モバイルセンサデータベースにおける 階層的クラスタリングを用いたTop-k検索結果の多様化について

横山　正浩† 　原　 隆浩†

† 大阪大学大学院情報科学研究科　〒565–0871 大阪府吹田市山田丘 1-5
E-mail: †{yokoyama.masahiro,hara}@ist.osaka-u.ac.jp

あらまし　近年のセンサ技術の発展に伴い，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され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．例えば，周辺の環境属性値に比べて極端な値を持つデータを取得することで，ホットスポットの地理的分布に関する知識が得られる．このような応用に対して，地理空間上のTop-k 検索結果の多様化が有効である．筆者らはこれまでに，環境情報の空間的自己相関に基いたデータのクラスタリングを利用した，効率的な Top-k 検索結果の多様化手法を提案した．しかし，センシングする環境属性の数が増えると，データ間の類似度の低下に起因してクラスタの数が大幅に増加し，従来手法では計算時間が大きくなってしまう．そこで本稿では，階層的クラスタリングによって環境属性の増加にともなうクラスタ数の増加の影響を軽減可能な，従来手法より効率的な拡張手法を提案する．

キーワード　モバイルセンサデータ，ホットスポット検出，参加型センシング

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．このようなモバイル端末によって，端末保持者付近の物理現象や環境の変化を観測しデータ収集を行う手法は，参加型センシング [8] と呼ばれ，現在注目が集まっている．収集されるデータは，騒音や大気汚染指数といった，観測した現象に関する環境属性値を持った位置情報付きデータである．これらのデータは**モバイルセンサデータ**と呼ばれ，多くの研究者によってモバイルセンサデータを利活用する研究が行われている．

現在，位置情報付きデータに対する検索について，検索範囲を指定し範囲内のデータを取得する，時空間範囲検索が主流である．また，ユーザは多くの場合，町中において高い気温や大気汚染指数を示すデータといった，極端なデータに興味があると考えられる．例えば，図 1 における各円形領域は，周辺の領域に比べてなんらかの環境属性値が大きいデータが分布するホットスポットであるとする．このとき，各ホットスポットで生成されたモバイルセンサデータを取得することで，検索範囲内のホットスポット検出が可能となる．このためには，分析の目的に応じたユーザの注目する環境属性の値に基いてデータにスコアを割り当て，スコアの高いデータを優先的に選択する必要がある．加えて，データ間の空間距離を考慮することで，検索結果が特定の空間位置に集中することなく，検索範囲内で分散した高スコアのデータを取得できる．

このようなデータ検索は，Top-k 検索結果の多様化 [2] によって実現できる．モバイルセンサデータベースにおける Top-k 検索結果の多様化では，データのスコアとデータ間の空間距離によって，データの評価値が定義される．最終的な検索結果は，検索範囲内のすべてのデータの評価値を計算し，評価値が最大

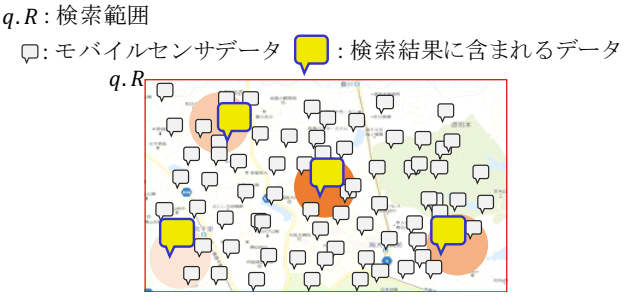


図 1　ホットスポットと Top-k 検索結果の多様化

のデータを選択するという動作を，k 回繰り返すことで算出される．筆者らの研究グループではこれまでの研究において，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 [9]．提案手法では，あらかじめ空間位置が近く，かつ環境属性値が似ているデータ同士をクラスタリングしておく．オンラインクエリ処理時にこれらのクラスタ情報を利用し，最適なデータを含まないクラスタを走査対象から除外することで，評価値が最大のデータを短時間で取得できる．

しかし，従来手法では，モバイルセンサデータの環境情報の数が増えると，計算時間が増大する問題があった．これは，環境属性値ベクトルが高次元になるほどデータ間の非類似度が大きくなるため，球形クラスタの半径が大きく，かつクラスタの数も大幅に増加することで，走査するデータの数が多くなってしまうためである．近年，様々な環境情報が参加型センシングによって観測，収集されているため，環境属性値ベクトルの次元が大きい場合でも検索結果を短時間で取得できることが望ましい．

そこで本稿では，階層的クラスタリングを行うことで，環境属性値ベクトルの次元数の増加にともなう計算効率低下の影響

を軽減可能な拡張手法を提案する．拡張手法の階層的クラスタリングでは，空間的に近接するすべてのデータは 1 つの上位クラスタのメンバとし，上位クラスタの中でメンバデータの環境属性値の類似度に基いてさらに下位クラスタに分割する．拡張手法のオンラインクエリ処理では，階層クラスタ構造を利用し，上位クラスタ内のデータが取りうる評価値の上界を推定することで，下位クラスタをまとめて走査対象から除外できる．空間的に近接するデータからなる上位クラスタの数は，環境属性数の増加に非依存であり，従来手法におけるクラスタの数よりも少なくなる．そのため拡張手法では従来手法と比べて，走査すべきデータを短時間で絞り込める．

以下では，2 章で関連研究について述べ，3 章で想定環境を紹介し本稿の問題を定義する．4 章で従来手法を紹介し，5 章で拡張手法について説明する．最後に，6 章で本稿をまとめる．

2. 関 連 研 究

Top-k クエリ処理は，全体のデータセットの中からスコアの最も高い（低い）k 個のデータを取得する検索であり，様々な研究分野において重要な役割を担っている．固定センサネットワークにおける Top-k 検索 [5] は，最も高い（低い）環境属性値を観測した k 個のノードを検索結果として返す．一般的に，固定センサネットワークでは，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内で分散している．一方，モバイルセンサネットワークでは，モバイルセンサは非同期かつ時空間上の任意の位置でデータを生成する．このような環境では，従来の Top-k 検索を用いた場合，多くの冗長なモバイルセンサデータが検索結果に含まれてしまう．よって，検索範囲内に存在するホットスポットにおいて生成されたモバイルセンサデータを取得するためには，別の手法が要求される．

従来の Top-k 検索では，検索結果はデータの関連度にのみ基いて定まるが，ユーザの検索要求をより満たせるように，検索結果の多様性を考慮した研究が盛んに行われている [1], [2]．研究対象は，文書に対するキーワード検索，EC サイトにおける商品推薦など多岐に渡る．モバイルセンサデータベースにおいては，データのクエリに対する関連度を環境属性値から算出されるスコア，多様性をデータ間の空間距離とすることで，ホットスポット検出に応用できる．

近年，各国で様々な参加型センシングのプロジェクトが進められており，大気汚染指数，騒音指数，夜間の光量といった，多様な環境情報が収集されている [4], [7], [8]．また，文献 [6] では，センサデータの複数の環境属性値から重回帰分析を用いて，別の環境属性値を推定する手法が提案されている．このような背景から，今後ますます多くの環境情報が，モバイルセンサデータベースにおいて包括的に扱われることが予想される．一般的にデータが高次元になると，検索アルゴリズムが非効率的となるため，適応的な手法が求められる．

3. Top-k 検索結果の多様化

3.1 想定環境と問題定義

モバイルセンサ端末は，周期的に付近の大気汚染指数，気温，湿度などの物理現象についてセンシングするものとする．ユーザの検索クエリを q，検索範囲を $q.R$ としたとき，検索範囲内に分布するデータ集合を O で表す．$q.R$ は Top-k 検索結果の多様化処理を行う時空間範囲であり，図 1 における矩形領域が該当する．以降では，$q.R$ の範囲外に存在するデータは無視し，データ集合 O のみを検索対象とする．

各データのスコアは，クエリ q に基づいて決定される．ユーザはクエリ q に対して，各環境属性に対する興味の度合いを示す重み付け係数 $q.\boldsymbol{w}$ を付与する．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．

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

式 (1) 中の w_i は i 番目の環境属性に対する重みを示す．以降では文脈上明らかな場合は，$p(q,o)$ を $p(o)$ のように略記する．2 つのデータ間の多様性は，空間距離 d で表される．$d(u,v)$ はデータ u，v の位置情報から算出される u，v 間のユークリッド距離であり，以下の式に従って計算される．

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

上述した環境属性値から算出されるデータのスコア，およびデータ間の位置情報から算出される空間距離に基づいて，モバイルセンサデータベースにおける Top-k 検索結果の多様化を以下のように定義する．

定義. クエリ $q = \{R, k, \lambda, \boldsymbol{w}\}$ が与えられたとき，データ集合 O を検索範囲内で観測されたデータ集合 $O = \{o_i \mid o_i \in q.R\}$ とする．このとき，以下の式で与えられる最適化問題を解くことによって，最適な検索結果 S_k^* が得られる．

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

ここで，$f(S_k, q, p(\cdot), d(\cdot,\cdot))$ は目的関数である．以降では文脈上明らかな場合は，$f(S_k, q, p(\cdot), d(\cdot,\cdot))$ を $f(S_k)$ のように略記する．本研究では，高いスコアを取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 [3] における Maxmin 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 S 内のデータのスコアが大きいほど目的関数の値は大きくなり，また，正解集合 S 内の任意のデータ間の距離が大きいほど目的関数の値は大きくなる．目的関数中の 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．

Algorithm 1 Algorithm for the Optimization Problems

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

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セットサイズ N が大きいとき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．様々なヒューリスティックな手法の中でも，グリーディアルゴリズムは得られる正解集合の質および計算時間の観点から効果的であることが知られており，様々な目的関数に応じた手法が提案されている．そこで，本研究においてもグリーディアルゴリズムをベースラインとする．

3.2 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，Algorithm 1 に示す．1，2 行目の初期化処理では，データセット内で最大のスコアをとるデータを正解集合に追加する．3 行目から 6 行目の反復により，正解集合の大きさが k となるまで，繰り返しデータを正解集合に追加する．4 行目の $d'(\cdot, S)$ は，データのスコアとデータ間の空間距離から算出される評価値であり，次の式で定義される．

$$d'(y, S) = \min_{u \in S} \{\frac{1}{2}(p(y) + p(u)) + \lambda d(y, u)\} \qquad (5)$$

以下の式に示すように，4 行目で追加されるデータ y^* は，評価値 $d'(\cdot, S)$ を最大化すると同時に，目的関数 $f(S \cup y)$ を最大化する．

$$y^* = \arg\max_{y \in O \backslash S} d'(y, S) = \arg\max_{y \in O \backslash S} f(S \cup y). \qquad (6)$$

このアルゴリズムの計算量は，データセットサイズ 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，単純にデータセット全体の走査が必要となり，計算量は $O(N)$ である．また，3 行目から 6 行目の反復については，反復回数が k，各反復につき最大 $k(N-k)$ 回の評価値の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，データセットサイズが大きくなると計算時間が長くなってしまう．

4. 従 来 手 法

従来手法は，オフライン事前クラスタリング処理とオンラインクエリ処理からなる．本章では，それぞれについて詳しく説明する．

4.1 オフライン事前クラスタリング処理

具体的なクラスタリングアルゴリズムを，Algorithm 2 に示す．

Algorithm 2 Algorithm for Clustering

Input: Data set O, r_1, r_2

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

1: clusterLabel $= 1$
2: **for** $i = 1$ to $|O|$ **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, O)
6: **for all** $o \in X$ **do**
7: **if** o is not in any clusters **then**
8: Mark o with current clusterLabel
9: **end if**
10: **end for**
11: clusterLabel++
12: **end if**
13: **end for**

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中心データかつ代表データとする．ここで，5 行目の retrieveNeighbors(o_i, r_1, r_2, O) は，データ集合 O から，データ o_i に空間位置が互いに近く，かつ環境属性値が互いに似ているデータを返す操作である．具体的には，データ o_i の空間位置ベクトル $o_i.\boldsymbol{loc}$ を中心とした半径 r_1 の円内に存在し，かつ，データ o_i の環境属性値ベクトル $o_i.\boldsymbol{att}$ 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．n 次元の環境属性値について，半径 r_2 の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．

$$O_i = \{o \mid \sqrt{\sum_{j=1}^{n} (o_i.att_j - o.att_j)^2} \leqq r_2\}. \qquad (7)$$

retrieveNeighbors(o_i, r_1, r_2, O) で取得したデータのうち，いずれのクラスタにも属していないデータに対し，現在作成中のクラスタのラベルを付与する．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．

4.2 クラスタを利用したオンラインクエリ処理

4.2.1 アルゴリズム

具体的なアルゴリズムを Algorithm 3 に示す．ここでは特に，走査データ数削減の要点である，Algorithm 3 の反復部分（3〜16 行目）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 o^*_{rep} を探索する（4 行目）．このデータの評価値が，各クラスタ内のデータを走査すべきかどうかを判断する基準値となるため，以降はこのデータを基準データと呼ぶ．またこのときに，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ように，クラスタ内データが取りうる評価値の上界を推定する際に必要となるためである．次に，全クラスタについて，各クラス

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

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（6 行目）．各クラスタの上界の推定値と，最初に計算した基準データの評価値 $d'(o^*_{rep}, S)$ を比較し，推定値のほうが大きい場合，走査対象クラスタ集合 $\boldsymbol{C'}$ 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内のデータよりも，基準データ 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，以降の走査から除外する．

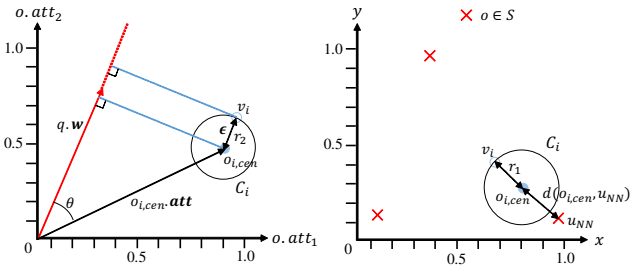
最後に，走査対象クラスタ集合 $\boldsymbol{C'}$ に含まれるすべてのデータを走査し，最大の評価値をとるデータを，正解集合 S に追加する．追加されたデータがいずれかのクラスタの代表データであった場合，クラスタ内のデータからランダムに新たな代表データを選択する（12〜14 行目）．

4.2.2 クラスタの上界の推定

ここで，Algorithm 3 の 6 行目における，各クラスタ内データが取りうる評価値の上界の推定方法について説明する．クラスタが含むデータの分布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，可能な限り正確に計算する．評価値は，データ間の空間距離と，環境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．

$$d'(v_i, S) = \frac{1}{2} p(v_i) + \min_{u \in S} \{\frac{1}{2} p(u) + \lambda d(v_i, u)\} \tag{8}$$

まず，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．クラスタ内に存在しうる仮想データ v_i のスコアは，重みベクトル $q.\boldsymbol{w}$ と環境属性値ベクトル $v_i.\boldsymbol{att} = o_{i,cen}.\boldsymbol{att} + \boldsymbol{\epsilon}$ の内積であり，最大値は以下の式で与えられる（図 2(a)）．



(a) 第1項の上界推定　(b) 第2項の上界推定

図 2　クラスタの上界の推定

$$\begin{aligned}
&\max_{v_i \in C_i} \{p(v_i)\} \\
&= \max_{|\boldsymbol{\epsilon}| \leqq r_2, 0 \leqq \theta \leqq 2\pi} \{q.\boldsymbol{w} \cdot (o_{i,cen}.\boldsymbol{att} + \boldsymbol{\epsilon})\} \\
&= (p(o_{i,cen}) + \max_{|\boldsymbol{\epsilon}| \leqq r_2, 0 \leqq \theta \leqq 2\pi} (|q.\boldsymbol{w}||\boldsymbol{\epsilon}| cos\theta)) \\
&= p(o_{i,cen}) + |q.\boldsymbol{w}| r_2.
\end{aligned} \tag{9}$$

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．クラスタの中心データと正解集合内のデータとの，各データのスコアを加味した距離を計算し，その時の距離が最小となる正解集合内のデータを $u_{NN} \in S$ とする．中心データとデータ u_{NN} を直線で結んだ時，2 つの交点が存在する（図 2(b)）．ここで，データ u_{NN} から最も離れる位置は，2 つの交点の内，データ u_{NN} から遠い方の点である．仮想データ v_i がこの点に位置するとき，正解集合 S からの距離も最大化される．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．

$$\begin{aligned}
&\max_{v_i \in C_i} \{\min_{u \in S} (\frac{1}{2} p(u) + \lambda d(v_i, u))\} \\
&= \frac{1}{2} p(u_{NN}) + \lambda \{d(o_{i,cen}, u_{NN}) + r_1\}
\end{aligned} \tag{10}$$

これらの式から，クラスタの上界を計算でき，以下の式で示される．

$$\begin{aligned}
\overline{d'(C_i, S)} &= \frac{1}{2} \{p(o_{i,cen}) + |q.\boldsymbol{w}| r_2\} \\
&\quad + \frac{1}{2} p(u_{NN}) + \lambda \{d(o_{i,cen}, u_{NN}) + r_1\} \\
&= \{\frac{1}{2} (p(o_{i,cen}) + p(u_{NN})) + \lambda d(o_{i,cen}, u_{NN})\} \\
&\quad + \frac{1}{2} |q.\boldsymbol{w}| r_2 + \lambda r_1 \\
&= d'(o_{i,cen}, S) + \frac{1}{2} |q.\boldsymbol{w}| r_2 + \lambda r_1
\end{aligned} \tag{11}$$

ここで，式 (11) の中心データの評価値 $d'(o_{i,cen}, S)$ は，アルゴリズムの上界推定の前に計算し記憶されている．

4.3 問　題　点

従来手法では，クラスタの上界が基準データの評価値未満であれば，そのクラスタ内のデータは走査する必要がなく，短時間で検索結果が得られる．できるだけ多くのクラスタを走査対象から除外するために，基準データの評価値はできるだけ大きい方が望ましく，従来手法では，全てのクラスタの代表データ

Algorithm 4 Algorithm for Hierarchical Clustering

Input: Data set O, r_1, r_2

Output: Set of upper clusters $\boldsymbol{UC} = UC_1, UC_2, ..., UC_l$

```
1: upperClusterLabel = 1
2: for i = 1 to |O| do
3:   if o_i is not in any clusters then
4:     Mark o_i as the center of the current upper cluster
5:     X = retrieveNeighbors(o_i, r_1, ∞, O)
6:     for j = 1 to |X| do
7:       lowerClusterLabel = 1
8:       if o_ij is not in any clusters then
9:         Mark o_ij as the center and initial representative of the
           current lower-level cluster
10:        X' = retrieveNeighbors(o_ij, ∞, r_2, X)
11:        for all o ∈ X' do
12:          if o is not in any clusters then
13:            Mark o with current lowerClusterLabel and up-
               perClusterLabel
14:          end if
15:        end for
16:      end if
17:      lowerClusterLabel++
18:    end for
19:    upperClusterLabel++
20:  end if
21: end for
```

の内，評価値が最大のものを探索する．また，基準データを探索した後，全てのクラスタの上界を計算し基準データの評価値と比較する．このため，走査するためのデータ数を削減するコストは，クラスタの数に依存する．しかし，モバイルセンサデータの環境属性の数が増えるほど，データ間の類似度が小さくなるため，クラスタの数は増加する．クラスタ半径 r_2 を大きくすればクラスタの増加は抑制できるが，基準データの評価値よりも大きくクラスタの上界を推定してしまう可能性が高く，効果的ではない．

5. 拡張手法

本章では，クラスタ増加の影響を軽減する，階層的クラスタリングを利用した拡張手法について説明する．拡張手法では，空間的に近接するデータを上位クラスタとしてまとめるため，上位クラスタの数は従来手法のクラスタの数と比べて小さくなる．よって，従来手法におけるクラスタ単位の走査対象データの絞り込みや基準データの選択を上位クラスタ単位で行うことで，環境属性値ベクトルが高次元である場合でも，効率的に評価値最大のデータを取得できる．

5.1 オフライン事前階層的クラスタリング処理

階層クラスタを構築するために，まず最初にいずれのクラスタにも属していないデータについて空間的に近接するデータ集合を取得し，これらをすべて上位クラスタのメンバとする．次に，上位クラスタ中のあるメンバに着目したとき，環境属性値が類似しているメンバを下位クラスタのメンバとする．上位クラスタのメンバがすべて，いずれかの下位クラスタに割り当てられるまで上述した処理を繰り返すことで，階層クラスタが構築される．

具体的な階層的クラスリングアルゴリズムを，Algorithm 4 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 o_i を見つけた場合，o_i の空間位置ベクトルを中心とした上位クラスタを生成する．5 行目で，全データ集合 O から，上位クラスタの中心データ o_i の空間位置ベクトルを中心とした半径 r_1 の円内に存在するデータを取得する．取得したデータのうち，ラベルが付与されていないデータは全てこの上位クラスタのメンバであり，これらのデータをさらに環境属性値に基づきクラスタリングする．8，9 行目で，上位クラスタのメンバのうち，いずれのクラスタにも属していないデータ o_{ij} を見つけた場合，そのデータを新たな下位クラスタの中心データかつ代表データとする．10 行目で，上位クラスタのメンバから，データ o_{ij} の環境属性値ベクトルを中心とした半径 r_2 の超球内に存在するデータを取得する．取得したデータのうち，ラベルが付与されていないデータに対し，現在作成中の上位クラスタおよび下位クラスタのラベルを付与する．ここで，上位クラスタ UC_i が m 個の下位クラスタを含むとき，上位クラスタおよび下位クラスタの関係を以下のように表現する．

$$UC_i = \{LC_{i1}, LC_{i2}, ..., LC_{im}\} \tag{12}$$

従来手法と同様，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．

5.2 階層クラスタを利用したオンラインクエリ処理

5.2.1 アルゴリズム

従来手法では，すべてのクラスタを探索し基準データを選択していたが，拡張手法では上位クラスタのみを探索し，上位クラスタの代表データの中から基準データを選択する．また，上位クラスタの評価値の上界を計算することで，走査対象データの絞り込みを上位クラスタ単位で行う．上位クラスタの上界が基準データの評価値を上回る場合は，下位クラスタの上界を計算し基準データの評価値と比較することで，より細かい粒度で走査対象データの絞り込みを行う．

具体的なアルゴリズムを Algorithm 5 に示し，反復部分（3〜25 行目）について説明する．まず，全上位クラスタを走査し，基準データを選択する（5 行目）．基準データの評価値は高いほど望ましいため，上位クラスタ内の下位クラスタの代表データをすべて走査し，スコアが最大のものを選択する．各上位クラスタごとに，スコアが最大の下位クラスタの代表データを記憶しておき，以降の反復ではこれら $|\boldsymbol{UC}|$ 個のデータの中から基準データを選択する．次に，全上位クラスタについて，各上位クラスタ内のデータの評価値が取りうる値の上界 $\overline{d'(UC_i, S)}$ を推定する（7 行目）．各上位クラスタの上界の推定値と，基準データの評価値 $d'(o^*_{rep}, S)$ を比較し，推定値のほうが大きい

Algorithm 5 Algorithm for Optimization Problems Leveraging Hierarchical Clusters

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

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

1: Initialize the set $S = \emptyset$
2: Find $x^* = \arg\max_{x \in O} p(x)$ and set $S = \{x^*\}$
3: **while** $|S| < k$ **do**
4: Initialize the sets $\boldsymbol{UC'} = \boldsymbol{LC'} = \emptyset$
5: Find o^*_{rep} such that $o^*_{rep} = \arg\max_{o_{i,rep} \in UC_i} d'(o_{i,rep}, S)$
6: **for all** $i = 1$ to $|\boldsymbol{UC}|$ **do**
7: Estimate upper bound of each upper cluster $\overline{d'(UC_i, S)}$
8: **if** $d'(o^*_{rep}, S) \leqq \overline{d'(UC_i, S)}$ **then**
9: $\boldsymbol{UC'} = \boldsymbol{UC'} \cup \{UC_i\}$
10: **end if**
11: **end for**
12: **for all** $i = 1$ to $|\boldsymbol{UC'}|$ **do**
13: **for all** $j = 1$ to $|UC_i|$ **do**
14: Estimate upper bound of each lower-level cluster $\overline{d'(LC_{ij}, S)}$
15: **if** $d'(o^*_{rep}, S) \leqq \overline{d'(LC_{ij}, S)}$ **then**
16: $\boldsymbol{LC'} = \boldsymbol{LC'} \cup \{LC_{ij}\}$
17: **end if**
18: **end for**
19: **end for**
20: Find $y^* \in \boldsymbol{LC'} \backslash S$ such that $y^* = \arg\max_{y \in \boldsymbol{LC'} \backslash S} d'(y, S)$
21: **if** y^* is representative data of LC_{ij} **then**
22: Select new representative data for LC_{ij}
23: **end if**
24: Set $S = S \cup \{y^*\}$
25: **end while**

場合，走査対象上位クラスタ集合 $\boldsymbol{UC'}$ に追加する．一方，推定値のほうが小さい場合，少なくともこの上位クラスタ内のすべてのデータよりも，基準データ 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．全上位クラスタをチェックした後，走査対象上位クラスタ集合 $\boldsymbol{UC'}$ に含まれる下位クラスタについて，走査対象をさらに絞り込む（12〜19 行目）．ここでは，上位クラスタの絞り込みと同様，各下位クラスタの上界の推定値と基準データの評価値を比較し，走査対象下位クラスタ集合 $\boldsymbol{LC'}$ に該当する下位クラスタを追加する．最後に，$\boldsymbol{LC'}$ 内のすべてのデータの評価値を計算し，最大の評価値を取るデータを正解集合 S に追加する．

5.2.2 上位クラスタおよび下位クラスタの上界の推定

ここで，Algorithm 5 の 7 行目における，上位クラスタ内データが取りうる評価値の上界 $\overline{d'(UC_i, S)}$ の推定方法について説明する．上位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．上位クラスタ内のデータは，環境属性値の類似度を考慮してさらに下位クラスタに分割される．そのため，上位クラスタの上界は，下位クラスタの上界のうち最大のものとして，以下の式で与えられる．

$$\overline{d'(UC_i, S)} = \max_{LC_{ij} \in UC_i, 1 \leqq j \leqq m} \{\overline{d'(LC_{ij}, S)}\} \tag{13}$$

次に，下位クラスタ内データが取りうる評価値の上界 $\overline{d'(LC_{ij}, S)}$ の推定方法について説明する．従来手法のクラスタの上界と同様，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し，別々に上界を計算することで求める．正解集合内のデータに非依存の項（第 1 項）の上界は，下位クラスタの中心データ $o_{ij,cen}$ を用いて，以下の式で計算できる（図 3(a)）．

$$\max_{v_{ij} \in LC_{ij}} \{p(v_{ij})\} = p(o_{ij,cen}) + |q.\boldsymbol{w}| r_2 \tag{14}$$

次に，正解集合内のデータに依存する項（第 2 項）の上界を計算する．下位クラスタ内データが取りうる空間位置は，上位クラスタの中心データ $o_{i,cen}$ の空間位置ベクトルを中心とした半径 r_1 の円内に限られる．よって，上位クラスタの中心データと正解集合内のデータとの，各データのスコアを加味した距離を計算し，その時の距離が最小となる正解集合内のデータを $u_{NN} \in S$ とすると，以下の式で計算できる（図 3(b)）．

$$\begin{aligned} &\max_{v_{ij} \in LC_{ij}} \{\min_{u \in S}(\frac{1}{2}p(u) + \lambda d(v_{ij}, u))\} \\ &= \frac{1}{2}p(u_{NN}) + \lambda\{d(o_{i,cen}, u_{NN}) + r_1\} \end{aligned} \tag{15}$$

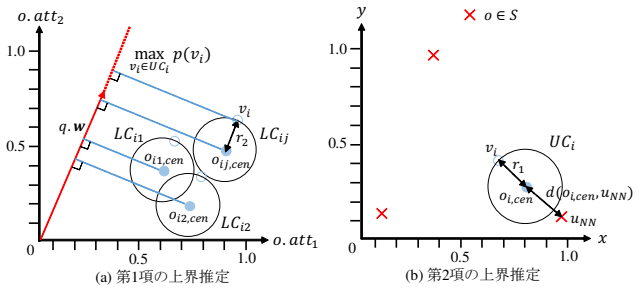
これらの式から，下位クラスタの上界を計算でき，以下の式で示される．

$$\begin{aligned} \overline{d'(LC_{ij}, S)} &= \frac{1}{2}\{p(o_{ij,cen}) + |q.\boldsymbol{w}| r_2\} \\ &\quad + \frac{1}{2}p(u_{NN}) + \lambda\{d(o_{i,cen}, u_{NN}) + r_1\} \\ &= \{\frac{1}{2}(p(o_{ij,cen}) + p(u_{NN})) + \lambda d(o_{i,cen}, u_{NN})\} \\ &\quad + \frac{1}{2}|q.\boldsymbol{w}| r_2 + \lambda r_1 \\ &= d'(o_{ij,axis}, S) + \frac{1}{2}|q.\boldsymbol{w}| r_2 + \lambda r_1 \end{aligned} \tag{16}$$

ここで $o_{ij,axis}$ は，上位クラスタ UC_i の中心データ $o_{i,cen}$ の空間位置ベクトルと，下位クラスタ LC_{ij} の中心データ $o_{ij,cen}$ の環境属性値ベクトルからなり，以下の式で与えられる．

$$o_{ij,axis} = (o_{i,cen}.loc_x, o_{i,cen}.loc_y, o_{ij,cen}.att_1, ..., o_{ij,cen}.att_n) \tag{17}$$

このデータの評価値 $d'(o_{ij,axis}, S)$ は，Algorithm 5 の 5 行目の基準データを探索する際に計算でき，値を下位クラスタごとに記憶しておくことで，以降の反復で継続して用いることができる．また，式 (16) から下位クラスタの上界は，下位クラスタ内のデータが取りうるスコア（第 1 項）の上界にのみ依存する．よって，上位クラスタの上界は，スコアの上界が最大の下位クラスタの上界となる（図 3(a)）．スコアの上界は，正解集合内のデータに非依存であるため一度だけ計算されればよく，以降は上位クラスタごとに記憶した値を用いる．

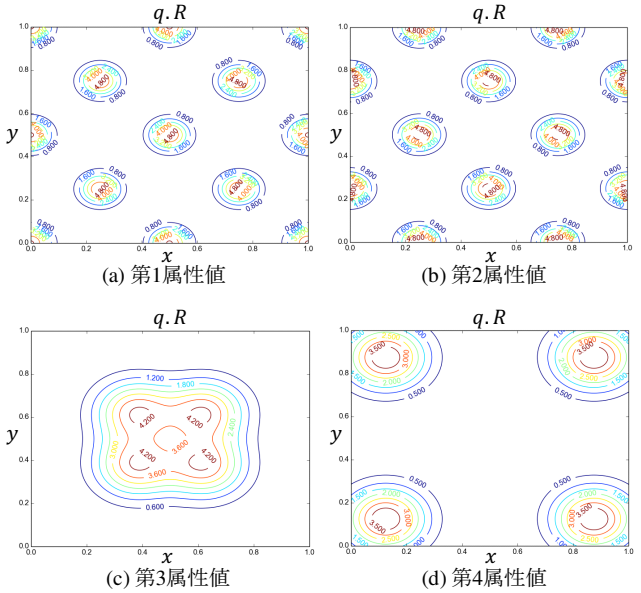


(a) 第1項の上界推定　(b) 第2項の上界推定

図 3　階層クラスタの上界の推定

表 1　パラメタの値

パラメタ	値
データセットサイズ N	1M
要求データ数 k	5 - 50 (**15**)
$\boldsymbol{w}$ の各要素	0.0∼1.0
λ	0.0∼5.0
環境属性の次元数 n	4, **10**, 20



(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 4　環境属性値分布

6. 評価実験

Top-k 検索結果の多様化処理における，拡張手法の性能を評価する．表 1 は各パラメタの値を示し，太字はデフォルト値とする．

データの位置情報を，各次元の値が区間 $[0, 1]$ 上の一様分布に従う，2 次元ベクトルで与えた．また，データの環境属性値は，図 4 に示すような空間的自己相関の特徴を有する分布に従う値とし，1 次元から 4 次元まで設定した．環境属性数が 10 または 20 の場合は，これらのパターンを繰り返し第 5 属性以降に割り当てた．具体的な環境属性値は，データの位置情報から決定される．また，センシング時の誤差を考慮して，位置情報から決定される環境属性値に対し，$N(0, 0.3)$ の正規分布に従う正規乱数を加算した．

比較手法は，3.2 節で説明したベースライン手法，および 4. 章で説明した従来手法を用いた．

実験においては，オンラインクエリ処理でセンサデータおよびクラスタデータを RAM 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．実験で用いたクエリは，$q.\boldsymbol{w}$ と $q.\lambda$ がそれぞれ表 1 に示す一定範囲内でランダムに設定されたものである．作成されたランダムな 100 個のクエリを処理した際の，計算時間の平均値を調べた．

6.1　空間半径 r_1 および環境属性値半径 r_2 の影響

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，環境属性が 4，10，20 次元の場合について，それぞれ図 5(a)，(b)，(c) に示す．また，クラスタの環境属性値半径 r_2 を変化させた場合の計算時間についても，同様に図 5(d)，(e)，(f) に示す．それぞれの図から，クラスタ半径が大きい場合，従来手法・拡張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，クラスタ内データの評価値の上界を過大に推定しており，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．一方，クラスタ半径が小さい場合も，従来・拡張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，クラスタ半径が小さくなると，生成されるクラスタ数が増加することによる．走査対象クラスタ数を削減できても，クラスタの上界を計算するためにすべてのクラスタの中心データを走査しなければならず，結果として全体の計算時間は長くなってしまう場合がある．ここで，拡張手法における最短の計算時間が，ベースライン手法，従来手法に比べて短くなっていることがわかる．

次節で述べる実験では，従来・拡張手法それぞれで，計算時間を最短にした r_1 および r_2 を用いている．

6.2　要求データ数 k の影響

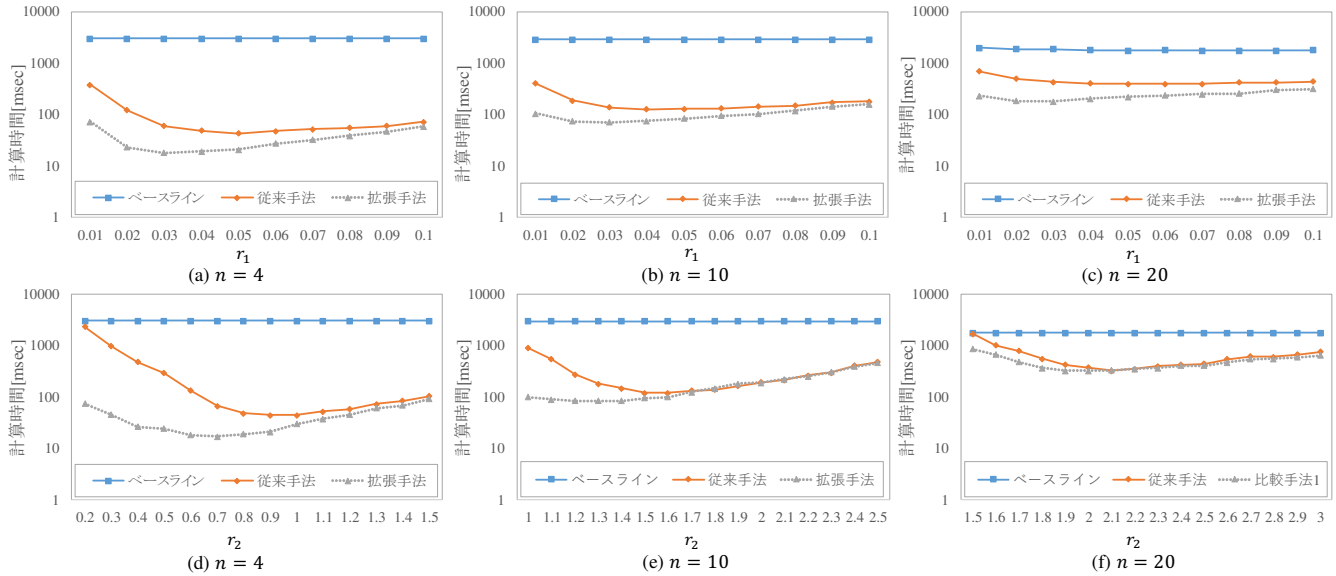
ユーザごとに，要求データ数は異なる．そこで，要求データ数 k を変化させた場合の計算時間を図 6 に示す．従来・拡張手法の効果は，検索範囲内に存在するホットスポットの数に依存する．すなわち，各ホットスポットにおけるデータが全て探索されるまで，走査データ数削減の効果は大きく働く．ここでも，拡張手法における最短の計算時間が，ベースライン手法，従来手法に比べて短くなっていることを確認した．

7. おわりに

本稿では，高次元な環境情報を扱うモバイルセンサデータベースにおいて，効率的な Top-k 検索結果の多様化手法を提案した．拡張手法では，事前にモバイルセンサデータに対し階層的クラスタリングを行う．オンラインクエリ処理では，上位クラスタおよび下位クラスタの上界を推定し，それぞれのクラスタ単位で走査対象データの絞り込みを行う．また，基準データの候補を上位クラスタごとに記憶することで，短時間で評価値の高い基準データを取得できる．

クラスタ半径，要求データ数 k を変化させたシミュレーション実験から，拡張手法は従来手法に比べ計算時間を削減できることを確認した．

近年，検索結果の多様化問題を継続的に行う研究が盛んに取り組まれている．今後の課題の一つとして，モバイルセンサデータベースにおける，効率的かつ高精度な多様化結果のモニ



(a) $n = 4$ (b) $n = 10$ (c) $n = 20$

(d) $n = 4$ (e) $n = 10$ (f) $n = 20$

図 5　空間半径 r_1 および環境属性値半径 r_2 の影響

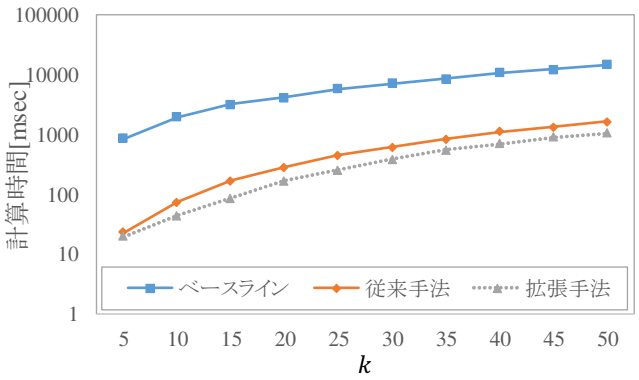


図 6　要求データ数 k の影響

タリング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．

文　　献

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

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

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

[4] Hasenfratz, D., Saukh, O., Walser, C., Hueglin, C., Fierz, M., Arn, T., Beutel, J. and Thiele, L.: Deriving high-resolution urban air pollution maps using mobile sensor nodes, *Pervasive and Mobile Computing*, Vol. 16, Part B, pp. 268 – 285 (2015).

[5] Jiang, H., Cheng, J., Wang, D., Wang, C. and Tan, G.: A General Framework for Efficient Continuous Multidimensional Top-k Query Processing in Sensor Networks, *Parallel and Distributed Systems, IEEE Transactions on*, Vol. 23, No. 9, pp. 1668–1680 (2012).

[6] Kurasawa, H., Sato, H., Yamamoto, A., Kawasaki, H., Nakamura, M., Fujii, Y. and Matsumura, H.: Missing sensor value estimation method for participatory sensing environment, *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 103–111 (2014).

[7] Ma, Q., He, C., Wu, J., Liu, Z., Zhang, Q. and Sun, Z.: Quantifying spatiotemporal patterns of urban impervious surfaces in China: An improved assessment using nighttime light data, *Landscape and Urban Planning*, Vol. 130, pp. 36 – 49 (2014).

[8] Rana, R. K., Chou, C. T., Kanhere, S. S., Bulusu, N. and Hu, W.: Ear-phone: An End-to-end Participatory Urban Noise Mapping System, *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, pp. 105–116 (2010).

[9] Yokoyama, M. and Hara, T.: Efficient Top-k Result Diversification for Mobile Sensor Data, *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pp. 477–486 (2016).