2C-6

大規模空間データからの最適領域集合の効率的な発見手法
Efficient Discovery of Optimal Regions from Large Spatial Data

森川裕章[1] Hiroaki Morikawa　浅井達哉[1] Tatsuya Asai　多湖真一郎[1] Shinichiro Tago　稲越宏弥[1] Hiroya Inakoshi
湯上伸弘[1] Nobuhiro Yugami　岡本青史[1] Seishi Okamoto

1 はじめに

近年，空間データを扱う情報処理が注目されている．そこで，我々は大規模空間データをリアルタイムに分析できるシステムの開発に取り組んでいる．空間データ分析において，最適領域を複数発見することが応用上重要である．ところが，発見した最適領域の交差を考慮しない場合，発見した最適領域はお互い似通ってしまう可能性があり，新しい知見が得られにくい問題がある．そこで，最適領域が互いに交差しないような領域発見問題を考える．

最適領域発見については，いくつかの手法が提案されている．Dobkin 等 [1] の手法は，点の値の総和が高い領域を見つけるものである．Fukuda 等 [2] の手法は，2 つの数値データを 2 次元領域上に配置し，グリッド (メッシュ) 領域に分解後，パラメータを満たす領域 (グリッドの集合) を見つけ出すものである．いずれの方法も見つける領域は一つである．

本稿では，交差なし複数最適領域発見の効率の良いアルゴリズムについて述べる．まず，交差なし複数最適領域発見問題を定式化し，その問題に対する効率的なアルゴリズムを提案する．さらに，アルゴリズムを実装し，実験による評価を行う．

2 準備

集合 A の大きさを $|A|$ で表す．$\mathbf{R}$ を実数集合とする．$P = \{p_1, \ldots, p_n\}$ を点集合とする．ただし，$p_i \in \mathbf{R}^2 (1 \leq i \leq n)$ である．$\mathcal{R} \in 2^{\mathbf{R}^2}$ を領域の集合とする．本稿で考察する領域集合 $\mathcal{R}$ について，下記を満たすと仮定する [1].

- 任意の $R \in \mathcal{R}$ は，x 軸と y 軸に平行な線分からなる長方形とする．
- 任意の $S \subseteq P$ に対して，$S \subseteq R$ を満たし，かつ，任意の $p \in P \backslash S$ に対して，$p \notin R$ を満たす領域 $R \in \mathcal{R}$ が唯一存在する．

関数 $f : P \to \mathbf{R}$ とし，関数 $cal : \mathcal{R} \to \mathbf{R}$ とした時，$score = cal(R) = \sum_{p \in R} f(p)$ は，R 内の点 $p \in (R \cap P)$ の値の総和を表す．次に，$maxscore := \max_{R \in \mathcal{R}} cal(R)$ とする．この時，$R_1 \in \mathcal{R}$ が最適領域とは，$maxscore = cal(R_1)$ を満たすことをいう．さらに，m-最適領域 $R \in \mathcal{R}$ を，以下のように再帰的に定義する．

- m=1 の時：最適領域 R は 1 −最適領域である．
- $m \geq 2$ の時：$\mathcal{A}_{m-1} = \{R_1, \ldots, R_{m-1}\}$ とする．さらに $\mathcal{R}_m := \mathcal{R} \backslash \mathcal{A}_{m-1}$ に対して，$maxscore_m = \max_{R \in \mathcal{R}_m} cal(R)$ とする．この時，R が m-最適領域とは，$maxscore_m = cal(R)$ を満たすことをいう．

[1]株式会社富士通研究所 ソフトウェア&ソリューション研究所 ナレッジテクノロジ研究部

任意の 2 つの領域 $U, V \in \mathcal{R}$ に対して，$U \cap V = \emptyset$ が成り立つ時，U と V は**非交差**であるという．交差最適領域集合を $\mathcal{I}_m \in \mathcal{A}_m$ を以下のように再帰的に定義する．

- m=1 の時：$\mathcal{I}_1 = \emptyset$ である．
- $m \geq 2$ の時：

$$\mathcal{I}_m = \begin{cases} \mathcal{I}_{m-1} \cup \{R_m\} & (\exists R \in \mathcal{A}_{m-1}, R \cap R_m \neq \emptyset) \\ \mathcal{I}_{m-1} & (otherwise) \end{cases}$$

次に，非交差最適領域集合 $\mathcal{N}_m \in \mathcal{A}_m$ を定義する．

- m=1 の時：$\mathcal{N}_1 = \{R_1\}$．ただし，R_1 は 1 −最適領域である．
- $m \geq 2$ の時：$\mathcal{N}_m = \mathcal{A}_m \backslash \mathcal{I}_m$

交差最適領域集合の定義により，$|\mathcal{N}_m|$ は $|\mathcal{N}_{m-1}|$+1 または $|\mathcal{N}_{m-1}|$ と等しい．(ただし，$m \geq 2$ の時：$|\mathcal{N}_1| = 1$)
以上より，次の問題を定義する．

定義 1 交差なし複数最適領域発見問題とは，入力として点集合 $P = \{p_1, \ldots, p_n\}$ と出力数 k 受け取り，$|\mathcal{N}_m|$=k となる $\mathcal{N}_m$ を見つける問題である．

3 アルゴリズム

本節では，交差なし複数最適領域発見問題を効率的に解くアルゴリズム **SplitRegionSearch** の基本的アイディアを述べる．任意の点集合 $S \subseteq P$ 内の点を全て含む領域を R_S とし，その 1-最適領域を R_{S_1} とする．この時，SplitRegionSearch は，$R_S \backslash R_{S_1}$ の領域を**分割長方形**と呼ぶ高々4 つの長方形領域に分割する．そして，それぞれの長方形に対し，再帰的に 1-最適領域を求めることにより，最適領域同士の交差判定をなくし，高速化を実現する．

定義 2 $[a, b]$ を $\{x \in \mathbf{R} | a \leq x \leq b\}$ となる閉区間とする．閉区間の直積を，$\mathbf{R}^2$ の閉区間･短形といい，これは周を含んだ長方形である．$R_S = [x_{min}, x_{max}] \times [y_{min}, y_{max}]$ とし，$R_{S_1} = [x_{lower}, x_{upper}] \times [y_{lower}, y_{upper}]$ とする．この時，**分割長方形** (図 1) は次の 4 つの長方形である．$R_{top} = [x_{min}, x_{max}] \times [y_{upper}, y_{max}]$，$R_{left} = [x_{min}, x_{lower}] \times [y_{min}, y_{max}]$，$R_{bottom} = [x_{min}, x_{max}] \times [y_{min}, y_{lower}]$，$R_{right} = [x_{upper}, x_{max}] \times [y_{min}, y_{max}]$

分割長方形には，図 1 内の編みかけ部分のように，探索領域が重複する領域が存在する．この領域については，交差判定を行う必要があるが，交差判定すべき領域の個数は，高々$3(k-1)+1$ である．

点の数を n，発見する最適領域の数を k とすると，SplitRegionSearch は，1-最適領域発見処理に Dobkin 等 [1] の手法を用いることにより，交差なし複数最適領域発見問題を $O(k^2 n^2 \log n)$ 時間で解く．

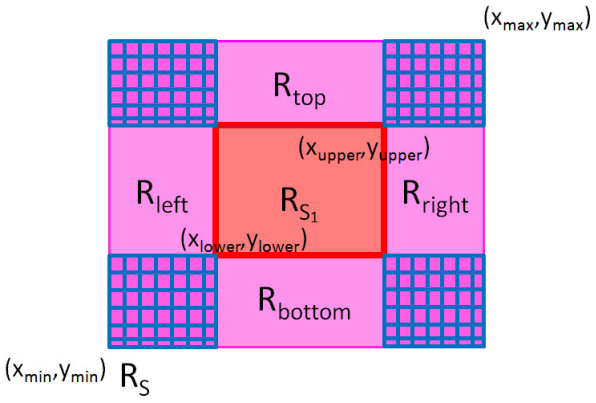

図 1: 分割長方形

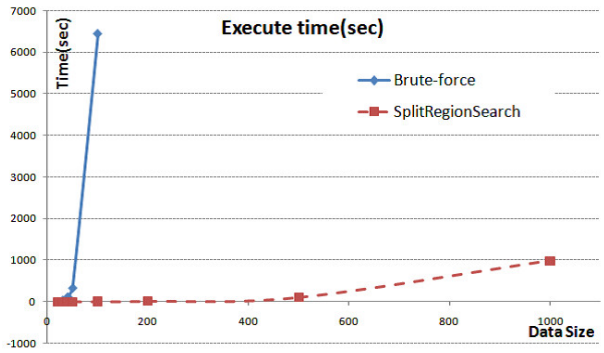

図 2: 実行時間の比較

図 3: 川崎市の人口増加数の高い地域

4 計算機実験

提案アルゴリズムを Java 言語 (Java 6) で実装し，PC (Corei3 2.93GHz，2.93GB RAM，Windows7) 上で実験を行なった．本実験では，1-最適領域発見処理に実装が容易な $O(n^3)$ の方法を用いた [1]．この場合，提案アルゴリズムの計算時間は $O(k^2n^3)$ 時間である．実験には，2 種類のデータ (共に CSV 形式) を用いた．1 つ目のデータは，テストプログラムにて $[0.0, 0.0] \times [1.0, 1.0]$ 空間内に 1000 個の点をランダムに配置し，点に対して-100 から 100 までの任意の値を与えたものを用いた (データ 1)．2 つ目のデータは，総務省統計局[1] が公開している平成 12 年と平成 17 年の川崎市の GIS データを用い，値は，平成 12 年と平成 17 年の人口の差分とした (データ 2)．データ 2 を処理し求める領域は人口増加数の多い上位 k 件の領域となる．2 つの実験共に $k = 3$ とし，上位 3 件の最適領域を求めた．

比較アルゴリズムとして，力任せ探索 (Brute-force search) を Java 言語で作成した．力任せ探索では入力データ内の任意の点について，その点を通る x 軸と y 軸に平行な線分を考える．x 軸に平行な線分の集合を $Edge_X$，y 軸に平行な線分の集合を $Edge_Y$ とする．$Edge_X$ と $Edge_Y$ からそれぞれ 2 本選択し，4 本の線分で囲まれる領域内の $score$ を求める．$score$ が一番高い領域を最適領域とする．$\mathcal{N}_m (m \geq 2)$ を求めるには，$\mathcal{N}_{m-1} \cap \{R_m\} = \emptyset$ となるような $R_m \in \mathcal{R}$ で $score$ が一番大きいものを，$\mathcal{N}_m$ に格納する．これを $|\mathcal{N}_m| = k$ になるまで実行する．力任せ探索の計算時間は $O(kn^5)$ 時間である．

規模耐性：大規模のデータに対して，提案手法の実行時間を検証した．データ 1 のデータ数を 10 から 1000 まで変化させながら，2 つの手法の実行時間を計測した．実行時間の比較のグラフを図 2 に示す．力任せ探索はデータ数 200 で約 20 時間かかった．データ数 100 で比較した場合，力任せ探索が約 6442 秒 (1 時間 47 分) である一方，提案手法は 1.92 秒であり，提案手法の方が約 3000 倍高速であることが分かった．

実データを用いた領域発見：次に実データ (データ 2) に対して，提案手法が意味のある最適領域を見つけることができるか検証した．データ 2 のデータサイズは 609 である．提案手法は，約 53 秒で上位 3 件の最適領域を見つけた．見つけた最適領域を Google マップ[2] 上に表示したものを図 3 に示す．実データに対し，提案手法を適用することにより，交差しない複数最適領域を効率的に探し出すことができた．

5 まとめ

本稿では，交差なし複数最適領域発見問題を定式化し，効率のよいアルゴリズムを提案した．プロトタイプを実装し，実験を通じて，提案手法が既存手法より高速に動作することを確認した．

今後は，1-最適領域を見つける他のスコア計算関数 (平均，密度) を考察し，今回提案した手法と組み合わせながら，複数最適領域発見問題の応用を検討していく．

参考文献

[1] D. P. Dobkin, D. Gunopulos, W. Maass. Computing the Maximum Bichromatic Discrepancy with Applications to Computer Graphics and Machine Learning. J. Comput. Syst. Sci. 52(3): 453–470, 1996.

[2] T. Fukuda, Y. Morimoto, S. Morishita, T. Tokuyama. Data Mining Using Two-Dimensional Optimized Accociation Rules:Scheme, Algorithms, and Visualization. SIGMOD Conference 1996: 13-23，1996.

[1]総務省統計局 地図で見る統計 (統計 GIS) http://www.e-stat.go.jp

[2]Google マップ http://maps.google.co.jp/