i (i = 1, ..., d)$ で表される．各データのスコアは，クエリ q に基づいて決定され，スコアが大きいほどそのデータは正解集合に含まれやすい．クエリ q は，重み付け係数 $q.\boldsymbol{w}$ を有しており，クエリ q における，データ o のスコア $p(q, o)$ は，以下の式に従って計算される．

$$p(q, o) = \sum_{i=1}^{d} q.w_i \cdot o.att_i \tag{1}$$

また，$d(u, v)$ は，データ u, v の位置情報に基づくユークリッド距離で，

$$d(u, v) = \sqrt{(u.loc_x - v.loc_x)^2 + (u.loc_y - v.loc_y)^2} \tag{2}$$

とする．このスコアリング関数（以降は，可読性を考慮して $p(o)$ と略記），およびデータ間の位置情報から算出されるユークリッド距離に基づいて，Top-k 検索結果の多様化を以下のように定義する．

定義（Top-k 検索結果の多様化）．
クエリ $q = \{R, k, \lambda, \boldsymbol{w}\}$ が与えられた時，データ集合 O を時空間範囲に観測されたデータ集合 $O = \{o_i \mid o_i \in q.R\}$ とする．この時，以下の式で与えられる部分集合 S_k^* を最適な検索結果とする．

$$S_k^* = \underset{S_k \subseteq O, |S_k| = k}{\arg\max} f(S_k, q, p(\cdot), d(\cdot, \cdot)) \tag{3}$$

ここで，$f(S_k, q, p(\cdot), d(\cdot, \cdot))$ は目的関数（以降は，可読性を考慮して $f(S_k)$ と略記）で，検索結果の多様化を実現するために，様々なものが提案されている．本研究では，高いスコアをとるデータを優先しながらも空間的に偏りの小さい結果が得られる，文献 [6] における Max-Min 問題を対象とする．この場合，目的関数は以下の式で与えられる．

$$f(S) = \min_{u \in S} p(u) + \lambda \min_{u, v \in S} d(u, v) \tag{4}$$

式 (4) から，正解集合内のデータに関してスコアの最小値が大きいほど目的関数の値は大きくなり，また，正解集合内の任意のデータ間の距離について，その最小値が大きいほど目的関数の値は大きくなる．式 (4) 中の λ は，ユーザの検索における地理的多様性についての重要性を表しており，λ が大きいほど地理的多様性を重視してデータを要求し，地理的により分散した結果が得られる．特に，$\lambda = 0$ のときはデータのスコアしか考慮されないため，正解集合は純粋な Top-k 検索結果と等しくなる．

しかし，上記の組合せ最適化問題を解くことは，NP 困難であることが示されており，検索範囲内のデータ数 N が大きい時に全ての部分集合候補について総当りで探索するのは，計算時間の観点から現実的ではない．そこで，近似解を求めるグリーディアルゴリズムが，様々な目的関数に応じて提案されており，本稿における提案手法もグリーディアルゴリズムを基本とする．

3. ベースライン手法

ベースラインとなる単純なグリーディアルゴリズムを，アルゴリズム 1 に示す．1，2 行目の初期化処理は，スコアの高いデータがユーザの検索したいデータであるため，全データ内で最大のスコアをとるデータを正解集合に追加することとした [5]．3 行目の反復により，正解集合の大きさが k となるまで，データを正解集合に追加する．4 行目について，$d'(\cdot, \cdot)$ は互いのデータのスコアとデータ間の空間距離を考慮した特殊距離で，以下の式で定義される．

Algorithm 1 Algorithm for Max-Min Problem

Input: Data set $O, k, \lambda, \boldsymbol{w}$

Output: Set $S(|S| = k)$ that maximizes $f(S)$

1: Initialize the set $S = \emptyset$
2: Find $u = \arg\max_{x \in O} p(x)$ and set $S = \{u\}$
3: **while** $|S| < k$ **do**
4: Find $x \in O \backslash S$ such that $x = \arg\max_{y \in O \backslash S} d'(y, S)$
5: Set $S = S \cup \{x\}$
6: **end while**

表 1 記号の概要

記号	意味
$p(\cdot)$	データのスコア
$d(\cdot, \cdot)$	データ間の空間距離
$d'(\cdot, \cdot)$	データ間の特殊距離
S	正解集合（多様化結果）
$d'(\cdot, S)$	データの評価値
$f(S)$	目的関数
o_{rep}	クラスタ代表データ
o_{cen}	クラスタ中心データ

$$d'(u, v) = \frac{1}{2}(p(u) + p(v)) + \lambda d(u, v) \qquad (5)$$

$d'(\cdot, \cdot)$ は，$x = y$ のとき $d'(x, y) = 0$ と定義すれば，距離の公理を満たす．また，アルゴリズム中の $d'(y, S)$ は，任意の正解集合に含まれないデータ $y \in O \backslash S$ と，正解集合との間の特殊距離を示す．ここで，データとデータ集合との距離は，集合内に含まれるデータ全てとの距離を計算した時の最小値とし，以下の式で定義される．

$$d'(y, S) = \min_{u \in S} d'(y, u) \qquad (6)$$

正解集合に含まれる任意の 2 データ間の特殊距離の最小値は，以下に示すように，式 (4) で定義した目的関数と一致する．

$$\min_{u,v \in S} d'(u, v) = \min_{u \in S} p(u) + \lambda \min_{u,v \in S} d(u, v) = f(S) \quad (7)$$

これにより，正解集合との特殊距離 $d'(\cdot, S)$ が最大となるデータを探索することで（4 行目），保持している状態に関して目的関数を最大化する最適なデータを選択できる．以降では特に，任意のデータ x と正解集合との特殊距離 $d'(x, S)$ を，データ x の評価値と呼ぶ．

このアルゴリズムの計算量は，全体のデータ量 N に依存する．初期化処理は，データのスコアが最大のデータを探索するため，単純に全データの走査が必要となり，計算量は $O(N)$ である．また，3 行目の反復については，反復回数が k，各反復につき最大 $k(N - k)$ 回の特殊距離計算が必要となるため，全体の計算量は $O(k^2 N)$ となる．そのため，データサイズが大きくなると計算時間が長くなってしまう．

Algorithm 2 Algorithm for Clustering Data

Input: Data set O, Spatial range r_1, Attribute range r_2

Output: $\boldsymbol{C} = C_1, C_2, ..., C_k$ Set of clusters

1: clusterLabel = 1
2: **for** $i = 1$ to N **do**
3: **if** o_i is not in any clusters **then**
4: Mark o_i as the center and initial representative of the current cluster
5: X = retrieveNeighbors(o_i, r_1, r_2)
6: **for** $j = 1$ to $|X|$ **do**
7: Mark all objects in X with current clusterLabel
8: **end for**
9: clusterLabel++
10: **end if**
11: **end for**

そこで，本稿では走査するデータ数を削減し，かつベースライン手法と同じ正解集合を取得する手法を提案する．表 1 は，本稿で用いる記号の概要を示す．

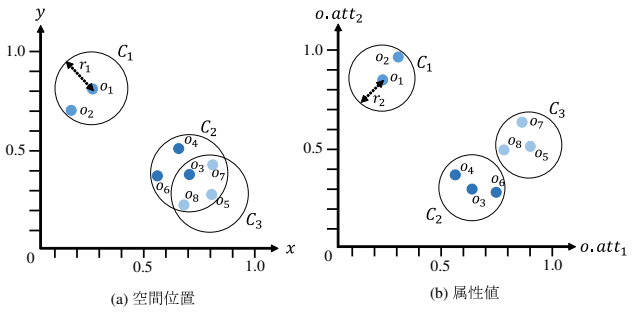
4. 提案手法

本章では，本稿の提案手法について説明する．ベースライン手法では，アルゴリズム中の各反復で最適データを探索するために，正解集合に含まれない全データを走査する必要があり，計算時間が長くなる．しかし，空間的に近くに存在するセンサデータは，時空間的相関性（spatio-temporal correlation）により互いに似た属性値を観測する可能性が高い [12]．そのようなデータは，ユーザの属性値に対する関心が異なる（スコアリング関数が異なる）場合でも，互いに似たスコアをとる．また，空間的に近くに存在するため，正解集合との空間距離も近い値となる．よって，空間的に近いデータは，評価値も互いに似た値となる可能性が高い．そこで，空間的に近いデータをクラスタ化し，クラスタ中心のデータをクラスタの中心データおよび初期代表データとする．クエリ処理では，最初に全クラスタの中心データと代表データについてのみ，評価値を計算する．中心データの評価値が十分に小さい場合，そのクラスタ内のデータについて評価値を計算する必要はなく，走査するデータ数を削減できる．

まず，4.1 節において，クエリ処理で利用するクラスタを作成するための，事前クラスタリング手法について紹介する．次に，4.2 節において，クラスタを用いたクエリ処理アルゴリズムを紹介する．

4.1 事前クラスタリング手法

まず，提案手法で用いるクラスタ作成方法について説明



(a) 空間位置　(b) 属性値

Cluster	Representative (Center)	Member
C_1	o_1	o_1, o_2
C_2	o_3	o_3, o_4, o_6
C_3	o_5	o_5, o_7, o_8

図 2　クラスタリング例

する．具体的なクラスタリングアルゴリズムを，アルゴリズム 2 に示す．3，4 行目で，いずれのクラスタにも属していないデータを見つけた場合，そのデータを新たなクラスタの代表データかつ中心データとする．ここで，5 行目の retrieveNeighbors(o_i, r_1, r_2) は，データ o_i の空間位置を中心とした半径 r_1 の円内に存在し，かつ，データ o_i のセンサ属性値を中心とした半径 r_2 の超球内に存在するデータを返す操作である．d 次元の属性値について，超球内に存在するデータ集合 O_i は，ユークリッド距離を用いた以下の式で表される．

$$O_i = \{o \mid \sqrt{\sum_{j=1}^{d}(o_i.att_j - o.att_j)^2} \leq r_2\} \qquad (8)$$

ここで，空間位置だけでなく，属性値についてもデータの近接性を考慮するのは，以下の理由による．センサデータは，互いに空間距離が近くても，観測した属性値に誤差を含んでいる場合や，観測時刻が他のデータと離れている場合がある．そのようなデータは，空間位置は近いが，属性値に差が生じることでスコアが大きく異なり，互いに評価値も大きく異なる可能性がある．属性値に関して近接性を考慮することで，このようなデータを互いに別々のクラスタに分割できる．

中心データから近くに存在するデータの集合は，R 木 [1] などの多次元インデックス構造を用いた範囲検索により，効率的に取得できる．クラスタ間でのデータの共有はないものとし，全てのデータがいずれかのクラスタに割り当てられるまでクラスタを生成する．

例． 図 2 を用いて，属性数 $d = 2$ の場合の具体例について説明する．まず，データ o_1 を中心としてクラスタ C_1 を生成する．データ o_2 はデータ o_1 を中心とした空間距離 r_1 の円内に存在し，かつ o_1 を中心とした属性値空間で半径 r_2 の円内に存在するため，同じクラスタに割り当てる．次に，データ o_3 を中心としてクラスタ C_2 を生成する．残りの未割り当てのデータは，全て o_3 を中心とした空間距離

Algorithm 3 Algorithm for Max-Min Problem via Clusters

Input: $\boldsymbol{C}, k, \lambda, \boldsymbol{w}, r_1, r_2$

Output: Set $S(|S| = k)$ that maximizes $f(S)$

1: Initialize the set $S = \emptyset$
2: Find $u = \arg\max_{x \in O} p(x)$ and set $S = \{u\}$
3: **while** $|S| < k$ **do**
4: Find $\overline{o_{rep}}$ such that $\overline{o_{rep}} = \arg\max_{o_{i,rep} \in C_i} d'(o_{i,rep}, S)$
5: Initialize the set $\boldsymbol{C'} = \{C \mid \overline{o_{rep}} \in C\}$
6: **for all** $i = 1$ to $|\boldsymbol{C}|$ **do**
7: Estimate upper bound of each cluster $\overline{d'(C_i, S)} = \max_{v_i \in C_i} d'(v_i, S)$
8: **if** $d'(\overline{o_{rep}}, S) \leq \overline{d'(C_i, S)}$ **then**
9: $\boldsymbol{C'} = \boldsymbol{C'} \cup \{C_i\}$
10: **end if**
11: **end for**
12: Find $x \in \boldsymbol{C'} \backslash S$ such that $x = \arg\max_{y \in \boldsymbol{C'} \backslash S} d'(y, S)$
13: Set $S = S \cup \{x\}$
14: **if** x is representative data of C_i **then**
15: Select new representative data for C_i
16: **end if**
17: **end while**

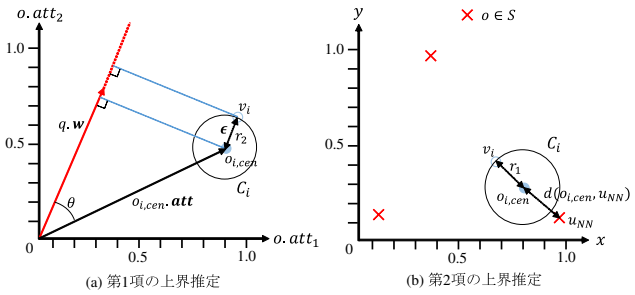
r_1 の円内に存在する．しかし，属性値空間では，o_3 を中心とした半径 r_2 の円内に存在するのはデータ o_4 と o_6 の 2 つのみであり，これらをクラスタ C_2 に割り当てる．同様に，データ o_5 を中心としてクラスタ C_3 を生成すると，残りのデータ o_7，o_8 はクラスタ C_3 に割り当てられ，結果，全データは 3 つのクラスタに割り当てられる．

4.2　クラスタを用いたクエリ処理

次に，クラスタを利用したクエリ処理について説明する．具体的なアルゴリズムを，アルゴリズム 3 に示す．ここでは特に，走査データ数削減の要点である，アルゴリズム 3 の反復部分について説明する．

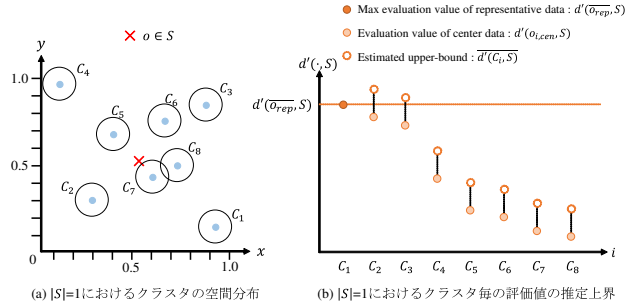
まず，全クラスタの代表データを走査し，評価値の最大値をとる代表データ $\overline{o_{rep}}$ を探索する（4 行目）．またこの時に，中心データの評価値も計算し記憶しておく．これは，本節の後半で説明する，クラスタ内データが取りうる評価値の上界推定のときに必要となるためである．評価値が最大値をとる代表データを含むクラスタは，正解集合に追加する最適なデータを含む可能性があるため，走査対象クラスタ集合 $\boldsymbol{C'}$ に追加する．

次に，全クラスタについて，各クラスタ内のデータの評価値が取りうる値の上界 $\overline{d'(C_i, S)}$ を推定する（7 行目）．上界の推定値と，最初に計算した代表データの最大評価値を比較し，推定値のほうが大きい場合，$\boldsymbol{C'}$ に追加する．一方，推定値のほうが小さい場合，少なくともこのクラスタ



(a) 第1項の上界推定　(b) 第2項の上界推定

図 3　クラスタ内データがとる評価値の上界推定



(a) $|S|$=1におけるクラスタの空間分布　(b) $|S|$=1におけるクラスタ毎の評価値の推定上界

図 4　クラスタリング例

内のデータよりも，最大評価値をとる代表データ $\overline{o_{rep}}$ のほうが正解集合に追加するデータとして適しているため，以降の走査からは除外する．

最後に，走査対象クラスタ集合に含まれるデータを全走査し，最大の評価値をとるデータを正解集合に追加する．追加されたデータがいずれかのクラスタの代表データだった場合，クラスタ内のデータからランダムに新たな代表データを選択する（15 行目）．

評価値の上界の推定． アルゴリズム 3 の 7 行目の，各クラスタ内データが取りうる評価値の上界推定について説明する．クラスタが含むデータの詳細は不明なため，クラスタ内に存在しうる仮想的なデータ v_i を考え，データ v_i が取りうる最大の評価値を計算する．

評価値は式 (5) および式 (6) より，空間距離と属性値に基づくスコアの 2 つの指標から算出される．ここで，評価値を正解集合内のデータに非依存の項と依存する項に分解する．

$$d'(y, S) = \frac{1}{2}p(y) + \min_{u \in S}(\frac{1}{2}p(u) + \lambda d(y, u)) \qquad (9)$$

まずは，正解集合内のデータに非依存の項（第 1 項）が取りうる最大値を計算する．スコアは式 (1) から，重みベクトル $q.\boldsymbol{w}$ と属性値ベクトル $o.\boldsymbol{att}$ の内積として捉えると，それぞれのベクトルのなす角を θ とした時，以下の式で計算できる．

$$\begin{aligned} p(q, o) &= \sum_{i=1}^{d} q.w_i \cdot o.att_i \\ &= q.\boldsymbol{w} \cdot o.\boldsymbol{att} \\ &= |q.\boldsymbol{w}||o.\boldsymbol{att}|cos\theta \end{aligned} \qquad (10)$$

よって，クラスタ内に存在しうる仮想データ $v_i (v_i.\boldsymbol{att} = o_{i,cen}.\boldsymbol{att} + \boldsymbol{\epsilon})$ を考えた時，第 1 項の最大値は以下の式で与えられる（図 3(a)）．

$$\begin{aligned} &\max_{v_i \in C_i}(\frac{1}{2}p(v_i)) \\ &= \max_{|\boldsymbol{\epsilon}| \le r_2, 0 \le \theta \le 2\pi}(\frac{1}{2}(q.\boldsymbol{w} \cdot (o_{i,cen}.\boldsymbol{att} + \boldsymbol{\epsilon}))) \\ &= \frac{1}{2}(p(o_{i,cen}) + \max_{|\boldsymbol{\epsilon}| \le r_2, 0 \le \theta \le 2\pi}(|q.\boldsymbol{w}||\boldsymbol{\epsilon}|cos\theta)) \\ &= \frac{1}{2}(p(o_{i,cen}) + |q.\boldsymbol{w}|r_2) \end{aligned} \qquad (11)$$

次に，正解集合内のデータに依存する項（第 2 項）が取りうる最大値を計算する．まず，クラスタ中心のデータと正解集合との特殊距離を計算し，その時の正解集合内のデータを $u_{NN} \in S$ とする．ここで，データ u_{NN} から見た時，円内に存在しうるデータで，最も空間距離が大きくなるのは，データ u_{NN} と円の中心を結んだ直線との交点で，データ u_{NN} から遠い方の点である（図 3(b)）．そのため，第 2 項が取りうる最大値は以下の式で与えられる．

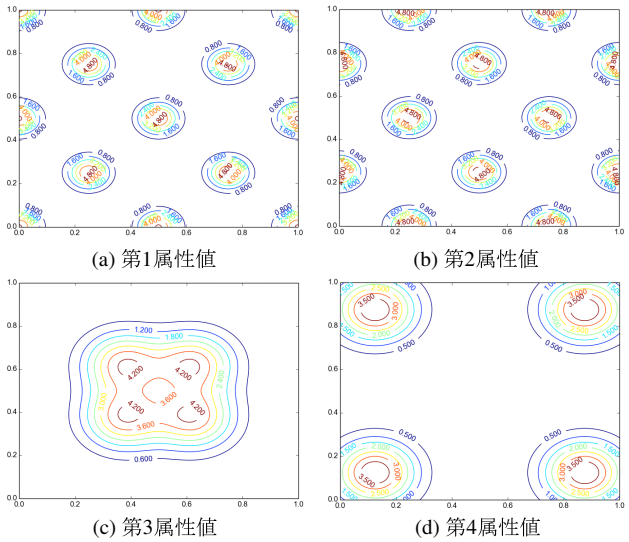
$$\max_{v_i \in C_i}(v_i, S) = \frac{1}{2}p(u_{NN}) + \lambda(d(o_{i,cen}, u_{NN}) + r_1) \qquad (12)$$

式 (11) および式 (12) の和が，各クラスタ内データが取りうる評価値の上界の推定であり，以下の式で示される．

$$\begin{aligned} \overline{d'(C_i, S)} &= \frac{1}{2}(p(o_{i,cen}) + |q.\boldsymbol{w}|r_2) \\ &\quad + \frac{1}{2}p(u_{NN}) + \lambda(d(o_{i,cen}, u_{NN}) + r_1) \\ &= \{\frac{1}{2}(p_{o_{i,cen}} + p(u_{NN})) + \lambda d(o_{i,cen}, u_{NN})\} \\ &\quad + \frac{1}{2}|q.\boldsymbol{w}|r_2 + \lambda r_1 \\ &= d'(o_{i,cen}, S) + \frac{1}{2}|q.\boldsymbol{w}|r_2 + \lambda r_1 \end{aligned} \qquad (13)$$

式 (13) より，上界の推定は，クラスタの中心データの評価値とクラスタ半径およびクエリ情報のみで単純に計算でき，計算コストは小さい．また，1，2 行目の初期化処理についても，式 (11) からクラスタ内データの取りうるスコアの上界が推定できるため，同様の手順で走査データ数を削減できる．

例． 走査データ数を削減する具体例を示す．図 4(a) は $|S| = 1$ で，評価値が最大の代表データ $\overline{o_{rep}}$ および各クラスタの中心データの評価値は計算済みの状態である．ここ



(a) 第1属性値　(b) 第2属性値

(c) 第3属性値　(d) 第4属性値

図 5　実験データの属性値

ではわかりやすくするために，評価値最大の代表データを含むクラスタを C_1，それ以外のクラスタ ID を中心データの評価値の降順に割り当てている．各クラスタの中心データの評価値から推定した評価値の上界と，代表データの最大評価値の関係が図 4(b) のようになったとする．この時，図中の破線を上界が下回っているクラスタ $C_4 \sim C_8$ は，$\overline{o_{rep}}$ よりも評価値の高いデータを含み得ないため，走査する必要はない．

5. 評価実験

Top-k 検索結果の多様化処理における，提案手法の性能を評価する．

データセット．データの位置情報を，1.0×1.0 の領域に 2 次元の一様分布で生成した．また，データの属性値は，図 5 のような分布を想定した．データの属性値はその位置情報から決定されるが，センシング誤差として平均 0，標準偏差 0.3 の正規乱数を加算した値をデータの属性値として与えた．

比較手法．比較手法として，3 章で説明したベースライン手法，および空間位置のみを考慮したクラスタを利用したクエリ処理手法を用いた．この比較手法は，クエリ処理のアルゴリズムは提案手法と同じだが，クラスタ内のデータのスコアについて，上界を推定するための情報を持たない．そのため，一度全てのデータのスコアを計算し，クラスタ毎に最大のスコアとそのデータを記憶する．以降，評価値の上界推定にはこの最大のスコアを用いることで，他の手法と全く同じ正解集合が取得できる．

設定．全てのアルゴリズムを Java7 で実装し，Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz with 24.0 GB RAM を搭載する Ubuntu 14.04 で動作する計算機によって実験した．

実験においては，センサデータおよびクラスタデータをメモリに読み込んだ時点から，検索結果を取得するまでの計算時間を測定した．ランダムなクエリ（$q.\boldsymbol{w}$ と $q.\lambda$ が一定範囲内でランダム）を 100 個生成し，計算時間の平均値を調べた．また，比較手法と提案手法については，事前クラスタリング処理によって生成されたクラスタ数も示した．表 2 はパラメータを示し，太字はデフォルト値とする．

表 2　パラメータ設定

パラメータ	値
データ数 N	**100K**, 500K, 1M
要求データ数 k	5, 10, **15**, 20, 25
クラスタの空間半径 r_1	0.01∼0.5
クラスタの属性値半径 r_2	0.1∼2.0
λ	0.0∼5.0
$\boldsymbol{w}$ の各要素	0.0∼1.0
属性の次元数 d	1, **2**, 3, 4

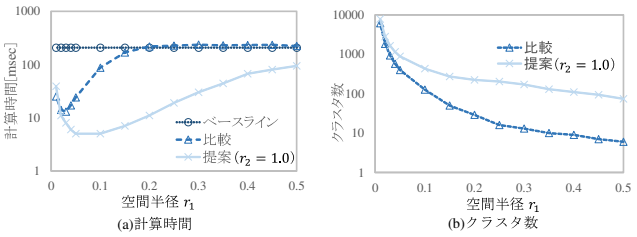


図 6　空間半径 r_1 の影響

5.1 空間半径 r_1 の影響

まず，クラスタの空間半径 r_1 を変化させた場合の，計算時間を図 6(a) に，クラスタ数を 6(b) に示す．図 6(a) から，空間半径が大きい場合，比較・提案手法ともに計算時間が長くなっていることがわかる．これは，クラスタ内データが広い地理範囲に存在しうるため，クラスタ内データの評価値の上界を過剰に大きく推定しており，走査対象から除外できたクラスタ数が少ないためである．このことは，属性値が急激に変化している領域におけるクラスタで顕著である．特に，$0.2 \leq r_1 \leq 0.5$ の場合の比較手法は，クラスタの中心データおよび代表データの余分な評価値計算が必要なため，ベースライン手法よりも計算時間が長くなっている．一方，空間半径が小さい場合も，比較・提案手法ともに計算時間が長くなっていることがわかる．これは，図 6(b) に示されるように，空間半径が小さくなると生成されるクラスタ数が増加し，走査が必要なデータの数も増加してしまうためである．

比較・提案手法における最短の計算時間は，それぞれ 13 ミリ秒，5 ミリ秒で提案手法は比較手法よりも計算時間を短縮している．比較手法は中心データから空間半径の円内のデータを全て同一のクラスタに含めてしまうため，誤差により周辺の属性値と大きく異なるデータが含まれる場合

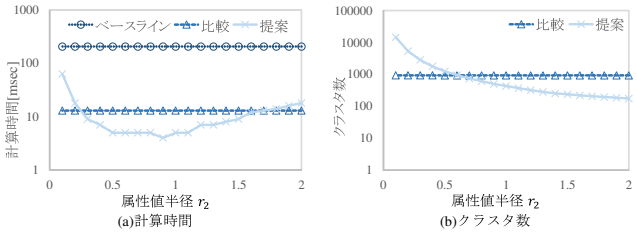


図 7　属性値半径 r_2 の影響

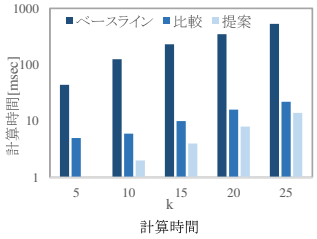


図 9　要求データ数 k の影響

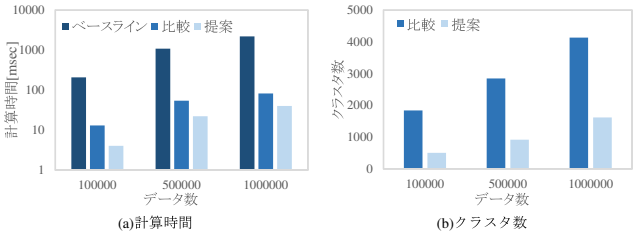


図 8　データ数 N の影響

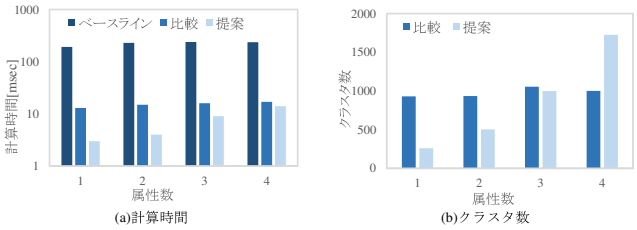


図 10　属性の次元数 d の影響

がある．そのため，クラスタ内にスコアの大きいデータが含まれていた場合，そのデータによってクラスタ内の他のデータを走査対象から除外できず，余分に計算時間を要している．

以降の実験では，比較・提案手法それぞれで計算時間を最短にしていた空間半径（それぞれ 0.03，0.1）を用いている．

5.2　属性値半径 r_2 の影響

次に，クラスタの属性値半径 r_2 を変化させた場合の，計算時間を図 7(a) に，クラスタ数を 7(b) に示す．図 7(a) から，属性値半径が大きい場合，提案手法における計算時間が長くなっていることがわかる．これは，空間半径を大きくした場合と同様に，クラスタ内データのスコアを大きく見積もることで評価値の上界も大きく推定してしまい，走査対象から除外できたクラスタ数が少ないためである．一方，属性値半径が小さい場合も，提案手法における計算時間が長くなっていることがわかる．これも，図 7(b) に示されるように，空間半径を小さくした場合と同様，属性値半径が小さくなると生成されるクラスタ数が増加し，走査が必要なデータの数も増加してしまうためである．

以降の実験では，提案手法において計算時間を最短にしていた属性値半径（0.9）を用いている．

5.3　その他のパラメータの影響

データ数 N の影響．ユーザはセンサデータの地理的分布を事前に知り得ないため，検索範囲内に存在するデータ数が大きい場合にも，クエリ結果を高速に取得できることが重要である．そこで，データ数 N を変化させた場合の，計算時間を図 8(a) に，クラスタ数を 8(b) に示す．いずれのデータ数の場合も，提案手法が計算時間を最短としていることがわかる．比較手法は，走査データ数削減のためにクラスタの空間半径を小さくする必要があり，それに伴いクラスタ数が増加してしまう．一方，提案手法は誤差や属性値の変化を考慮してクラスタを分割できるため，比較手法に比べて少数のクラスタで効率的に走査クラスタ数を削減し，計算時間を短縮できる．

要求データ数 k の影響．ユーザごとに，要求データ数は異なる．そこで，要求データ数 k を変化させた場合の，計算時間を図 9 に示す．いずれの要求データ数の場合も，提案手法が計算時間を最短としていることがわかる．これは，k が大きく，グリーディアルゴリズムの反復が進んでも，提案手法は走査データ数を安定して削減し，計算時間を短縮できるためである．

属性の次元数 d の影響．ユーザごとに，注目するセンサ属性の次元数は異なる．そこで，属性の次元数 d を変化させた場合の，計算時間を図 10(a) に，クラスタ数を 10(b) に示す．いずれの次元数の場合も，提案手法が計算時間を最短としていることがわかる．しかし，提案手法は，次元数が増加するに伴い，計算時間およびクラスタ数が増加している．これは，属性の次元数が増加するに伴い，各属性値のわずかな差で，センサデータ同士の非類似度が大きくなることが原因である．これにより，次元数が大きいほどクラスタ数が増加し，走査データが増加する．

5.4　考察

提案手法の性能は，事前クラスタリング処理における空間半径 r_1 および属性値半径 r_2 に依存する．クラスタ半径が小さい場合，クラスタ内データの評価値の上界は小さく推定できるため，走査対象クラスタからは除外しやすい．一方で，クラスタ半径が小さいことでクラスタ数が増加し，余分な評価値計算が必要となる．よって，5.1，5.2 節の実

験結果が示すように，クラスタ半径には最適値が存在する．現段階では，クラスタ半径はシステム管理者が手動で設定することを想定しているが，クラスタ半径の最適値はデータ分布に依存するため，自動による設定方法は今後の課題の一つである．

6. 関連研究

文献 [10] では，固定センサネットワークにおける Top-k 検索手法を提案している．ユーザが注目する属性についてスコアの高いデータを検索できるが，設置されているセンサはセンシング間隔が同期して動作しており，設置位置も観測範囲に分散している．一方，モバイルセンサは，非同期かつ時空間領域の任意の位置でデータを測定する．データ分布が不明な検索領域から，地理的に偏りなく注目するデータを取得するために，別のアプローチが必要となる．

センサデータ以外のデータについて，Top-k 検索結果の多様化に関する研究は数多く行われている（[3], [8] など）．検索結果とするデータを選ぶ基準として，互いのデータの非類似性の高さを考えることで，検索結果の冗長性を小さくできる．そのため，ユーザの興味が曖昧なキーワード検索の検索結果として，より有用な情報を提供できると考えられている．しかし，これらの研究では単にデータ間の多様性しか考慮されておらず，本稿で目的としている，ユーザが興味をもつデータの取得は実現できない．文献 [5] は，検索対象のデータにユーザの検索に対する関連度（relevance）が付与されており，データ間の多様性を考慮しつつ関連度の高いデータを検索する点で，本稿と類似する．しかし，文献 [5] では，関連度は対象データ全てについて既知とした状態を想定している．本稿の提案手法では，事前クラスタリング処理によって，ユーザの検索が到着する前にクエリ処理効率化のためのデータ構造化がなされているため，ユーザごとの検索に依存せず高速に処理できる．

検索結果の多様化を実現するために，本稿で扱った Max-Min 問題の他にも，Max-Sum 問題や MMR（Maximal Marginal Relevance）などがある（[2], [6]）．これらの最適組合せ問題も，グリーディアルゴリズムで近似解を求めることができ，本稿における提案手法をそれぞれのアルゴリズムにおける評価値に修正することで，効率的に正解集合を計算できる．

7. おわりに

本稿では，モバイルセンサデータベースにおける効率的な Top-k 検索結果の多様化手法を提案した．単純なグリーディアルゴリズムでは，すべてのデータについて評価値を計算し，最適なデータを逐次正解集合に追加する必要が有るため，計算コストが大きい．提案手法では，モバイルセンサデータの時空間的相関性に着目した事前クラスタリング処理を行うことで，グリーディアルゴリズムの評価値が互いに似た値を取るデータをクラスタにまとめる．評価値計算をクラスタ単位で行い，最適なデータを含み得ないクラスタを評価値計算から除外することで，走査するデータ数を削減する．シミュレーション実験から，提案手法は最適データを探索するのに走査するデータ数を削減し，計算時間を短縮できることを確認した．

モバイルセンサデータは，ストリーム環境などデータの更新が頻繁に生じる環境で，多様化検索結果をモニタリングするようなアプリケーションが考えられる．今後の課題の一つとして，これらを正確かつ効率的にモニタリングする手法の検討が挙げられる．

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)(26240013) 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．

参考文献

[1] Beckmann, N., Kriegel, H.-P., Schneider, R. and Seeger, B.: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles, *ACM SIGMOD*, New York, NY, USA, ACM, pp. 322–331 (1990).

[2] Carbonell, J. and Goldstein, J.: The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries, *ACM SIGIR*, ACM, pp. 335–336 (1998).

[3] Drosou, M. and Pitoura, E.: Disc diversity: result diversification based on dissimilarity and coverage, *VLDB*, Vol. 6, No. 1, pp. 13–24 (2012).

[4] EllieD ’Hondt, Stevens, M., Jacobs, A.：Participatory noise mapping works! An evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring, *Pervasive and Mobile Computing*, Vol. 9, No. 5, pp. 681–694 (2013).

[5] Fraternali, P., Martinenghi, D. and Tagliasacchi, M.: Top-k Bounded Diversification, *ACM SIGMOD*, ACM, pp. 421–432 (2012).

[6] Gollapudi, S. and Sharma, A.: An Axiomatic Approach for Result Diversification, *World Wide Web*, ACM, pp. 381–390 (2009).

[7] Khan, H. A., Sharaf, M. A. and Albarrak, A.: DivIDE: Efficient Diversification for Interactive Data Exploration, SSDBM ’14, ACM (2014).

[8] Qin, L., Yu, J. X. and Chang, L.: Diversifying top-k results, *VLDB*, Vol. 5, No. 11, pp. 1124–1135 (2012).

[9] Sazaki, H., Kanzaki, A., Hara, T. and Nishio, S.: SeRAVi: A Spatio-Temporal Data Distribution Visualization System for Mobile Sensor Data Retrieval, *Reliable Distributed Systems Workshops (SRDSW)*, IEEE, pp. 88–93 (2014).

[10] Silberstein, A., Braynard, R., Ellis, C., Munagala, K. and Yang, J.: A Sampling-Based Approach to Optimizing Top-k Queries in Sensor Networks, *ICDE*, pp. 68–68 (2006).

[11] Tao, Y., Hristidis, V., Papadias, D. and Papakonstantinou, Y.: Branch-and-bound processing of ranked queries, *Information Systems*, Vol. 32, No. 3, pp. 424–445 (2007).

[12] Vuran, M. C., Akan, Ö. B. and Akyildiz, I. F.: Spatio-temporal correlation: theory and applications for wireless sensor networks, *Computer Networks*, Vol. 45, No. 3, pp. 245–259 (2004).