Dense Nearest Neighborhood 問題の検索手法

鈴木 日奈[†1] 陳 漢雄[†2] 古瀬 一隆[†3]

†1 筑波大学情報学群情報メディア創成学類 〒305-8577 茨城県つくば市天王台１丁目１－１
†2 筑波大学システム情報系 〒305-8577 茨城県つくば市天王台１丁目１－１
†3 白鷗大学経営学部 〒323-8585 栃木県小山市駅東通り２丁目２－２
E-mail: †1 s1611445@u.tsukuba.ac.jp, †2 chx@cs.tsukuba.ac.jp, †3 furuse@fc.hakuoh.ac.jp

あらまし 空間データベースにおいて，クエリ点から最も近いオブジェクトを検索するという最近傍検索問題の中に NNH (Nearest Neighborhood) 問題がある．これは，指定個数のオブジェクトのグループ (i.e. クラスタ) のコンパクト性，クエリ点との距離を合わせて比較，判断し，最適なグループを検索する問題である．しかし，これまでの研究では，グループがもつオブジェクトの個数を指定しなければならないため，例えば指定した数からいくつかオブジェクト数を増減したグループで，ユーザーにとってより望ましいグループが存在していたとしても，それを求めることができないという課題がある．本研究では，この課題を解決すべく，グループがもつデータの個数を指定しない問題を考え，対応した新しい検索アルゴリズムを提案する．

キーワード 空間データベース，検索アルゴリズム，情報検索

1. はじめに

空間データベースにおいて最近傍検索問題は，データマイニングおよび情報検索など多くの分野において基本的かつ重要な問題であり，パターン認識を用いたサービスや施設情報，位置情報を用いた地図・ナビゲーションサービスといった様々なアプリケーションにおいて広く利用されている．近年ではさらに多様な検索問題に対する需要，そして，より効率的かつより正確にデータを検索する必要性が高まり，様々な研究が行われてきた．

その一種である NNH 問題[1]は、指定個数のオブジェクトのグループのコンパクト性・クエリ点との距離を合わせて比較・判断し，最適なグループを検索する問題である．応用例としては，観光客が，今いる場所から近く，ある程度密集して食べ歩きができるような複数の飲食店の一覧を検索したい場合や，SNS で特定のユーザーに近いコミュニティの検索，特定の施設に近く事故が多数起きている場所を調査するといったデータマイニングが挙げられる．図 1.1 は NNH 問題の一例を表す．黒い点がデータセットの各データ点 (i.e. オブジェクト), q がクエリ点, k がグループ内データ点の指定個数, ρ がグループを表す円の指定半径である．そして，これらのパラメータを基に,半径 ρ の円内に k 個のデータ点を持つ候補グループC_1，C_2，C_3が現れる．ここで，d_1，d_2，d_3はグループ円C_1，C_2，C_3の中心とクエリ点との距離を表す. NNH 問題では, ρ でコンパクト性を，d でクエリ点との距離を表し，この例では，クエリ点との距離が最も近いC_2が解となる．

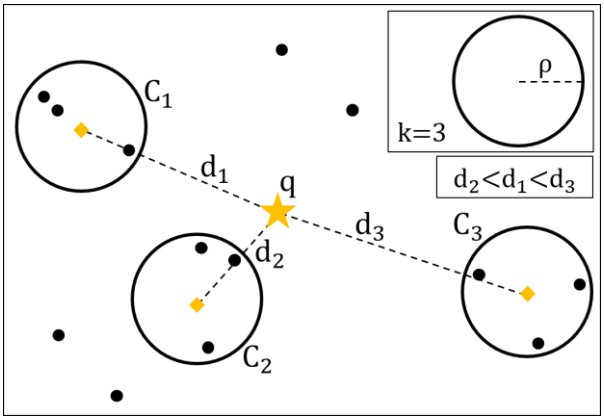

図 1.1: NNH 問題

しかし，NNH 問題ではグループのサイズ (i.e. グループを表す円の半径) やグループ内データ個数を指定しなければならないことから，解が空となり，ユーザーはパラメータを変更して何度も検索を行わなければならなくなる可能性がある．この課題を解決する拡張として既に提案されている BNNH (Balanced Nearest Neighborhood) 問題[2]では，グループのサイズを指定しないが，グループ内データ個数は指定することから，応用によっては，解がユーザーの望むものにならない可能性がある．そこで，本研究では，グループ内データ個数も指定しない問題を考え，対応する検索アルゴリズムを提案する．これにより，既存手法では検出できなかった，より望ましいグループを得ることができるようになる．

2. 関連研究

最近傍検索問題は，最も基本的な NN (Nearest

Neighbor) 問題にはじまり，数多くの効率化，拡張の研究が行われている．NN 問題とは，クエリ点に最も近い (指定個数の) データ点を検索する問題である．データ量が巨大化したことや多様な応用に対する需要が高まったことで，グループからデータ点を検索する ANN (Aggregate Nearest Neighbor) 問題[3], [4]，グループからグループを検索する GNG (Group Nearest Group) 問題[5]といった拡張研究が活発に行われている．

その中で本研究に最も関連するのは NNH 問題で，Choi, Chung により提案された[1]．また，この問題のさらなる拡張として Le らによって提案された BNNH 問題[2]がある．BNNH 問題では，NNH 問題で指定していたコンパクト性を測る指標，グループのサイズを指定しないという特徴があり，Le らは最適なグループを測る指標として以下の式で表される総合近似度を定義している．

$$\alpha|qc_i| + (1-\alpha)\rho_i$$

ここで，$|qc_i|$はクエリ点 q とグループ円 C_iの中心 c_iとの距離であり，ρ_iはC_iの半径を表す．そして，クエリ点との距離とコンパクト性のどちらを重視するかについて，ユーザー定義のパラメータ α ($0<\alpha<1$, 0 に近いほどコンパクト性を，1 に近いほどクエリ点との距離を重視) によって定義する．つまり，BNNH 問題では図 2.1 のように，サイズが異なるグループを比較し，αが 0 に近い場合は最もコンパクトなC_6を，1 に近い場合は最もクエリ点との距離が短いC_5を，0.5 つまりコンパクト性とクエリ点との距離双方を重視する場合はC_4を解とする．

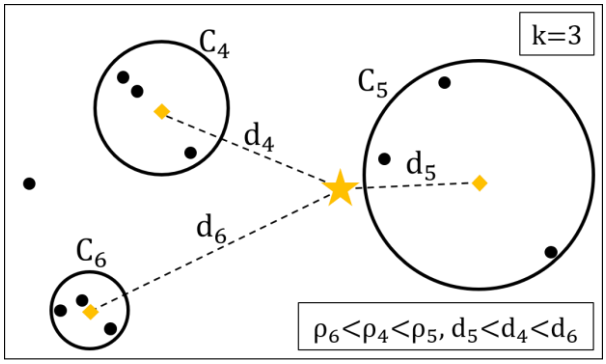

図 2.1: BNNH 問題

3. 提案手法

3.1. 既存手法の課題

既存手法の課題は，応用によっては，ユーザーの望む結果にならない可能性があることである．いくつかの例を挙げて説明する．まず，図 3.1 のC_7, C_8を比較する場合，クエリ点との距離が等しく，C_8のほうが小さいサイズであるため，C_8が解となる．しかし，C_7は 1 データ点が離れているためにサイズが大きくなっているだけで，残りのデータ点は明らかにC_8より密集しており，この密集したデータ点こそ，ユーザーにとって望ましいグループといえる．

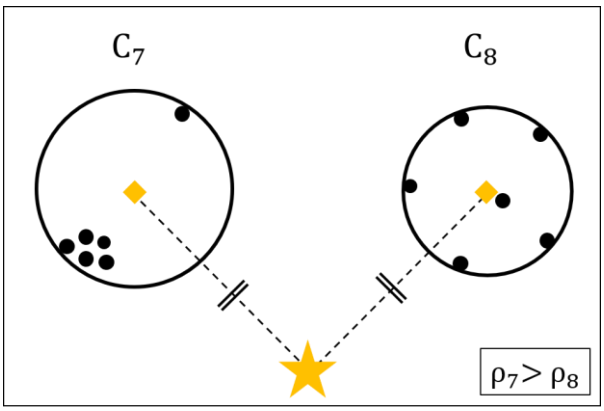

図 3.1: 既存手法の課題(1)

次に，図 3.2 のC_9, C_{10}を比較する場合，サイズが等しいため，クエリ点との距離が小さいC_{10}が解となる．しかし，それぞれのデータ点の分布をみると，C_{10}のデータ点のほとんどはC_9のデータ点のほとんどよりも，明らかにクエリ点から遠い位置にあり，C_9の密集したデータ点こそ，ユーザーにとって望ましいグループといえる．

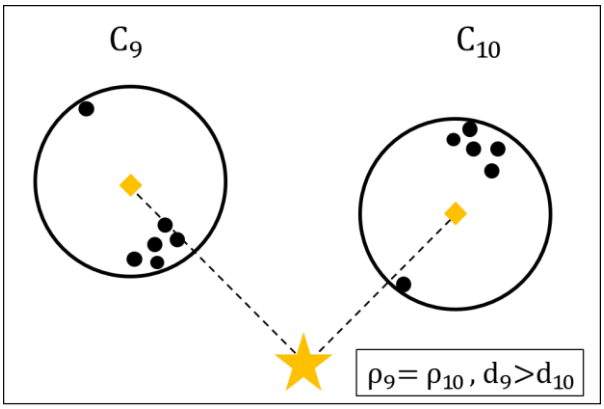

図 3.2: 既存手法の課題(2)

3.2. 問題定義

前述した課題の原因は，一部のデータ点が他のデータ点よりも遠く外れた位置にあるグループといった，明らかにクラスタと判断できないグループを比較してしまっていることにある．これは，グループ内データ個数を指定しなければならないことから発生する問題であり，指定しなければ解決できると考える．例えば，

図 3.1 のC_7は，図 3.3 のC'_7のように，比較するグループとしてより望ましいものにできる．

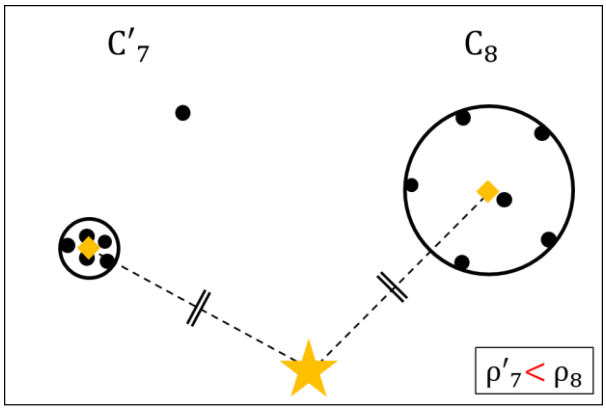

図 3.3: 改良例(1)

これは図 3.2 のC_9，C_{10}も同様に，図 3.4 のC'_9，C'_{10}のようになり，より望ましい比較を行うことができるようになる．

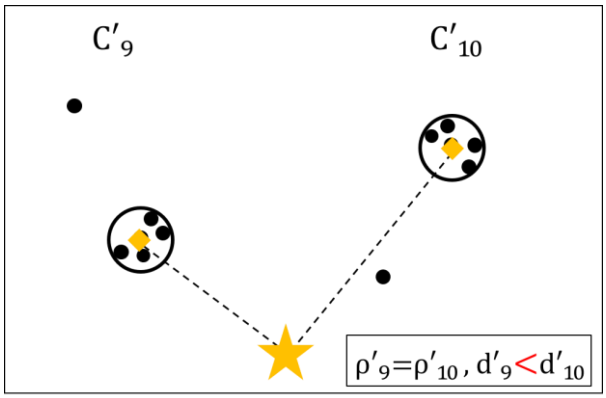

図 3.4: 改良例(2)

そこで，本研究では，このようにグループ内データ個数を指定しない，以下のように表される問題を定義する．

定義. (Dense Nearest Neighborhood 問題, DNNH). 入力として，データセットP，クエリ点qを与えたとき，DNNH 問題は出力として，最も小さい$\Delta(C,q)$の値をもつグループCを出力する．

ここで，Δは総合近似度を表し，最適なグループを測る指標として，本研究では以下の式により定義する．

$$\Delta(C,q) = |qc| + sd(C)$$

$$\left(sd(C) = \sqrt{\frac{1}{|C|}\sum_{i=1}^{|C|}(x_i - \bar{x})^2} \right)$$

Cはグループ，cはグループCの重心，sd(C)はグループCの標準偏差である．ここで，|C|はグループC内のデータ個数，x_iはC内の各データ点，$\bar{x}$はCの重心を表す．この式において，|qc|によってクエリ点とグループとの距離を，sd(C)によってグループのコンパクト性を測る．

これを図 3.3 のC'_7，C_8の例に適用すれば，図より$|qc'_7| = |qc_8|$であり，明らかに$sd(C'_7) < sd(C_8)$であるため$\Delta(C'_7, q) < \Delta(C_8, q)$，つまり望ましいグループ$C'_7$が解となる．図 3.4 の例においても，$|qc'_9| < |qc'_{10}|$，$sd(C'_9) \fallingdotseq sd(C'_{10})$であるため，$\Delta(C'_9, q) < \Delta(C'_{10}, q)$となり，こちらも，より望ましいグループ$C'_9$が解となる．

3.3. 提案アルゴリズム

DNNH 問題を解くための近似アルゴリズムとして，直感的に，まず，データセットに対しクラスタリングを行い，その後，結果の各クラスタに対しΔを計算し最適なクラスタを求める，という方法を考える．クラスタリング手法として Arthur, Vassilvitskii の k-means++[6]を用い，クラスタ数の推定には Tibshirani, Walther, Hastie の Gap 統計量[7] による方法と，Pelleg, Moore により提案され，石岡により改良された X-means[8][9]による方法を用いることとした．

3.3.1. Gap 統計量を用いたアルゴリズム

Gap 統計量とは，クラスタ数kのクラスタリング結果の評価として，クラスタのコンパクト性を表す指標W_kを用い，与えられたデータセットPと，一様乱数分布の(つまり，クラスタを持たない) B 個のデータセット$P_b(b = 1, \ldots, B)$のW_kを比較することで最適なクラスタ数を推定する手法である．Bはブートストラップ回数といい，後の計算において，このB個のデータセットの平均をとることで，与えられたデータセットとの対照としてより精度の高い比較を実現する．それゆえ，この値が高いほど精度が向上する．応用にもよるが，一般的には 100 前後で用いられ，500 程度で極めて高い正確性をもつ[10]．W_kは以下のように定義される．

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} \left(\sum_{\substack{i,i' \in C_r \\ i \neq i'}} \|p_i - p_{i'}\|^2 \right)$$

ここで，C_rは各クラスタ，n_rはクラスタC_rのデータ個

数，p_i，$p_{i'}$はデータセットPの各データ点を示す．また，与えられたデータセットと比較する一様乱数分布のデータセットP_bは，以下のように作成する．

$$P_b = \begin{pmatrix} p_{b,11} & \cdots & p_{b,1M} \\ \vdots & \ddots & \vdots \\ p_{b,N1} & \cdots & p_{b,NM} \end{pmatrix}$$

$$\min_i p_{ij} \le p_{b,ij} \le \max_i p_{ij}$$

Gap 値は以下のように表される．

$$\mathrm{Gap(k)} = \frac{1}{B}\sum_{b=1}^{B} \log W_{b,k} - \log W_k$$

最後に，最適なクラスタ数として，以下の式を満たす最小のkが解となる．

$$\mathrm{Gap(k)} \ge \mathrm{Gap(k+1)} - s(k+1)$$

$$\left(s(k) = sd_k \sqrt{1 + \frac{1}{B}} \right)$$

ここで，sd_kは$\{\log W_{b,k}\}_{b=1}^{B}$の標準偏差を表す．

提案するアルゴリズムは以下の通りである．

Algorithm: DNNH with Gap statistics

Input: P, q
Output: C

1: $C \leftarrow \emptyset$
2: $\mathrm{Gap}(0) \leftarrow -\infty$
3: **for** k in 1 to $|P|$ **do**
4: $\quad C_{set}^{(k)} \leftarrow k\text{-}means{+}{+}$
5: $\quad$ **if** $\mathrm{Gap}(k\text{-}1) \ge \mathrm{Gap}(k) - s(k)$ **then**
6: $\quad\quad C_{set} \leftarrow C_{set}^{(k-1)}$
7: $\quad\quad$ **break**
8: **for** each $C_i \in C_{set}$ **do**
9: $\quad$ if $\Delta(C_i, q) < \Delta(C, q)$ *then*
10: $\quad\quad C \leftarrow C_i$
11: **return** C

3-7 行目がクラスタリング，8-10 行目がΔの計算と評価である．k=1 から k-means++によるクラスタリングを行い，Gap 統計量の判定基準により最適なクラスタ数であると確認された場合は，その最適なクラスタセットを保持し，クラスタリングを終了する．その後，各クラスタに対しΔを計算していき，その値が最小となるクラスタを最適グループとして解とする．

3.3.2. X-means を用いたアルゴリズム

X-means は k-means を拡張したものであり，再帰的な 2-means による分割と情報量規準 BIC による分割停止基準を用いることによって，あらかじめクラスタ数を指定する必要なく，クラスタ数の推定とクラスタリングを同時に行えることが特徴の手法である．本研究においては，石岡により改良されたアルゴリズム[9]を用いた．これは，分散がクラスタ毎に異なる可能性を考慮した点，一部の計算に近似計算を用いることで，計算速度の向上を図っている点等が Pellog, Moore によるもとのアルゴリズム[8]と異なっている．このアルゴリズムの大まかな手順は以下の通りである．

0. データセットとして，n 個の p 次元データを用いる．
1. クラスタ数の初期値 k_o を指定する．(特に指定がなければ，$k_o = 2$)
2. $k\text{-}means(k = k_o)$ によるデータセットの分割を行い，分割後のクラスタを $C_1, C_2, \dots, C_{k_o}$ とする．
3. $i = 1,2,\dots,k_o$ として，手順 4~9 を繰り返す．
4. C_i に対して $k\text{-}means(k = 2)$によるデータセットの 2 分割を行い，分割後のクラスタを C_i^1, C_i^2 とする．
5. C_i に含まれるデータ x が p 変量正規分布 $f(x;\theta_i) = exp\{-\frac{1}{2}(x-\mu_i)^T V_i^{-1}(x-\mu_i)\}/\sqrt{(2\pi)^p |V_i|}$に従って分布していると仮定し，そのときの BIC を以下の通り定義する．

$$\mathrm{BIC} = -2\log L(\widehat{\theta_i}; x \in C_i) + q\log|C_i|$$

ここで，$\widehat{\theta_i} = [\widehat{\mu_i}, \widehat{V_i}]$ ($\mu_i \dots C_i$ の p 次平均値ベクトル，$V_i \dots C_i$ の $p \times p$ 分散共分散行列) は p 変量正規分布の最尤推定値である．q はパラメータ空間の次元数であり，$q = p + p + (p * p - p)/2 = p(p+3)/2$ (共分散を考慮しなければ $q = p + p = 2p$) となる．L は尤度関数であり，$L(\widehat{\theta_i}; x \in C_i) = \prod f(x \in C_i; \widehat{\theta_i})$ となる．

6. C_i^1, C_i^2 それぞれに含まれるデータ x が p 変量正規分布 $f(x;\theta_i^1), f(x;\theta_i^2)$ に従って分布していると仮定し，2 分割後のモデルにおいてデータの従う確率密度を以下の通りおく．

$$x \sim \alpha_i \left[f(x;\theta_i^1)\right]^{\delta_i} \left[f(x;\theta_i^2)\right]^{1-\delta_i}$$

$$\left(\delta_i = \begin{cases} 1\ (x \in C_i^1) \\ 0\ (x \in C_i^2) \end{cases} \right)$$

ここで，α_iはこの式を確率密度とするための基準化定数で，$\alpha_i = 1/\int [f(x;\theta_i^1)]^{\delta_i} [f(x;\theta_i^2)]^{1-\delta_i}\, dx$ であるが，計算量削減のため，以下の通り近似する．

$$\alpha_i = 0.5/K(\beta_i)$$

$$\left(\beta_i = \sqrt{\frac{\|\mu_1 - \mu_2\|^2}{|V_1| + |V_2|}} \right)$$

Kは標準正規分布の下側確率である．そして，2 分割後のモデルにおけるBICをBIC′とし，以下の通り定義する．

$$\mathrm{BIC}' = -2\log L\left(\widehat{\theta'}_i; x \in C_i\right) + q'\log|C_i|$$

ここで，$\widehat{\theta'}_i = [\theta_i^1, \theta_i^2]$は 2 つの$p$変量正規分布の最尤推定値である．$q'$はパラメータ空間の次元数であり，$q' = 2q = p(p+3)$ (共分散を考慮しなければ$q' = 4p$)となる．

7. BIC > BIC′であれば，2 分割後のモデルをより好ましいと判断し，$C_i \leftarrow C_i^1$とし，C_i^2をスタックに積んで手順 4 へ．
8. BIC ≤ BIC′であれば，分割前のモデルをより好ましいと判断し，分割を停止．スタックからデータを取り出し$C_i \leftarrow C_i^2$とし，手順 4 へ．スタックが空であれば次の手順へ．
9. C_iにおける分割がすべて終了．クラスタ番号等を整理する．

以上を踏まえて，本研究で提案する DNNH 問題を解く近似アルゴリズムは以下の通りになる．

Algorithm: DNNH with X-means

Input: P, q
Output: C

1: $C, C_{set} \leftarrow \emptyset$
2: $C_1, C_2, \ldots, C_{k_0} \leftarrow$ k-$means$++ $(P, k = k_0)$
3: **for** each $C_i \in \{C_1, C_2, \ldots, C_{k_0}\}$ **do**
4: splitClusterRecursively(C_i)
5: **for** each $C_i \in C_{set}$ **do**
6: if $\Delta(C_i, q) < \Delta(C, q)$ *then*
7: $C \leftarrow C_i$
8: **return** C

function splitClusterRecursively(C)
f-1: $C_1, C_2 \leftarrow$ k-$means$++ $(C, k = 2)$
f-2: **if** BIC(C) > BIC′(C_1, C_2) **then**
f-3: **for** each $C_i \in \{C_1, C_2\}$ **do**
f-4: splitClusterRecursively(C_i)
f-5: **else**
f-6: *Insert C into* C_{set}

2-4 行目が X-means によるクラスタリング，5-7 行目が各クラスタに対するΔの計算と評価である．f-1 から f-6 行目では，再帰的なクラスタ分割処理を行う関数 splitClusterRecursively の処理を表す．提案アルゴリズムでは，まず，入力データセット P に対する最初の分割をクラスタ数$k = k_0$ (特に指定が無ければ 2)として行い，分割結果それぞれのクラスタに対し再帰分割関数 splitClusterRecursively を適用する．splitClusterRecursively では，入力されたクラスタを 2 分割し，分割前と分割後どちらが，より有り得るクラスタモデルであるか情報量規準BICを用いて比較する．分割前のBICが分割後のBIC′より大きければ，分割後のモデルがより好ましいと判断し，分割後のクラスタそれぞれに対し再び splitClusterRecursively を適用していく．もし，そうでなければ分割前のモデルがより好ましいと判断して分割を終了し，分割前のクラスタを結果クラスタの 1 つとして確定する．各クラスタの評価においては，Gap 統計量を用いたアルゴリズム同様，各クラスタに対しΔを計算していき，その値が最小となるクラスタを保持，最終的に保持していたものを最適グループとする．

4. 実験

実装は C++で行われ，k-means++はオープンソースのライブラリである OpenCV の実装を用いた．実験は 3.4GHz Intel Core i7 のプロセッサ及び 16GB 2133MHz DDR4 のメモリーで構成される PC 上で行った．

データセットは，実データと合成データ両方を用意した．実データは US Census Bureau の TIGER project ページで入手可能な座標データであり，名前を KL (1957 データ)と ML (1510 データ)とする．合成データは，サイズが 1K/3K/5K/7K/9K/10K/50K/100K/150K/200K の一様分布の乱数データである UN と，サイズが 1K/3K/5K/7K/9K でクラスタ数が 1/5/10/15/20，サイズが 10K/50K/100K/150K/200K でクラスタ数が 1K/2K/3K/4K/5K となるように分布させた RN を用いた．

本実験では，Gap 統計量を用いた提案アルゴリズムに対してはデータサイズ，データ分布(クラスタ数)，ブートストラップ回数の 3 要素の変更による提案アルゴリズムの実行時間の変化を，X-means を用いた提案アルゴリズムに対してはデータサイズ，データ分布(クラスタ数)による変化を計測した．

4.1. データサイズによる実行時間の変化

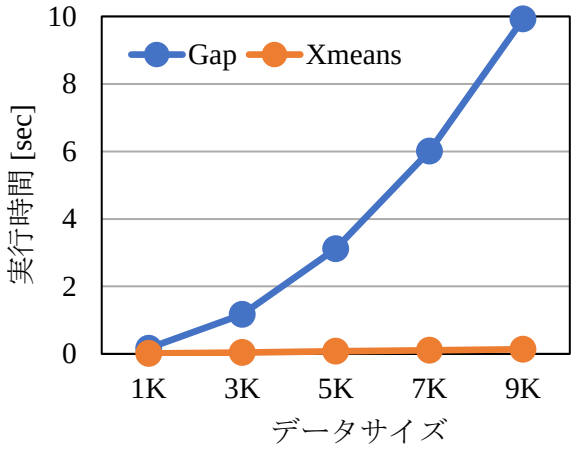

図 4.1: データサイズによる実行時間の変化

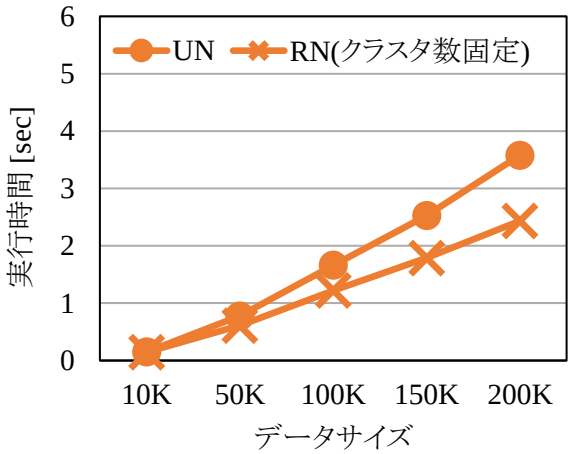

図 4.2: X-means を用いた提案アルゴリズムにおける
データサイズによる実行時間の変化

用いたデータセットは UN データ，RN データである．図 4.1 は Gap 統計量を用いた提案アルゴリズム，X-means を用いた提案アルゴリズム双方における 1K/3K/5K/7K/9K の UN データを入力としたときの実行時間の変化を表す．図 4.2 は 10K/50K/100K/150K/200K の UN データ，及び，クラスタ数がおおよそ 2K となるよう分布させた RN データを入力としたときの実行時間の変化を表す．

図 4.1 より，Gap 統計量を用いた提案アルゴリズムの実行時間はデータサイズの二乗に比例して増加することがわかる．これは，Gap 値計算時の評価指標 W_k を求める際に，各クラスタ内サンプル間すべての組み合わせ距離を計算することが大きく影響している．また，X-means を用いた提案アルゴリズムは Gap 統計量を用いたものと比較して明らかに高速であり，図 4.2 より，データサイズに対し線形に増加する．ここで，図 4.2 でクラスタ数固定の RN データを UN データに併せて用いているのは，データが一様に分散した UN データセットすべてに対し Gap 統計量を用いたクラスタリングが一定のクラスタ数の結果となったのに対し，X-means では多く分割される傾向にあり，データセットそれぞれに対しデータサイズに比例して増加したクラスタ数の結果となったため，クラスタ数の増加による影響を可能な限り排除して比較できる RN データを用いる必要があると考えたことによる．

4.2. データ分布(クラスタ数)による実行時間の変化

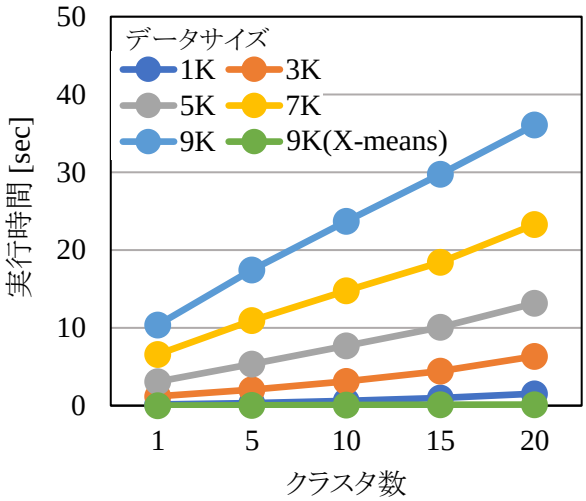

図 4.3: データ分布(クラスタ数)による
実行時間の変化

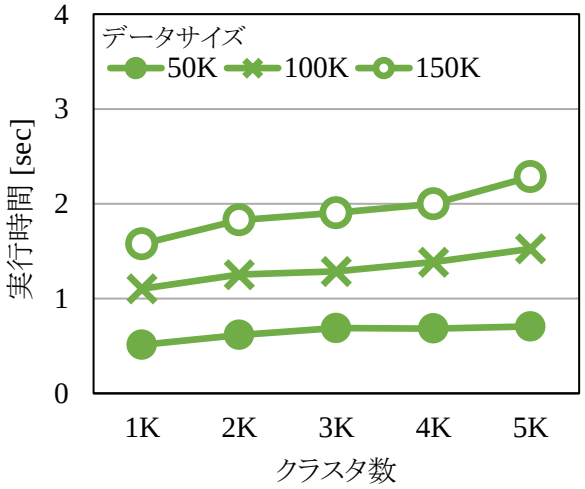

図 4.4: X-means を用いた提案アルゴリズムにおける
データ分布(クラスタ数)による実行時間の変化

用いたデータは RN データである．図 4.3 は Gap 統計量を用いた提案アルゴリズムにおいて，データサイ

ズ 1K/3K/5K/7K/9K でクラスタ数がおおよそ 1/5/10/15/20 となるよう分布させた RN データを入力として与えた場合と，X-means を用いた提案アルゴリズムにおいて，データサイズ 9K でクラスタ数がおおよそ 1/5/10/15/20 となるよう分布させた RN データを入力として与えた場合の実行時間の変化を表す．図 4.4 は X-means を用いた提案アルゴリズムにおいて，データサイズ 50K/100K/150K でクラスタ数がおおよそ 1K/2K/3K/4K/5K となるよう分布させた RN データを入力として与えた場合の実行時間の変化を表す．

図 4.3 より，Gap 統計量を用いた提案アルゴリズムでは，データ分布(クラスタ数)を変更した場合，実行時間はおおよそ線形に増加することがわかる．また，データ分布による増加幅よりも，データサイズによる増加幅が大きく，データサイズほど強い影響はないことが読み取れる．また，X-means を用いた提案アルゴリズムは Gap 統計量を用いたものと比較して明らかに高速であり，図 4.4 より，クラスタ数に対しゆるやかに増加している．増加量が安定でない (例えば，150K のデータセットにて，クラスタ数が 3K から 4K に増加する場合に比べ，4K から 5K に増加する場合の増加幅が大きい) のは，同じクラスタ数でも，細かいデータ分布の違いによって，分割時の k-means の処理時間に差が出るためである．なお，Gap 統計量を用いた提案アルゴリズムに関して，クラスタ数 20 までしか実験を行っていないのは，提案アルゴリズムの Gap 統計量が局所的な解を求めているため，20 より大きいクラスタ数のデータセットに対しては望ましいクラスタ数の結果とすることが困難であったことによる．

4.3. ブートストラップ回数による実行時間の変化

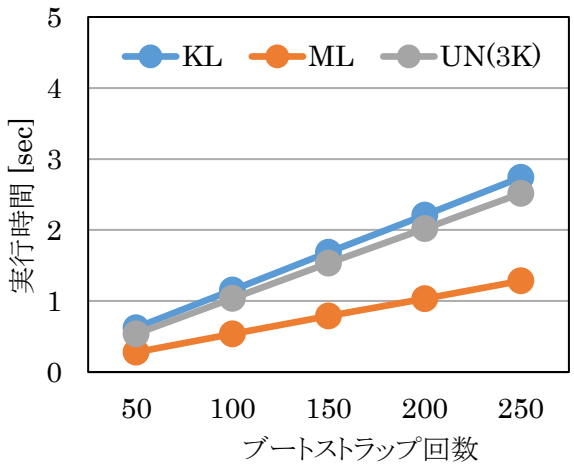

図 4.5: ブートストラップ回数による実行時間の変化

用いたデータは KL，ML，サイズが 3K の UN データであり，図 4.5 は Gap 統計量を用いた提案アルゴリズムにおける，それぞれのデータセットを入力とした場合の実行時間の変化である．

図 4.5 より，Gap 統計量を用いた提案アルゴリズムでは，ブートストラップ回数を変更した場合，実行時間は線形に増加することがわかる．なお，サイズ 1957 の KL の実行時間がサイズ 3K の UN の実行時間より大きくなっているのは，UN よりも KL のほうが，クラスタリング結果のクラスタ数が大きいことが影響している．

4.4. 考察

Gap 統計量を用いた提案アルゴリズムの実行時間はデータサイズの 2 乗に比例して増加し，データ分布(クラスタ数)，ブートストラップ回数の変更による変化は線形であった．データ分布による実行時間の変化(4.3 参照)においても，データサイズが大きいほど実行時間が急激に増加していることから，データサイズによる影響が他の要素による影響よりも大きいことがわかる．一方，X-means を用いた提案アルゴリズムの実行時間はデータサイズ，クラスタ数ともに線形に増加し，クラスタ数においては特にゆるやかな増加であり，双方において Gap 統計量よりも明らかに高速であった．また，X-means では Gap 統計量を用いたアルゴリズムと比較してクラスタリング結果のクラスタ数が多くなる傾向があった．Gap 統計量を用いた提案アルゴリズムでは，大きいクラスタ数のデータセットは正確に処理できないものの 1 から 20 程度の小さいクラスタ数のデータセットをほぼ正確なクラスタ数にクラスタリングできたのに対し，X-means を用いた提案アルゴリズムでは少ないクラスタ数のデータセットを正確に分割することが難しく，本実験で用いたサイズ 50K から 200K 程度のデータセットにおいては数百から数千の数の，多数のクラスタを持つデータセットに対しては正確なクラスタ数に近いクラスタリングを行えていたように感じる．

以上より，Gap 統計量を用いた提案アルゴリズムはデータサイズの影響を受けやすい一方で，他の要素による影響は小さく，小規模かつ少ないクラスタ数のデータセットであれば高速かつ正確な処理を見込めると思われる．X-means を用いた提案アルゴリズムでは，少ないクラスタ数のデータセットのクラスタリングを正確に処理することは難しいものの，多数のクラスタをもつデータセットに対しては正確で，Gap 統計量によるアルゴリズムよりもさらに高速な処理を見込める．

5. まとめ

本研究では，グループのサイズだけでなくグループ

内データ個数も指定せずクエリ点に近いコンパクトなグループを検索する DNNH (Dense Nearest Neighborhood) 問題を提案した. これにより, ユーザーの望む結果とならない可能性があった既存問題の課題が改善される. また, Arthur らの k-means++ と Tibshirani らの Gap 統計量, Pellog らの X-means を用いたクラスタリング, そして独自に定義したクラスタの評価関数による, DNNH 問題を解く新しい検索アルゴリズムを提案した. Gap 統計量を用いた提案アルゴリズムは, 小規模かつ少ないクラスタ数のデータセットに対しては高速かつ正確に処理でき, X-means を用いた提案アルゴリズムは, 多数のクラスタを持つデータセットに対しては正確で, 比較的大きな規模のものにおいても高速な処理が見込める.

今後の課題としては, アルゴリズムのさらなる効率化によってデータサイズによる影響を軽減することを検討している. 本研究の提案手法は, 直感的に考え, データセット全体をクラスタリング後に各クラスタの評価を行う, という方法をとっているが, 実際に解となるクラスタはクエリ点に近い, データセットのほんの一部である可能性が高く, 全体をクラスタリングすることは非効率である. したがって, クエリ点に近いデータから, 所属するクラスタを検索しつつ, そのクラスタの評価を行うといったアルゴリズムや, クラスタリング処理中にフィルタリングを加え, 処理対象のデータを制限しサイズを減らすアルゴリズムなどのほうが, この問題の解法として適していると思われる.

謝辞

本研究は JSPS 科研費 JP19K12114 の助成を受けたものである.

参 考 文 献

[1] D.-W. Choi and C.-W. Chung, “Nearest neighborhood search in spatial databases,” in Data Engineering (ICDE), 2015 IEEE 31st International Conference on. IEEE, 2015, pp. 699–710.

[2] S. Le, Y. Dong, H. Chen, K. Furuse. “Balanced Nearest Neighborhood Query in Spatial Database,” in Big Data and Smart Computing (BigComp), 2019 IEEE International Conference on. IEEE, 2019. [Online]. Available: https://doi.org/10.1109/BIGCOMP.2019.8679425

[3] D. Papadias, Q. Shen, Y. Tao, and K. Mouratidis, “Group nearest neighbor queries,” in Proceedings of the 20th International Conference on Data Engineering, ICDE 2004, 30 March - 2 April 2004, Boston, MA, USA, 2004, pp. 301–312. [Online]. Available: http://dx.doi.org/10.1109/ICDE.2004.1320006

[4] D. Papadias, Y. Tao, K. Mouratidis, and C. K. Hui, “Aggregate nearest neighbor queries in spatial databases,” ACM Trans. Database Syst., vol. 30, no. 2, pp. 529–576, 2005. [Online]. Available: http://doi.acm.org/10.1145/1071610.1071616

[5] K. Deng, S. W. Sadiq, X. Zhou, H. Xu, G. P. C. Fung, and Y. Lu. “On group nearest group query processing,” IEEE Trans. Knowl. Data Eng., vol. 24, no. 2, pp. 295–308, 2012. [Online]. Available: https://doi.org/10.1109/TKDE.2010.230

[6] Wikipedia contributors, “k-means++法 - Wikipedia” 2019.[Online: accessed 5-November-2019]. Available: https://ja.wikipedia.org/wiki/K-means%2B%2B%E6%B3%95

[7] R. Tibshirani, G. Walther, T. Hastie. “Estimating the number of clusters in a data set via the gap statistic,” journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 63, no. 2, pp. 411-423, 2001. [Online]. Available: https://doi.org/10.1111/1467-9868.00293

[8] D. Pelleg, A. Moore. “X-means: Extending K-means with Efficient Estimation of the Number of Clusters,” 2000. [Online]. Available: https://www.researchgate.net/profile/Dan_Pelleg/publication/2532744_X-means_Extending_K-means_with_Efficient_Estimation_of_the_Number_of_Clusters/links/0deec525a4992012a6000000/X-means-Extending-K-means-with-Efficient-Estimation-of-the-Number-of-Clusters.pdf

[9] 石岡 恒憲. “クラスタ数を自動決定する k-means アルゴリズムの拡張について,” 2000. [Online]. Available: http://www.rd.dnc.ac.jp/~tunenori/doc/xmeans_euc.pdf

[10] M. Maechler. “clasGap function,” R Documentation. [Online: accessed 29-October-2019]. Available: https://www.rdocumentation.org/packages/cluster/versions/2.1.0/topics/clusGap