

情報基礎数理学特選
有限距離空間の離散幾何学とアルゴリズム
(6) 近似最近点探索

岡本 吉央

北陸先端科学技術大学院大学

2011年1月12日

”最終更新：2012/02/06 10:40”

① 近似最近点探索

② バケット法

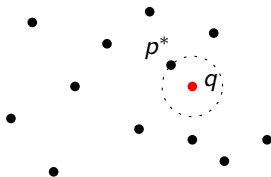
③ 近似最近点探索：補足

最近点

(X, μ) : 距離空間, $P \subseteq X$

定義：最近点

P における $q \in X$ の最近点 (nearest neighbor) とは,
 $\min\{\mu(p, q) \mid p \in P\}$ の最適解のこと



最近点探索

(X, μ) : 距離空間, $P \subseteq X$, $|P| = n$

問題 : 最近点探索 (nearest neighbor search)

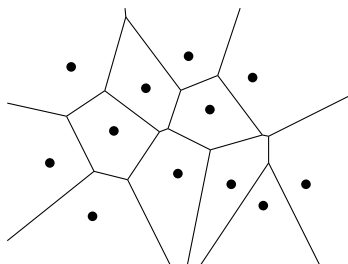
P に前処理を施し, 任意の $q \in X$ (質問点 (query point)) に対して, P における q の最近点を高速に答えられるようにせよ

最近点探索アルゴリズムのよさの指標

- ▶ 前処理時間 : 前処理にかかる時間計算量
- ▶ データ構造サイズ : 前処理によって作られるデータ構造の大きさ
- ▶ 問合せ時間 : 各質問に対して答えを返す時間計算量

ユークリッド空間の場合

一般的なアプローチ：Voronoi 図 (Voronoi diagram) の利用



- ▶ $X = \mathbb{R}^2$, l_2 距離
 - ▶ 前処理時間： $O(n \log n)$ (Shamos, Hoey '75, Fortune '85, ...)
 - ▶ データ構造サイズ： $O(n)$
 - ▶ 問合せ時間： $O(\log n)$

ユークリッド空間 (高次元) の場合

▶ $X = \mathbb{R}^d$, l_2 距離

	データ構造サイズ	問合せ時間
Clarkson '88	$O(n^{\lceil d/2 \rceil (1+\delta)})$	$O(2^{O(d \log d)} \log n)$
Meiser '93	$O(n^{d+\delta})$	$O(d^5 \log n)$

($\delta > 0$ は任意の定数)

次元 d に対する依存性が強い (次元の呪い (curse of dimensionality))

↪ 近似を考える

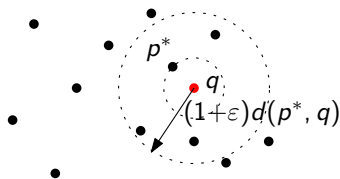
近似最近点

定義： ε -最近点

P における $q \in X$ の ε -最近点 (ε -nearest neighbor) とは,

$$d(p^*, q) \leq d(p, q) \leq (1+\varepsilon)d(p^*, q)$$

を満たす $p \in P$ のこと. (ただし, p^* は P における q の最近点)



問題：近似最近点探索 (approximate nearest neighbor search)

P に前処理を施し, 任意の質問点 $q \in X$ に対して,
 P における q の ε -最近点を高速に答えられるようにせよ

l_2^d で近似最近点探索問題を解く手法 — ナイーブな手法

単純に JL 補題を適用

アルゴリズム A

前処理

- ① $A \in \mathbb{R}^{k \times d}$: JL 補題の行列 ($k = O(\frac{1}{\epsilon^2} \log n)$)
- ② P から $\tilde{P} = \{Ap \mid p \in P\} \subseteq \mathbb{R}^k$ を作成
- ③ \tilde{P} に対して最近点探索の前処理をほどこす

問合せ：質問点 q が与えられたとき

- ① \tilde{P} における Aq の最近点を探索
- ② $Ap \in \tilde{P}$ が返されたら, p を ϵ -最近点として出力

前処理 (3), 問合せ (1) で Meiser の手法を用いると (ϵ : 定数として)

- ▶ データ構造サイズ: $n^{O(\log n) + \delta}$
- ▶ 問合せ時間: $O(\log^6 n)$

それほどよい計算量ではない

l_2^d で近似最近点探索問題を解く手法 — 改善のポイント

「ナイーブな手法」のやっていること

JL 補題を適用した後に，最近点探索問題を「厳密」に解いている

- ▶ 厳密の近似は近似

改善のポイント

JL 補題を適用した後に，最近点探索問題を「厳密」に解く必要はない

改善のために行うこと

JL 補題を適用した後に，近似最近点探索問題を解く

- ▶ 近似の近似は近似

まず，JL 補題を使わずに近似最近点探索問題を解くことを考える

l_2^d で JL 補題を使わずに近似最近点探索問題を解く手法

(Indyk, Motwani '98)

アルゴリズムの概要

① 問題を近似最近点探索問題に帰着する

② 近似最近点探索問題を解く

手法 1: バケット法

手法 2: LSH 法 (Locality-Sensitive Hashing)

- ▶ この講義では「バケット法」のみを説明 (LSH 法については割愛)
- ▶ バケット法も LSH 法もハッシュを利用

近点

$r \geq 0$: 実数

定義 : r -近点

P における $q \in X$ の r -近点 (r -near neighbor) とは
点 $p \in P$ で,

$$\mu(p, q) \leq r$$

を満たすもの

近似近点探索問題

設定

- ▶ (X, μ) : d 次元ユークリッド空間
- ▶ $P \subseteq X$, $|P| = n$
- ▶ $r \geq 0$, $\varepsilon > 0$: 実数

問題 : ε -近似 r -近点探索 (ε -approximate r -near neighbor search)

P に前処理を施し , 任意の質問点 $q \in X$ に対して ,
次が高速にできるようにせよ

- ▶ P における q の r -近点が存在するとき , 「Yes」 を出力し ,
 q の $(1 + \varepsilon)r$ -近点を答える
- ▶ P における q の $(1 + \varepsilon)r$ -近点が存在しないとき , 「No」 を出力する
- ▶ どちらでもないとき , 「Yes」 か 「No」 を出力する

近似最近点探索問題を解く方法 — 近似最近点探索に帰着

アルゴリズム B

前処理

- ① $r = 1, 1 + \varepsilon/3, (1 + \varepsilon/3)^2, (1 + \varepsilon/3)^3, \dots$ に対して
- ② $\varepsilon/3$ -近似 r -近点探索問題に対するデータ構造 D_r を作成

質問点 $q \in X$ が与えられたとき

- ① $r = 1$
- ② D_r に対して, q の近似 r -近点を問合せ
- ③ D_r から p を出力 $\Rightarrow p$ を q の ε -最近点として出力
- ④ 何も出力しない $\Rightarrow r \leftarrow (1 + \varepsilon/3)r$ として (2) に戻る

注: $\min\{\mu(p, q) \mid p, q \in P\} = 1$ と (一般性を失わず) 仮定

アルゴリズム B の正当性

証明すべきこと

ε が十分小さいとき, 出力 p は q の ε -最近点である

証明: p^* を q の最近点とする

- ▶ $r = (1 + \varepsilon/3)^i$ のときに p を ε -最近点として出力したと仮定
(つまり, $\mu(p, q) \leq (1 + \varepsilon/3)r = (1 + \varepsilon/3)^{i+1}$)
- ▶ $\therefore r = (1 + \varepsilon/3)^{i-1}$ のときにアルゴリズム A は出力をしなかった
- ▶ $\therefore (1 + \varepsilon/3)^{i-1}$ -近点は存在しない
- ▶ $\therefore (1 + \varepsilon/3)^{i-1} < \mu(p^*, q)$
- ▶ $\therefore \mu(p, q) \leq (1 + \varepsilon/3)^{i+1} < (1 + \varepsilon/3)^2 \mu(p^*, q) \leq (1 + \varepsilon) \mu(p^*, q) \quad \square$

注: ε が十分小さくないときは, $\varepsilon/3$ の「3」を大きな定数に変えればよい

アルゴリズム B の計算量 (1)

解決すべき問題

どこまで r を大きくすれば十分か？

- ▶ $D = \max\{\mu(p, p') \mid p, p' \in P\}$ とする
- ▶ P において q の D/ε -近点が存在しないと仮定

- ▶ $p^* : q$ の最近点 $(\mu(p^*, q) > D/\varepsilon)$
- ▶ 任意の $p \in P$ に対して

$$\mu(p, q) \leq \mu(p, p^*) + \mu(p^*, q) \leq D + \mu(p^*, q) < (1 + \varepsilon)\mu(p^*, q)$$

- ▶ \therefore 任意の p が q の ε -最近点

$\therefore r$ を D/ε より大きくなったら, それ以上大きくする必要はない

近似最近点探索問題を解く方法 (改訂版)

アルゴリズム B

前処理

① $r = 1, 1 + \varepsilon/3, (1 + \varepsilon/3)^2, (1 + \varepsilon/3)^3, \dots, D/\varepsilon$ に対して

② $\varepsilon/3$ -近似 r -近点探索問題に対するデータ構造 D_r を作成

質問点 $q \in X$ が与えられたとき

① $r = 1$

② D_r に対して, q の近似 r -近点を問合せ

③ D_r から p を出力 $\Rightarrow p$ を q の ε -最近点として出力

④ 何も出力しない $\Rightarrow r \leftarrow (1 + \varepsilon/3)r$ とする

① $r \leq D/\varepsilon \Rightarrow$ (2) に戻る

② $r > D/\varepsilon \Rightarrow$ 任意の $p \in P$ を q の ε -最近点として出力

アルゴリズム B の計算量 (2)

前処理

- ▶ 作成するデータ構造の数 = $O(\log(D/\epsilon))$
- ▶ \therefore 前処理時間 = $\epsilon/3$ -近似 r -近点探索の前処理時間 $\times O(\log(D/\epsilon))$
- ▶ \therefore データ構造サイズ = $\epsilon/3$ -近似 r -近点探索のデータ構造サイズ $\times O(\log(D/\epsilon))$

問合せ

- ▶ 問合せ時間 = $\epsilon/3$ -近似 r -近点探索の問合せ時間 $\times O(\log(D/\epsilon))$

注：二分探索 (binary search) を用いれば

「 $O(\log(D/\epsilon))$ 」を「 $O(\log \log(D/\epsilon))$ 」にできる

l_2^d で JL 補題を使わずに近似最近点探索問題を解く手法

(Indyk, Motwani '98)

アルゴリズムの概要

① 問題を近似最近点探索問題に帰着する

② 近似最近点探索問題を解く

手法 1: バケット法

手法 2: LSH 法 (Locality-Sensitive Hashing)

- ▶ この講義では「バケット法」のみを説明 (LSH 法については割愛)
- ▶ バケット法も LSH 法もハッシュを利用

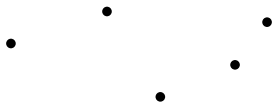
① 近似最近点探索

② バケット法

③ 近似最近点探索：補足

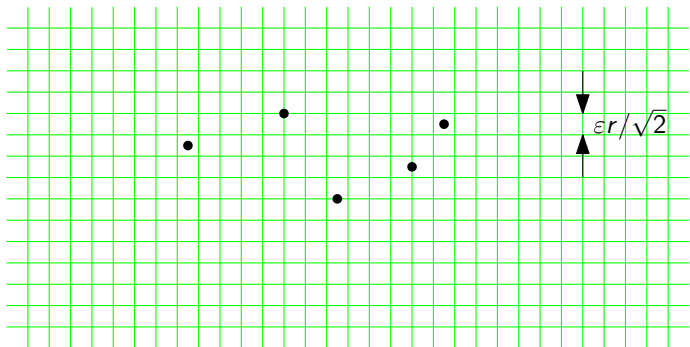
バケット法：前処理

- ▶ 1辺の長さが $\varepsilon r / \sqrt{d}$ の格子を考える
- ▶ ある $p \in P$ からの距離が r 以下の直方体を (ハッシュ表に) 蓄える
- ▶ どの p からの距離が r 以下なのかも同時に蓄える



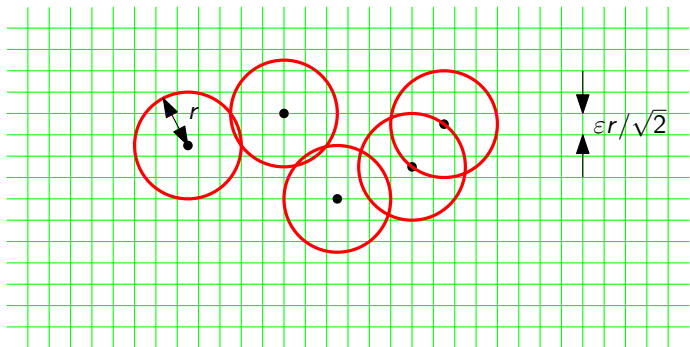
バケット法：前処理

- ▶ 1辺の長さが $\epsilon r / \sqrt{d}$ の格子を考える
- ▶ ある $p \in P$ からの距離が r 以下の直方体を (ハッシュ表に) 蓄える
- ▶ どの p からの距離が r 以下なのかも同時に蓄える



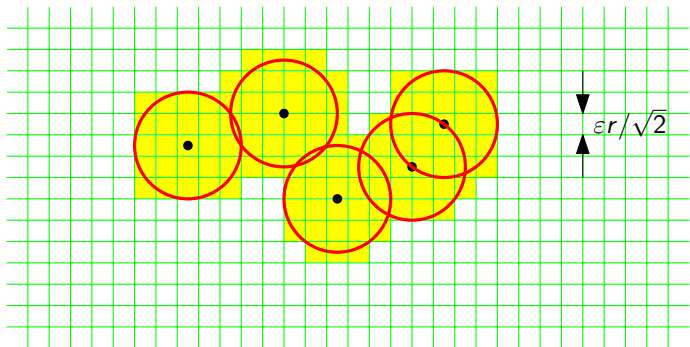
バケット法：前処理

- ▶ 1辺の長さが $\epsilon r / \sqrt{d}$ の格子を考える
- ▶ ある $p \in P$ からの距離が r 以下の直方体を (ハッシュ表に) 蓄える
- ▶ どの p からの距離が r 以下なのかも同時に蓄える



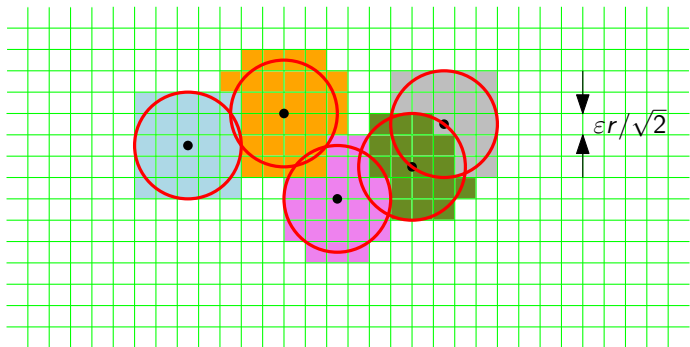
バケット法：前処理

- ▶ 1辺の長さが $\epsilon r / \sqrt{d}$ の格子を考える
- ▶ ある $p \in P$ からの距離が r 以下の直方体を (ハッシュ表に) 蓄える
- ▶ どの p からの距離が r 以下なのかも同時に蓄える



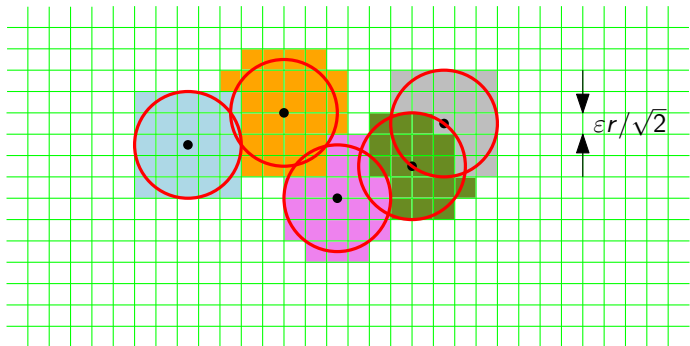
バケット法：前処理

- ▶ 1辺の長さが $\epsilon r / \sqrt{d}$ の格子を考える
- ▶ ある $p \in P$ からの距離が r 以下の直方体を (ハッシュ表に) 蓄える
- ▶ どの p からの距離が r 以下なのかも同時に蓄える



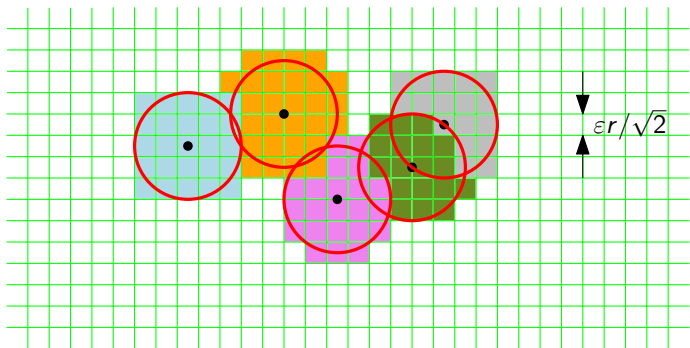
バケット法：問合せ

- ▶ 質問点 q を含む直方体が蓄えてあるか確認
- ▶ 蓄えてあれば，その直方体に関連付けられた点 $p \in P$ を出力
- ▶ 蓄えてなければ「No」を出力



バケット法の正当性

- ▶ 質問点 q が $p \in P$ に関連付けられた直方体 B に含まれると仮定
- ▶ B の対角線の長さ $= \sqrt{d \times (\epsilon r / \sqrt{d})^2} = \epsilon r$
- ▶ p と q の間の距離 $\leq r + \epsilon r = (1 + \epsilon)r$



バケット法の計算量

- ▶ 問合せ時間 = ハッシュ関数の計算時間 ($= O(d)$)
- ▶ 前処理時間, データ構造サイズ = 蓄える直方体の数に比例

$$\begin{aligned}
 \text{蓄える直方体の数} &\leq n \times \frac{\text{vol}(B_2(0, (1 + \varepsilon)r))}{(\varepsilon r / \sqrt{d})^d} \\
 &\leq n \times \frac{O(1/\sqrt{d})^d ((1 + \varepsilon)r)^d}{(\varepsilon r / \sqrt{d})^d} \\
 &= n \times O(1/\varepsilon)^d
 \end{aligned}$$

- ▶ 前処理時間, データ構造サイズ = $O(O(1/\varepsilon)^d n)$

バケット法 + JL 補題

バケット法と Johnson–Lindenstrauss の補題を組み合わせると

- ▶ 各点に対する射影の計算量 = $O(\frac{d}{\epsilon^2} \log n)$
(高速 JL 変換により, 改善可能)
- ▶ 射影後の次元 = $O(\frac{1}{\epsilon^2} \log n)$

よって

- ▶ 前処理時間 = $O(n \frac{d}{\epsilon^2} \log n) + O(O(1/\epsilon)^{O(\frac{1}{\epsilon^2} \log n)} n)$
= $O(n \frac{d}{\epsilon^2} \log n + n^{O(1/\epsilon^3)})$
- ▶ データ構造サイズ = $O(O(1/\epsilon)^{\frac{1}{\epsilon^2} \log n} n) = O(n^{O(1/\epsilon^3)})$
- ▶ 問合せ時間 = $O(\frac{d}{\epsilon^2} \log n) + O(\frac{1}{\epsilon^2} \log n) = O(\frac{d}{\epsilon^2} \log n)$

まとめ

定理 6.1

近似最近点探索問題は次の計算量で解くことができる

- ▶ 前処理時間： $O((n \frac{d}{\epsilon^2} \log n + n^{O(1/\epsilon^3)}) \log(D/\epsilon))$
- ▶ データ構造サイズ： $O(n^{O(1/\epsilon^3)} \log(D/\epsilon))$
- ▶ 問合せ時間： $O(\frac{d}{\epsilon^2} \log n \log(D/\epsilon))$

ただし，

- ▶ n は与えられる点の数
- ▶ d は次元
- ▶ D は与えられる点間の最大距離
- ▶ ϵ は近似精度 (十分小さい)

d に対する依存性は圧倒的に小さい

① 近似最近点探索

② バケット法

③ 近似最近点探索：補足

補足

- ▶ JL 補題の与える埋め込みが「 P に依存しない」ということは、問合せに答えるためには重要
 - ▶ Indyk, Motwani '98 は計算量に対してもっと強い上界を示している
(詳細は原論文参照)
 - ▶ バケット法は任意の $p \in [1, 2]$ に対する ℓ_p^d で使える
 - ▶ LSH 法 (Locality Sensitive Hashing) は他の空間でも使える
 - ▶ 活発な研究分野で, wikipedia 情報も充実
- 近似最近点探索に関するオンライン情報源
- ▶ Similarity Search Wiki
<http://sswiki.tierra-aoi.net/index.php>
 - ▶ ここで紹介したものが最新研究状況というわけではないので注意

ランダム射影の応用全般 (近似最近点探索に1章費やしている)

- ▶ S. Vempala. The Random Projection Method. AMS, 2004.

幾何近似アルゴリズム全般について (近似最近点探索にも詳しい)

- ▶ S. Har-Peled. Geometric Approximation Algorithms. Lecture Notes, 2004–.

<http://valis.cs.uiuc.edu/~sariel/teach/notes/aprx/>

<http://valis.cs.uiuc.edu/~sariel/teach/notes/aprx/book.pdf>

(2011年にAMSから書籍として出版された)

① 近似最近点探索

② バケット法

③ 近似最近点探索：補足