筑波大学大学院博士課程

システム情報工学研究科修士論文

最適化に基づくサイズ均等クラスタリング手法に関する研究

平野 翼

修士（工学）

（リスク工学専攻）

指導教員　遠藤 靖典

2016年3月

概要

クラスタリングはデータ解析に用いられる手法であり，個体の集合であるデータセットに対し教師なし分類を行う．この手法はデータマイニングを始めとして様々な場面で利用されている．中でも K-member Clustering (KMC) は各クラスタが少なくとも K 個の個体を持ち，クラスタ内距離の総和を最小にするクラスタリング手法であり，情報セキュリティにおいて重要な手法である K-匿名化への応用が期待されている．しかしながら，KMC の既存手法は不自然な分類が行われることが多く，K-匿名化を行う際の情報損失が大きくなってしまうことが問題であった．この原因として既存手法は最適化に基づいてないということが挙げられる．クラスタリングの代表的手法である Hard c-means (HCM) や Fuzzy c-means は目的関数の最適化を行っており，KMC においても最適化を用いることでより良い分類結果を得ることが期待できる．一方，クラスタサイズの制約について，均等な大きさに分割することを考える．これはサイズ均等クラスタリングと呼ばれ，K-匿名化にも応用できる他，運送エリアの分割問題やタスク分配の問題等，応用の幅は広い．

以上の背景から，最適化に基づくサイズ均等クラスタリング (Even-sized Clustering Algorithm Based on Optimization; ECBO) が提案されている．これは HCM の目的関数および制約式に関し，クラスタサイズを均等にする制約式を加え，目的関数の最適化を行うことによりサイズを均等にするクラスタリングを行う手法である．

ECBO は HCM が基となっているため，初期値問題や外れ値に対するロバスト性，線形分離となることが問題となる．20 世紀後半から研究されてきたクラスタリング手法全体をみれば，このような問題に対応する手法は多く存在している．そこで，本研究ではこれらの問題に対し，既存のクラスタリング手法の考え方やアルゴリズムを利用したサイズ均等クラスタリングの手法を 4 つ提案する．さらに，複数のデータセットに対する数値例を通してこれらの手法について考察する．

目 次

図 目 次

表 目 次

第1章　序論

1.1　背景

近年のインターネットの普及やコンピュータの性能の向上によって，膨大な量のデータの収集・蓄積が非常に容易となっている．コンビニエンスストアでの購買データや，スマートデバイスによる行動履歴データといった，不特定多数のユーザーから集められるこのようなデータはビッグデータと呼ばれ，多くの企業でサービスの改善や効率化のために利用されている．さらに，Internet of Things (IoT) の浸透によりスマートデバイスが身の回りにあふれるようになると，そのようなデータの量は更に膨大なものとなることが予想される．一方で我々が直接これらのデータを扱うことは非常に困難となっている．そこでデータマイニングが用いられている．これは大量のデータを網羅的に解析することにより，有用な情報や，想像の及びにくい新たな知見を得ることができる手法である．

クラスタリング [1] はそのようなデータ解析手法の 1 つである．これは個体の集合であるデータセットをクラスタと呼ばれる集合に分割する手法である．教師データを与えることなく，個体同士に定義された類似度や非類似度に基づき，類似する個体同士を自動的に分割するという特徴を持つ．20 世紀後半から盛んに研究されているこの手法は，現在までに実に様々なアルゴリズムが考案されており，その応用も多岐にわたっている．代表的な手法である c-平均法 [2] (Hard c-means; HCM) は J.MacQueen によって一般的になったアルゴリズムである．これはクラスタ中心として重心を取り，クラスタ内の個体との距離の総和を最小とするよう最適化を行う．また，凝集型階層的クラスタリング (Agglomerative Hierarchical Clustering; AHC) は最初に一つ一つの個体をクラスタとし，最も近いクラスタ同士を結合することで大きなクラスタを作っていくアルゴリズムである．出力はデンドログラムと呼ばれる二分木の形となり，クラスタの結合過程がわかる他，クラスタの結合レベルから分割を得ることもできる．他にも，個体が球面上にあるデータに対して用いられる球面クラスタリングや，カーネルを用いて非線形な分割を得ることのできるカーネルクラスタリングなどがある．

ところで，個人情報の取り扱いに際して，プライバシー保護の配慮は必要不可欠なものとなっている．2005 年に施行された「個人情報の保護に関する法律」は，2015 年には改正が行われ，ビッグデータとしての利用がしやすくする一方で罰則が強化される改正案が可決されている．したがってビッグデータの活用の際にはプライバシー保護について大きな責任が伴うこととなった．また，ビッグデータ活用時はプライバシー保護のため，匿名化が行われる．匿名加工情報に関して，米国では 2003 年に Health Insurance Portability and Accountability Act (医療保険の相互運用性と説明責任に関する法律) として医療情報の二次利用の際のガイドラインを策定している．韓国でも 2014 年にビッグデータのプライバシーガイドラインを策定し

ており，匿名化に対する関心は世界的にも高まっていると言える．

匿名化には K-匿名化という手法があり，これは年齢や住所等の個人を特定することができる情報をあいまいな情報へと変換し，同一のデータが K 個以上存在させるようにする．これによって外部のデータとの照合によりある個人を特定しようとする際には，少なくとも K 個の候補が挙がることになるため特定は困難となる．しかしながら，データを加工するため，匿名化の際は情報の損失が伴う．ビッグデータの解析のためにはこの情報損失をできるだけ抑えることが望ましい．そこで，似ている個体同士を集め，最低限のデータ加工を行うことで情報損失を小さくする必要がある．ゆえに，クラスタサイズを K 以上とするクラスタリングを用いることで妥当な K-匿名化を行うことができる．

また，運送エリアの分割問題を考える．あるエリアに運送の拠点を複数設置しようとしているが，運送の距離を最小とし，かつ各拠点の仕事量を均等にするエリア分割を行うことを仮定する．この場合は距離の近い住宅同士でグループに分割することと同時に，仕事量すなわち運送対象となる住宅の数を均等にすることが求められる．そして，このような分割を行えば，分割された住宅の重心となる座標へと運送拠点を配置することによって最適な運送を行うことができる．

クラスタサイズに着目した手法の一つにサイズ調整変数クラスタリング [3] がある．ファジィ c-平均法 (Fuzzy c-means; FCM) によるクラスタリングは各クラスタに所属するクラスタの数が近くなる傾向がある．そのため，クラスタが同程度のサイズとなるようなデータセットならば良いクラスタリング結果が得られやすい．しかし，サイズが大きく異なるクラスタが存在するものや個体の分布が偏っている場合には，大きいクラスタの外側の一部が小さいクラスタに分類されることがある．そこで，サイズ調整変数を導入したクラスタリング手法が提案されている．これは FCM の目的関数にサイズ調整変数 $\alpha = \{\alpha_1, \ldots, \alpha_c\}$ を導入した手法で，クラスタ C_i に対応した変数 α_i の大きさによってクラスタサイズが変化する．計算は FCM と同様のアルゴリズムで，α についても最適化を行い，各クラスタのサイズを調整している．しかしながら，クラスタサイズの調整は自動的に最適化するため，指定することはできない．

所属するデータ同士が類似し，かつサイズが均等であるクラスタに分割することは様々なケースで利用される．そこでクラスタに属する個体数を均等にするクラスタリング手法が提案されている．これはクラスタ内の個体数またはクラスタ数を指定し，クラスタ内の個体数が均等かつクラスタ内距離の総和が最小となるようにクラスタリングを行う手法である．アルゴリズムは Extended Two-division Clustering for K-anonymity of Cluster Maximization [4] (E2DCKM) や最適化に基づくサイズ均等クラスタリング [5] (Even-sized Clustering Algorithm Based on Optimization; ECBO) が提案されている．E2DCKM は最初にデータセット全体を 1 つのクラスタとし，2 つのクラスタへの分割と個体数の調整を繰り返し，均等なクラスタ分割を求める手法である．しかし，最適化に基づいていないため分類精度はあまり高くはない．

代表的なクラスタリングアルゴリズムであり，広く利用されている HCM や FCM は目的関数の最小化を基本として構成されている他，スペクトラルクラスタリングも最適化の範疇で構成されたものである．このように，最適化を基本にアルゴリズムを構成することは分類精

度の問題に対する解決策の一つである．

そこで，目的関数最適化に基づいた手法である ECBO が提案された．ECBO は目的関数および制約式を設定し，個体の分類とクラスタ中心の交互最適化を行うことにより，均等なサイズのクラスタを生成する．その際に最適化手法の 1 つであるシンプレックス法を用いており，高い分類精度を持っている．しかしながら，外れ値の存在や初期値の与え方によって結果に悪影響が出やすいという点が問題となっている．

1.2 目的

本研究では ECBO に対し，他のクラスタリング手法の考え方やアルゴリズムを利用した新たな手法をそれぞれ提案することによって諸問題の解決を図る．利用する手法は k-means++, L_1 Fuzzy c-means, k-medoids, Kernel Hard c-means の 4 手法である．k-means++は FCM の初期値問題の解決策として考案された手法である．最初に求めるクラスタ中心が偏らないよう確率的に決定し，その後通常の FCM のアルゴリズムに従ってクラスタリングを行う．

L_1 Fuzzy c-means は距離尺度としてユークリッド 2 乗距離の代わりに L_1 ノルムと呼ばれる距離によって非類似度を定義したクラスタリングである．ユークリッド 2 乗距離と比較して，外れ値の影響を受けにくいことが利点として挙げられる．k-medoids はクラスタ中心を個体で表すクラスタリング手法である．個体同士の距離さえ求める事ができれば実行することができるため，グラフのクラスタリング等にも用いることができる．Kernel Hard c-means はカーネル関数を用いてデータを高次元の特徴空間上に写像してクラスタリングを行う手法である．従来の HCM の手法では境界面が線形であるため，非線形な特徴を持つデータに対して最適な分類を行うことができなかった．しかし KHCM は特徴空間上で処理を行うことにより，実空間上では非線形な分類を行うことができる．

これらの手法を導入した提案手法の実行結果を人工データを用いて比較・評価し，分類の傾向などをそれぞれの手法についての考察も行う．

1.3 本論文の構成

本論文では，第 2 章でサイズ均等クラスタリングの基礎であるクラスタリング手法と関連手法，新たなサイズ均等クラスタリングに用いるクラスタリング手法について述べる．第 3 章ではサイズ均等クラスタリングの中でも最適化を利用し，より良い分割を行うことができる，ECBO について述べる．また，この基となる手法である K-member Clustering についての説明も行う．第 4 章では，ECBO に他のクラスタリング手法の考え方を導入した 4 つの手法の提案を行う．第 5 章では人工データを用いた数値例を示し，評価および考察を行う．最後に第 6 章において，本研究で得られた結論を述べる．

第2章　種々のクラスタリング手法

この章ではまず，クラスタリングについての概要を説明する．その次に，現在提案されている様々なクラスタリング手法について，特徴とアルゴリズムを説明する．

2.1　クラスタリングについて

クラスタリングは個体の集合であるデータセットをクラスタと呼ばれる分割に分類するデータ解析手法である．

この手法は個体同士に定義された類似度や非類似度に基づいており，教師データを与える必要なく，自動的に分類が行われる．

クラスタリングは階層的手法と非階層的手法の 2 つに大別される．階層的手法は凝集型の手法と分割型の手法がある．凝集型階層的手法は最も類似する小さなクラスタ同士を逐次併合していく手法である．分割型階層的手法は最初にデータセット全体を 1 つのクラスタにし，逐次分割したクラスタを生成していく手法である．凝集型の手法に対して，クラスタの分割には計算量が膨大にかかってしまうため，分割型の手法は現在あまり用いられていない．これらの結果はクラスタの併合が樹形図として出力される．樹形図を利用することにより，データセット全体を俯瞰して，クラスタ内での個体の階層的な位置づけを確認できる．また，樹形図において適当な類似度を指定することにより，任意のクラスタ数へとデータセットを分割することができる．一方，非階層的手法は単にデータセットの分割を行うものである．しばしば目的関数が設定され，その値を最適化するようにデータセットの分割が行われる．階層的手法と比較して，大きなデータセットでも計算量の増加が抑えられることも特徴の一つである．

最適化に基づくサイズ均等クラスタリングは非階層的手法であり，目的関数を設定し，その最適化により均等なサイズのクラスタを得る手法である．したがって，この手法での利用を念頭に，非階層的手法を中心に紹介していく．

以下の説明では，データセット X に含まれる個体を $x_k = (x_{k1}, \ldots, x_{kp})\ (k = 1, \ldots, n)$, クラスタ中心の集合を V とし，クラスタ中心を $v_i\ (i = 1, \ldots, c)$ と表す．また，$C_i (i = 1, \ldots, c)$ はクラスタを表す．また，帰属度 u_{ki} は個体 x_k がクラスタ C_i に所属している度合いを表し，帰属度行列を $U = (u_{ki})$ とする．

2.2 Hard c-means

Hard c-means (HCM) は最も基本的なクラスタリング手法である．各個体とそれが所属するクラスタの中心との距離（クラスタ内距離）の総和を最小とする分割を求める．そこで，目的関数 J_{HCM} を以下のように定義し，J_{HCM} の最小化を行う．

$$J_{\mathrm{HCM}} = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ki}\|x_k - v_i\|^2.$$

ここで，帰属度 u_{ki} は，個体 x_k がクラスタ C_i に属する場合は $u_{ki} = 1$，そうでない場合は $u_{ki} = 0$ とする．また，U に関する制約として

$$\sum_{i=1}^{c} u_{ki} = 1 \quad (k = 1, \cdots, n)$$

を与える．この制約により，個体 x_k はただ一つのクラスタ C_j に対して $u_{kj} = 1$ となり，その他のクラスタについては $u_{ki} = 0, \forall i \neq j$ となる．

手順としては，最初に c 個のクラスタ中心を選ぶ．その後，各個体の所属の最適化と，クラスタ中心の最適化を交互に行う．

所属の最適化では，各個体を最も近いクラスタに所属させることで行うことができる．

また，クラスタ中心の最適化では，目的関数の v_i についての導関数を求めることにより，

$$v_i = \frac{\sum_{k=1}^{c} u_{ki}x_k}{\sum_{k=1}^{c} u_{ki}}$$

と求めることができる．これはクラスタ C_i の重心を表している．

以上の交互最適化を解が収束するまで行う．

クラスタリングの結果は超球状のクラスタが得られる．また，クラスタのサイズは均等に近くなる傾向にあるため，小さいクラスタと大きいクラスタが混在する場合は自然な結果を出力しづらい．

HCM のアルゴリズムを Algorithm 1 に示す．

Algorithm 1 HCM

Step 1. ランダムに個体を c 個選び，それぞれをクラスタ中心 v_i とする．

Step 2. 各個体と各クラスタ中心の距離を計算し，各個体を最も近いクラスタに所属させる．

Step 3. V を更新する：クラスタの重心を再計算し，新たなクラスタ中心とする．

Step 4. 分類が前回と変化しなければ終了．そうでなければ Step 2. に戻る．

2.3 Fuzzy c-means

HCM の帰属度行列 U は個体がクラスタに属するか属さないかの 2 値のみしか表すことができず，ハードクラスタリングやクリスプなクラスタリングと呼ばれる．対して，Fuzzy c-means (FCM) は $u_{ij} \in [0, 1]$ の実数値を取るようにし，属するか属さないかを程度で扱うことにより，柔軟なクラスタリングを行う．このようなクラスタリングはファジィクラスタリングと呼ばれる．標準的な FCM は目的関数を以下のように定義する．

$$J_{\mathrm{FCM}} = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \|x_i - v_j\|^2.$$

ここで，m はファジィ化パラメータであり，$m > 1$ である．

この目的関数を U と V の交互最適化によって最小化することにより，結果を求める．

U の最適化については，V を固定し，ラグランジュの未定乗数法を用いて，

$$u_{ki} = \left\{ \sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|^2}{\|x_k - v_j\|^2} \right)^{\frac{1}{m-1}} \right\}^{-1} \tag{2.1}$$

と求めることができる．

V の最適化については，J_{FCM} を v_i について偏微分することにより，

$$v_i = \frac{\sum_{k=1}^{n} (u_{ki})^m x_k}{\sum_{k=1}^{n} (u_{ki})^m} \tag{2.2}$$

である．この v_i はクラスタ C_i の重み付き重心を表している．

以上のアルゴリズムを Algorithm 2 に示す

Algorithm 2 FCM

Step 1. ランダムに個体を c 個選び，それぞれをクラスタ中心 v_i とする．

Step 2. U を更新する： V を固定し，式 2.1 より求める．

Step 3. V を更新する： U を固定し，式 2.2 より求める．

Step 4. 前回と変化しなければ終了．そうでなければ Step 2. に戻る．

2.4 k-means++

k-means++ [6] は従来の FCM の初期値選択を改善した手法である．FCM は初期値選択をランダムに行っていたが，k-means++はすでに選ばれたクラスタ中心からの 2 乗距離に基づいた

確率を用いて行う．ここで用いる偏りの少ない初期値を用いて FCM を実行することにより，計算量や分類結果の改善が見込まれる．このことは David ら [6] によって証明されている．

$D(x)$ を個体 x とすでに選択した最も近いクラスタ中心との距離とする．初期値選択は最初のクラスタ中心を 1 つだけ，ランダムに選択した後，以下の確率で 1 つずつ個体を選ぶ．

$$\frac{D(x)^2}{\sum_{x\in X} D(x)^2}. \tag{2.3}$$

このように初期値選択を行うことで，データセット上において偏りの少ないようにクラスタ中心を選択することができる．このアルゴリズムを Algorithm 3 に示す．

Algorithm 3 k-means++

Step 1a. ランダムに個体を 1 つ選び，クラスタ中心 v_1 とする．

Step 1b. 式 2.3 の確率で個体を 1 つ選び，新しいクラスタ中心 v_i とする．

Step 1c. c 個のクラスタ中心を選ぶまでステップ 1b. を繰り返す．

Step 2-4 通常の FCM を行う．

2.5 L_1 Fuzzy c-means

L_1 ノルムはマンハッタン距離とも呼ばれ，碁盤状に整備された市街地での移動距離を表すため，人間にとって自然な距離尺度の一つである．また，データ解析等においてもよく用いられている．この距離尺度は 2 点の各座標の距離の差の総和であり，以下の式で表される．

$$\|\boldsymbol{x}-\boldsymbol{y}\|_1 = \sum_{j=1}^{p} |x_j - y_j|.$$

L_1 ノルムを用いたファジィクラスタリングは Jajuga [7] や Bobrowski ら [8] によって提案されている．この手法は，一般的なクラスタリングで用いられるユークリッド 2 乗距離に対して，外れ値に対するロバスト性が高いことが利点として挙げられている．しかしながら，これらのアルゴリズムはクラスタ中心を求める際に計算量が多くかかってしまうことが問題となっていた．その後，宮本ら [9] によってクラスタ中心の効率的な再計算方法が提案されている．本節ではこの宮本らの手法について説明する．

FCM 同様にクラスタ中心 V と帰属度 U の交互最適化により目的関数を最適化する．目的関数は以下のものを用いる．

$$J_{\mathrm{L}_1}(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ki})^m \|x_k - v_i\|_1.$$

V を固定した場合の U の最適化については，ラグランジュの乗数法を用れば，

$$u_{ki} = \frac{\|x_k - v_i\|_1^{-\frac{1}{m-1}}}{\sum_{l=1}^{c} \|x_k - v_l\|_1^{-\frac{1}{m-1}}}, \tag{2.4}$$

と最適解 U を求めることができる．

U を固定した場合の V の最適化では，V の各クラスタおよび各成分で独立に最適化を行うことができる．そこで準目的関数 J_{ij} を次のように定義する．

$$J_{ij} = \sum_{k=1}^{n} (u_{ki})^m |x_{kj} - v_{ij}|.$$

すると，目的関数は次のように表すことができる．

$$J_{L_1} = \sum_{i=1}^{c} \sum_{j=1}^{p} J_{ij}.$$

ここで定義した準目的関数 J_{ij} が全ての i, j において最適であれば，もとの目的関数 J_{L_1} が最適となる．また，J_{ij} は x_{kj} 以外の点では微分可能な凸関数である．したがって，x_{kj} を除いて微分した $\frac{\partial J_{ij}}{\partial v_{ij}}$ の符号が負から正に変わるときが最適解となる．

この解を探索する準備として，各個体の j 成分 $\{x_{1j}, \ldots, x_{nj}\}$ を昇順にソートし，$x_{q_j(1)j} \leq x_{q_j(2)j} \leq \cdots \leq x_{q_j(n)j}$ とする．ここで，$q_j(k)(k = 1, \ldots, n)$ は $\{1, \ldots, n\}$ の置換であり，もとの個体の 1 つ目の添字，すなわち個体の番号と対応している．$x_{q_j(k)}$ を用いれば，J_{ij} は，

$$J_{ij} = \sum_{k=1}^{n} (u_{ki})^m |x_{kj} - v_{ij}| = \sum_{k=1}^{n} (u_{q_j(k)i})^m |x_{q_j(k)j} - v_{ij}|$$

と表すことができる．

ここで，$x_{q(k)j}$ を用いれば，$\frac{\partial J_{ij}}{\partial v_{ij}}$ は次のように表される．

$$\frac{\partial J_{ij}}{\partial v_{ij}}(x_{q_j(r)j}) = -\sum_{k=r+1}^{n} (u_{q_j(k)i})^m + \sum_{k=1}^{r} (u_{q_j(k)i})^m.$$

この導関数が負から正に変わるときの節点 $v_{ij} = x_{q_j(r)j}$ が最適値である．

以上より，最適解 v_{ij} を求めるアルゴリズムを Algorithm 4 に，L_1 ノルムに基づいた FCM(FCM-L_1) のアルゴリズムは Algorithm 5 に示す．

2.6 k-medoids

k-means はクラスタ中心を各クラスタの重心としていたのに対し，k-medois はメドイドと呼ばれるクラスタ内を代表する個体をクラスタ中心とする手法である．そのため，k-means よ

Algorithm 4 最適解 v_{ij} の導出

1 x_{kj} を昇順ソートし，$x_{q_j(k)j}$ とする．

2 $S = -\frac{1}{2}\sum_{k=1}^{n}(u_{q_j(k)i})^m, r = 0$ とする．

3 S が負から正に変わるまで $S = S + (u_{q_j(k)i})^m, r = r + 1$ とする．

4 最適解 $v_{ij} = x_{q_j(r)j}$ を出力する．

Algorithm 5 FCM-L_1

Step 1. 初期値 V をランダムに与える．

Step 2. V を固定して J_{L_1} の最小化問題を解く：式 2.4 より最適解を U を更新する．

Step 3. U を固定して J_{L_1} の最小化問題を解く：Algorithm 4 より最適解 V を更新する．

Step 4. 解 U, V が収束すれば終了．そうでなければ Step.2 へ戻る．

りも，k-medoids の方が外れ値の影響を受けづらいことが知られている [10]．また，個体がクラスタ中心であるため，個体同士の距離さえ与えられればクラスタリングを行える．したがって，グラフデータ等にも適用することができる．ここで，距離尺度は通常のユークリッド距離を用いる．

初期の代表的なアルゴリズムに PAM (Partitioning Around Medoids) [11] や CLARA(Clustering LARge Applications) [11] がある．PAM は BUILD と SWAP の 2 ステップで構成されるアルゴリズムである．BUILD では初期のメドイドを選択するが，この際データセットの中心に近い k 個の個体が選択される．SWAP ではメドイドを更新するステップである．メドイドと，メドイドでない個体を全ての組み合わせについて探索しクラスタ内距離が最も小さくなるようなペアについて，メドイドを交換する．PAM は良い分類を得られる反面，計算量が非常に大きいという欠点がある．CLARA はこの欠点を解決するために提案されたアルゴリズムであり，SWAP ステップの際に全個体ではなくランダムにとったサンプルを新たなメドイドの候補とする．しかしながら，限られた個体がメドイドの候補となるため，クラスタリング結果は悪くなってしまう．

これらの手法に対して，計算量が小さく良いクラスタリング結果を得られるアルゴリズムが Park ら [10] によって提案されている．このアルゴリズムを Algorithm 6 に示す．ここで，距離尺度はユークリッド距離を利用する．ユークリッド距離は以下のように表される．

$$d(x_i, x_j) = \sqrt{\sum_{a=1}^{p}(x_{ia} - x_{ja})^2}, \quad (i, j = 1, \ldots, n).$$

この手法は k-means と同様に，メドイドの更新と個体の所属の更新を交互に最適化を行う．

メドイドの更新では，個体の所属を固定して，クラスタ内距離の総和を最小とする新たなメドイドを求める．すなわち，クラスタ C_i において，ある個体と C_i に属する全ての個体との距離の総和が最も小さい個体を新たなメドイドとすることを全ての i において行う．

$$v_i = \arg\min_{x_k} \sum_{x_j \in C_i} d_{kj}$$

所属の更新については，k-means と同様に，個体 x_k を最も近いメドイドを持つクラスタに所属させる．

また，初期値の選択は，データセットの中心に近いような個体を選択するため，以下の式を用いてどの程度中心に近いかを計算する．

$$w_j = \sum_{i=1}^{n} \frac{d_{ij}}{\sum_{l=1}^{n} d(x_k, x_j)}. \tag{2.5}$$

そして，最も w_j の値が小さい k 個を初期のメドイドとする．

この Park らの手法では，計算量はメドイドの更新で $O(n^2c)$，所属の更新で $O(n)$ となるため，全体として $O(n^2c)$ の計算量となる．HCM の計算量は $O(ncp)$ であり，大抵は $n > p$ であるため，反復回数を定数とみなせば，計算量については HCM の方が少ないといえる．

Algorithm 6 k-medoids

Step 1 初期値選択

1-1. 全ての個体同士の距離 d_{ij} を計算する

1-2. 全ての個体 x_j に関して，w_j を式 2.5 を用いて計算する．

1-3. w_j を昇順にソートし，最も小さい k 個の個体を初期のメドイドとして選択する．

1-4. 各個体を最も近いメドイドのクラスタに所属させる．

1-5. すべての個体とそれが所属するクラスタのメドイドとの距離の総和を求める．

Step 2 メドイドの更新

各クラスタの中で，ある個体とその他のクラスタに属する全ての個体との距離の総和が最も小さい個体を選び，新たなメドイドとする．

Step 3 個体の所属の更新

3-1. 各個体に最も近いメドイドのクラスタに所属させる．

3-2. 全ての個体とそれが所属するクラスタのメドイドとの距離の総和を求める．

2.7 Kernel Hard c-means

通常の HCM ではデータセットは超球状に分割され，境界は超平面となる線形分離の形となる．しかし，実データでは超球状でない形のものや線形分離ができないものも多い．そこで，元の空間では線形分離でないデータセットに対して，非線形に分割することができるカーネルクラスタリング (Kernel Hard c-means; KHCM) が提案されている [12]．これは，個体を元の空間よりも高次元の特徴空間に写像してクラスタリングを行うことにより，元の空間では非線形な分割を得る手法である．

特徴空間 F への写像を $\phi : \mathcal{R}^p \to F$ とする．すると，特徴空間上で HCM を行うような目的関数は

$$J_{\mathrm{KHCM}}(U, V) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ki} \|\phi(x_k) - v_i^{\phi}\|^2$$

となる．ここで，v_i^{ϕ} は特徴空間上でのクラスタ中心である．

データセットを特徴空間上に写像した個体 $\phi(x)$ を直接用いてクラスタリングを行うと，計算量が膨大となってしまう．そこで，一般的には特徴空間上のベクトルの内積を表すカーネル κ を用いて計算する．

$$\kappa(x, y) = \langle \phi(x), \phi(y) \rangle.$$

また，通常 ϕ は明示的に定義せず，カーネルを設定することによって，特徴空間への写像を行う．代表的なカーネルは以下の式で表される，ガウシアンカーネルである．

$$\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right).$$

ここで，σ はパラメータである．このようにカーネルを設定する場合，実際に計算を行って v_i^{ϕ} や実空間上での値を直接求めることはできない．そこで，以下のように変形することで個体とクラスタ間の距離を求める．

$$\begin{aligned} \left\| \phi(x_k) - v_i^{\phi} \right\|^2 &= \left\| \phi(x_k) - \frac{1}{|C_i|} \sum_{x_l \in C_i} \phi(x_l) \right\| \\ &= \kappa(x_k, x_k) - \frac{2}{|C_i|} \sum_{x_l \in C_i} \kappa(x_k, x_l) + \frac{1}{|C_i|^2} \sum_{x_l \in C_i} \sum_{x_m \in C_i} \kappa(x_l, x_m). \end{aligned} \quad (2.6)$$

また，上の式を用いるため，初期値選択でクラスタ中心を選ぶと，特徴空間上での距離を計算することができない．したがって，初期値選択のステップでは全ての個体をランダムにクラスタへ所属させることにする．以上のアルゴリズムを Algorithm 7 に示す.

Algorithm 7 KHCM

Step 1. 各個体をランダムにクラスタへ所属させる.

Step 2. 式 2.6 により，特徴空間上の各個体とクラスタ中心の距離を計算する.

Step 3. Step2. で得られた距離を用いて，各個体を最も近いクラスタに所属させる.

Step 4. クラスタ分類が前回と変化しなければ終了．そうでなければ Step 2. に戻る.

第3章　サイズ均等クラスタリング

この章ではサイズ均等クラスタリングについて説明する．まず，サイズ均等クラスタリングの基となっている手法として K-member Clustering について説明する．その後，サイズ均等クラスタリングの手法について，2 分割に基づくものと最適化に基づくものを説明する．

3.1　サイズ均等クラスタリングについて

サイズ均等クラスタリングは各クラスタの持つ個体数が均等であり，クラスタ内距離を最小とするようなクラスタリング手法である．これは，K-member Clustering を元として提案された手法である．

K-member Clustering は各クラスタのサイズが K 以上かつ，クラスタ内距離の総和が最小となるようにする手法である．これは K-匿名化への利用のために提案されたものである．K-匿名化は個人データを個体とするデータセットについてデータの加工を行うことにより，どのレコードに対しても，それと同一の属性をもつデータが K 個以上存在するデータセットに変換することである．この匿名化によって，外部のデータと照合して個人を特定しようとしても，K 人以上の候補が上がり，特定の 1 人を見つけることはできなくなる．従来の方法 [13–15] では，失われる情報の量が大きいことが問題となっていた．そこで，類似している K 個以上の個体の集合に分割するクラスタリング手法が提案された．クラスタリングの手法を利用することにより情報の損失を少なく，効率的に匿名化をすることができる．代表的な手法として，Greedy K-member Clustering (GKC), One-pass k-means Algorithm for K-anonymization (OKA), Clustering-based K-anonymity (CBK) の 3 つが挙げられる．

しかしながら，GCK や OKA にはクラスタのまとまりが良くないこと，CBK にはクラスタ数が最大化されず情報損失が大きくなりやすいという問題点があった．そこで Two-division Clustering for K-anonymity of Cluster Maximization (2DCKM) が提案された．この手法は CBK を基に，最終的なクラスタ数が最大となるよう，分割のたびに調整を行う手法である．

サイズ均等クラスタリングは，以上のような K-member Clustering に対し，さらにクラスタサイズの制約を強めたものである．クラスタサイズを K または $K+1$ とする制約を設定するが，この場合，データセットのサイズ n によっては，この制約を満たすことができない．したがって，最初に与えられた K に対し，制約を満たす値に調整することが必要となる．

最初に提案されたサイズ均等クラスタリング手法は，2DCKM をサイズ均等クラスタリングとして拡張したものである Extended Two-division Clustering for K-anonymity of Cluster Maximization (E2DCKM) である．まず，サイズに関する制約として最終的なクラスタサイズ

が K または $K+1$ となるように，与えられた K の値を調節する．その後，最終的なクラスタのサイズが制約を満たすように，クラスタの 2 分割とクラスタサイズの調整を繰り返す手法である．

ところで，クラスタリングの一般的な手法である HCM や FCM は最適化を基本としたものであり，スペクトラルクラスタリングも最適化の範疇で構成された手法である．このように，クラスタリングに最適化を用いることは非常に有効な手段である．しかしながらここで述べた K-member Clustering やサイズ均等クラスタリングでは，最適化を用いた手法は存在しなかった．そこで，サイズ均等クラスタリングに最適化を用いた，最適化に基づくサイズ均等クラスタリング (Even-sized Clustering Based on Optimization; ECBO) が提案されている．これは，HCM の目的関数および制約式に加え，サイズを K 以上，$K+1$ 以下とする制約を加える事により定式化したものである．最適化手法には線形計画法であるシンプレックス法を用いている．以上の方法により分類の精度が向上したほか，k-means のような交互最適化に基づく手法のアルゴリズムや考え方を取り入れることが容易となった．

K-匿名化への応用を主として提案された K-member Clsutering に対して，サイズ均等クラスタリングも K-匿名化への応用が可能な他，均等な分割を行う問題に対して応用する事ができる．例えば，ある地域に同じ規模の配送拠点を複数作ることを考える場合，配送の対象である住宅等の座標関してサイズ均等クラスタリングを行うことにより，効率的な配送地域の決定および配送拠点の配置を行うことができる．

3.2 K-member Clustering

K-member Clustering は n 個の個体を持つデータセットについて，各クラスタが少なくとも K の個体を持ち，クラスタ内距離の総和が最小となるようなクラスタの集合を探し出す問題である [16].

これは，K-匿名化への応用を目的として提案されているが，この対象となる個人データには年齢等の量的データの他に，住所等のカテゴリーデータも含まれている．また，クラスタリングによる K-匿名化の際にはデータ損失を最小化する必要がある．そのため，K-匿名化問題へ利用するために量的データおよびカテゴリーデータの距離尺度が定義されている．

量的データについては 2 つの値の差を取るのが最も自然な方法であり，K-member Clustering でもこの方法で距離を測る．ここで，匿名化の際には同じクラスタに属する個体の各属性について，代表的な値に置き換えられる．そこで，個々の属性で差を計算する．したがって属性 A における 2 つの値 $a_i, a_j \in A$ の正規化した距離を次のように定める．

$$\delta_N(a_i, a_j) = \frac{|a_i - a_j|}{|A|}.$$

ここで，$|A|$ は A の最小値と最大値の差を表す.

カテゴリーデータについては，量的データのように差を計算することはできない．1 つの解決法として，同じ値であるかどうかで判定する方法がある．例えば，全く同じ値の場合は 0，異なる場合は 1 とした距離を与えるという方法である．しかしながら,「住所」における

「県」,「市」といった階層構造のように，別の値であっても意味的な関係性を持つような属性は多くあり，このような属性についてはその関係に基づいた距離を定義することが望ましい．そこで，その属性については分類木によって距離を定義する．一方,「職業」の様に値同士で関係を持たない属性も存在する．そのような場合は一つの根の下にそれぞれの値として葉を持つ分類木を設定する．以上のように決めた分類木について，次のように正規化した距離を定義する．

$$\delta_C(a_i, a_j) = \frac{H(\Lambda(a_i, a_j))}{H(T_A)}.$$

ここで，$\Lambda(x, y)$ は x と y の共通の祖先を持つ最小の部分木，T_A は属性 A の分類木であり，$H(R)$ は分類木 T の高さを表す．

以上の量的データとカテゴリーデータの距離の定義を合わせて，2 つの個体の距離を定義する．データセット X において量的データを $N_s(s = 1, \ldots, g)$，カテゴリーデータを $C_t(t = 1, \ldots, h)$ とすると個体 x_i, x_j の距離は次のように表される．

$$\Delta(x_i, x_j) = \sum_{s=1,\ldots,g} \delta_N(x_i[N_s], x_j[N_s]) + \sum_{t=1,\ldots,h} \delta_C(x_i[C_t], x_j[C_t]).$$

ここで，$x_i[A]$ は x_i の属性 A の値を表している．

3.2.1 Greedy K-member Clustering

これは Byun ら [16] によって最初に提案された K-member Clustering である．従来の K-匿名化手法では匿名化によって失われる情報の量が大きいことが問題となっていた．これを，サイズが K 以上のクラスタに分割するというクラスタリング問題に帰着させ，効果的な K-匿名化を行えるようにしている．この手法では，中心となる個体を 1 つ選択し，それに最も近い個体 K 個を 1 つのクラスタとすることを繰り返す．そして，余った個体は最も近いクラスタに所属させるというアルゴリズムになっている．このアルゴリズムを Algorithm 8 に示す．

最初に選ばれる個体からはよくまとまったクラスタが得られる．しかしながら，ランダムに選ばれた個体に最も近い個体から選択されるため，クラスタが生成されるにつれまとまりの良くないクラスタとなりやすい．特に最後に得られるクラスタは大きく離れた個体同士が所属しやすくなってしまう．

3.2.2 One-pass k-means Algorithm for K-anonymization

One-pass k-means Algorithm for K-anonymization [17] (OKA) は One-pass k-means により通常のクラスタリングを行った後，クラスタサイズの調整を行う手法である．

手順としては，クラスタリングフェーズにおいて，まず全ての準識別子について個体をソートし，クラスタ数 c の設定を行った後，c 個の個体を選択し，それぞれを初期のクラスタとする．その後，データセット中のクラスタに所属していない最初の個体について，最も近いクラスタへ所属させることを全ての個体の所属が決定するまで繰り返す．調整フェーズでは，ク

Algorithm 8 Greedy K-member Clustering

Step 1. ランダムに個体を 1 つ選び r とする.

Step 2a. r から最も遠い個体を選択し，クラスタ C_i とする.

Step 2b. クラスタ C_i に最も近い個体を C_i に加える.

Step 2c. $|C_i| = K$ になるまで Step 2b. を繰り返す.

Step 3. クラスタに属していない個体の数が K 未満なら Step4. へ.
そうでなければ，C_i から最も遠い個体を r とし，Step2a. に戻る.

Step 4. クラスタに属していない個体をランダムに選択し，最も近いクラスタに所属させる.
これをクラスタに属していない全ての個体に対して行う.

ラスタリングフェーズで得られたクラスタの中で，サイズが K よりも大きいクラスタ C における，最もクラスタ中心から遠い $|C| - K$ 個の個体を一旦所属から外す．その後，所属から外された個体について，もしサイズが K よりも小さいクラスタが存在する場合，その中の最も近いものに割り当てる．そうでない場合は，全体のクラスタの中の最も近いものに割り当てる.

通常のクラスタリングを行ってから調整を行うため，GKC よりもまとまりの良いクラスターを得ることが期待される．しかしながら，クラスタリングフェーズにおいて，ランダムに初期クラスタを与えており，その後のクラスタへ 1 度割り当てた後は所属の改善がされない．したがって，初期クラスタの分布が偏ってしまうと，不自然な分割となってしまい，それを修正できないままとなってしまう．また，調整フェーズにおいては，サイズが K より大きいクラスタと，サイズが K より小さいクラスタが離れている場合がある．このとき，遠く離れた個体であるにもかかわらず，小さい方のクラスタへと所属を変更されてしまい，不自然なクラスタとなってしまう.

アルゴリズムを Algorithm 9, 10 に示す.

3.2.3 Clustering-based K-anonymity

Clustering-based K-anonymity [18] (CBK) はデータセットの 2 分割を繰り返すことにより，K-member Clustering を行う．$2K$ 以上のクラスタの 2 分割を繰り返すことにより，最終的に K 以上 $2K - 1$ 以下のサイズのクラスタを得ることができる．また，分割に基づいているため，GKC や OKA のように近くにクラスタがあるにも関わらず，クラスタ中心から遠く離れた個体が所属してしまうようなイレギュラーな分類になることはない.

アルゴリズムは，最初にデータセットの全ての個体を 1 つのクラスタとし，$2K$ 以上のクラスタが存在する場合にそのクラスタ C を C_1, C_2 に 2 分割することを繰り返す．分割では，ク

Algorithm 9 OKA：クラスタリングフェーズ

Step 1. データセットの個体を全ての属性についてソートし，X' とする．

Step 2. $c = \lfloor \frac{n}{k} \rfloor$ とする．

Step 3. ランダムに c 個の個体 $r_1, \ldots, r_c$ を選択する．

Step 4. 個体 r_i をクラスタ C_i に所属させ，X' から除く．

Step 5a. X' の最初の個体を r とし，X' から除く．

Step 5b. r を最も近いクラスタに所属させ，クラスタ中心を更新する．

Step 6. Step 5a-b. を全ての個体に対して行う．

Algorithm 10 OKA：調整フェーズ

Step 1. サイズが K より大きいすべてのクラスタ C について，Step 1a,b. を行う．

Step 1a. C 内の個体をクラスタ中心からの距離でソートする

Step 1b. 最も遠い $|C| - K$ 個の個体を所属から外す．

Step 2. 所属から外したすべての個体に対し，Step 2a,b. を行う．

Step 2a. 所属から外した個体をランダムに選択する．

Step 2b. サイズが K より小さいクラスタがある場合は，最も近いものに所属させる．
そうでない場合は，単純に最も近いクラスタに所属させる．

ラスタ中心を更新しながら，r 回分割を試行し，最もクラスタ内距離の小さいクラスタ C_1, C_2 を新たなクラスタとする．クラスタ中心は C の最初の分割では，ランダムに 2 つの個体 v_1, v_2 を選択する．最初の分割でない場合は，前回までのクラスタで最もクラスタ内距離の小さい C_1, C_2 のクラスタ中心を v_1, v_2 とする．そして，C のそれぞれの個体について，クラスタ中心 v_1, v_2 との距離を求め，小さい方のクラスタに所属させる．このアルゴリズムを Algorithm 11 に示す．

K-匿名化における情報損失の最小化という観点からは，クラスタサイズは小さくするべきである．これは，クラスタサイズが大きい場合は，クラスタ中心から遠い位置に個体があるため，クラスタ内で各属性で取りうる値が広がってしまい，データ加工の際に失う情報量が大きくなるためである．したがって，データセットを分割する際は，クラスタ数を最大化し，クラスタの大きさを K に近づけたほうが良い．しかしながら，CBK は 2 分割に基づいたクラスタリングであるため，クラスタサイズは K 以上 $2K-1$ 以下となる．ゆえに，比較的サイズの大きいクラスタも生成され，それに伴いクラスタ数についても必ずしも最大とはならない．

Algorithm 11 Clustering-based K-anonymity

Step 1. データセット全体を 1 つのクラスタとする．分割の反復回数 r を設定する．

Step 2. $2K$ 以上のクラスタがある場合，これを 1 つ選び C として，Step3. へ．そうでない場合は，終了する．

Step 3. Step 3a-c. を r 回繰り返し，その中で最もクラスタ内距離の小さい C_1, C_2 を新たなクラスタとする．その後，Step 2. に戻る．

Step 3a. 1 回目の分割の場合は，ランダムに選んだ 2 つの個体をクラスタ中心 v_1, v_2 とする．そうでない場合は，前回の C_1, C_2 のクラスタ中心を v_1, v_2 とする．

Step 3b. C のすべての個体について，C_1, C_2 のうちクラスタ中心が近い方のクラスタに分類する．

Step 3c. C_1, C_2 がともに少なくとも K の個体を持つように調整を行う．

3.2.4 Two-division Clustering for K-anonymity of Cluster Maximization

Two-division Clustering for K-anonymity of Cluster Maximization [4](2DCKM) は CBK に対して改良を行った手法であり，クラスタ数が最大となる分割を行う．

最初にデータセットを 1 つのクラスタとみなした後，サイズが $2K$ よりも大きいクラスタの 2 分割とクラスタサイズの調整を繰り返すアルゴリズムとなっている．調整は，小さい方のクラスタのサイズが βK となるように個体の所属を移動させる事によって行う．その際に，クラスタ C_1 から移動する場合と C_2 から移動する場合の両方の情報損失量を計算し，その値が小さい方を実行する．ここで，個体 x の所属を移動した場合の情報損失量の変化 Δ_x は，移動後と移動前のクラスタ中心 v_{after}, v_{before} を用いて，次のように表す．

$$\Delta_x = d(x, v_{after}) - d(x, v_{before}).$$

以上のアルゴリズムについて，Algorithm 12 に主となる部分を，Algorithm 13 にクラスタサイズ調整の部分を示す．

3.3 サイズ均等クラスタリング

K-member Clustering では，サイズの制約として K 以上の個体数を持つようなクラスタリングを行っていた．ここに，さらにサイズの制約を付け加える事によってサイズを均等化することを考える．K-member Clustering において，データセットの大きさ n に対して K の値を適切に指定することにより，ほぼ均等なサイズのクラスタ分割を得ることができる．しかしながら，ある程度の偏りは発生してしまう．そこで，サイズを均等にするよう，サイズ均

Algorithm 12 2DCKM

Step 1. K を設定し，$R = \{T\}$ とする．

Step 2. R 内で $2K$ 以上のクラスタを 1 つ選び C とする．

Step 3. C をクラスタリングにより 2 分割し，C_1, C_2 とする．

Step 4a. $|C_1|$or$|C_2| < K$ の場合，もし $|C_1| < K$ なら，C_1 に最も近い個体 $x_k \in C_2$ を r とし，$C_1 = C_1 \cup \{r\}, C_2 = C_2 \setminus \{r\}$ とする．もし $|C_2| < K$ なら C_1 と C_2 を逆にした操作を同様に行う．

Step 4b. $|C_1|$ and $|C_2| \geq K$ の場合，$|C_1|, |C_2|$ の小さいほうが βK(β は自然数) でなければ Algorithm 13 へ．そうでなければ，Step 5. へ．

Step 5. $R = R \cup \{C_1, C_2\} \setminus \{C\}$ とし，R 内のクラスタのサイズが $2K$ 未満ならば終了．そうでなければ，Step 2. へ．

Algorithm 13 Adjustment Algorithm (2DCKM)

$|C_1| < |C_2|$ の場合の操作を以下に示す．$|C_1| > |C_2|$ の場合は C_1, C_2 を逆にした処理を行う．

Step 1. $|C_1| \mod K = 0$ となるまで，C_1 に対して最も非類似度の小さい個体 $x_k \in C_2$ を r として $C_1 \cup \{r\}, C_2 \setminus \{r\}$ の操作を繰り返した場合の $\sum_r \Delta^A$ を求める．

Step 2. $|C_2| \mod K = 0$ となるまで，C_2 に対して最も非類似度の小さい個体 $x_k \in C_1$ を r として $C_1 \setminus \{r\}, C_2 \cup \{r\}$ の操作を繰り返した場合の $\sum_r \Delta^B$ を求める．

Step 3. $\sum_r \Delta^A, \sum_r \Delta^B$ のうち，値の小さい方の処理を選び，実際に行う．ただし，$|C_1|$ or $|C_2| = 0$ となるケースは選ばない．

等クラスタリングが提案されている．ここで，n によってはサイズを完全に均等にすることはできないため，実際にはサイズを K 以上かつ $K+1$ 以下とする．

また，K-匿名化以外への応用も想定しているため，距離尺度は K-member Clustering の尺度を用いず，一般的に用いられているユークリッド 2 乗距離をを用いる．

3.3.1 Extended Two-division Clustering for K-anonymity of Cluster Maximization

Extended Two-division Clustering for K-anonymity of Cluster Maximization [4] (E2DCKM) は 2DCKM を基に，クラスタのサイズを均等にする手法である．

一つのクラスタ C が最終的にクラスタサイズが K または $K+1$ のクラスタに分割できる

ための条件は次の式を満たすことである [4].

$$|C| \geq (K+1) \cdot (|C| \bmod K) \tag{3.1}$$

この条件式を満たすクラスタに対して，分割後の調整の度に同様の式を満たすように所属の移動を行うことで，均等なサイズのクラスタを得ることができる．

データセットのサイズ n と指定したクラスタのサイズ K の値によっては式 3.1 を満たさないため，指定したサイズに均等にすることができない．この場合は，サイズ均等クラスタリングを行えないため，2DCKM を行う．

式 3.1 を満たすデータセットに対して，この手法を用いるが基本的な流れは 2DCKM と同様である．2DCKM と異なる点は，調整の際は，分割後のクラスタ C_1, C_2 の両方が式 3.1 を満たすように行うという点である．この式を満たしていればその後の C_1 および C_2 についても 3.1 を満たすことができるクラスタへと分割することができ，最終的に K または $K+1$ のサイズへと分割できる．

このアルゴリズムについて Algorithm 14 に主な部分を，Algorithm 15 に調整の部分を示す．

Algorithm 14 E2DCKM

Step 1. $|T| \geq (K+1) \cdot (|T| \mod K)$ を満たすなら Step 2. へ．そうでなければ 2DCKM を行う．

Step 2. K を設定し，$R = \{T\}$ とする．

Step 3. R 内で $2K$ 以上のクラスタを 1 つ選び C とする．

Step 4. C をクラスタリングにより 2 分割し，C_1, C_2 とする．

Step 5. $|C_i| \geq (K+1) \cdot (|C_i| \mod K), i = 1, 2$ ならば Step6 へ．そうでなければ，Algorithm 15 へ．

Step 6. $R = R \cup \{C_1, C_2\} \setminus \{C\}$ とし，R 内のクラスタのサイズが $2K$ 未満ならば終了．そうでなければ，Step 3. へ．

3.4 最適化に基づくサイズ均等クラスタリング

最適化に基づくサイズ均等クラスタリング (Even-sized Clustering Based on Optimization; ECBO) は HCM をもとに，クラスタサイズを均等にする制約式を追加してクラスタリングを行う．また，目的関数を設定し，最適化手法を用いてクラスタリングを行うため，より情報損失の少ない分類をすることができる．クラスタリングは式 3.2-3.5 の最適化問題を解くこと

Algorithm 15 Adjustment Algorithm (2DCKM)

Step 1. $|C| \geq (K+1) \cdot (|C| \mod K), i = 1, 2$ となるまで，C_1 に対して最も非類似度の小さい個体 $x_k \in C_2$ を r として $C_1 \cup \{r\}, C_2 \setminus \{r\}$ の操作を繰り返した場合の $\sum_r \Delta^A$ を求める．

Step 2. $|C| \geq (K+1) \cdot (|C| \mod K), i = 1, 2$ となるまで，C_2 に対して最も非類似度の小さい個体 $x_k \in C_1$ を r として $C_1 \setminus \{r\}, C_2 \cup \{r\}$ の操作を繰り返した場合の $\sum_r \Delta^B$ を求める．

Step 3. $\sum_r \Delta^A, \sum_r \Delta^B$ のうち，値の小さい方の処理を選び，実際に行う．ただし，$|C_1|$ or $|C_2| = 0$ となるケースは選ばない．

によって行う．

$$\text{minimize} \quad J_{\text{ECBO}} = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ki} \|x_k - v_i\|^2 \tag{3.2}$$

$$\text{s.t.} \quad \sum_{i=1}^{c} u_{ki} = 1 \quad (k = 1, \cdots, n) \tag{3.3}$$

$$\sum_{k=1}^{n} u_{ki} \geq K \quad (i = 1, \cdots, c) \tag{3.4}$$

$$\sum_{k=1}^{n} u_{ki} \leq K + 1 \quad (i = 1, \cdots, c). \tag{3.5}$$

これらの目的関数および制約式は u_{ki} に関して線形であるため，線形計画法により U の最適化を行うことができる．そこで，シンプレックス法を用いて U の最適化を行う．また，V の最適化については，V についての制約式は加えられていないため，HCM と同様にクラスタの重心を求めれば良い．このアルゴリズムを Algorithm 16 に示す．

Algorithm 16 最適化に基づくサイズ均等クラスタリング

Step 0. 定数 K, c を設定する．

Step 1. ランダムに c 個のクラスタ中心を選択する．

Step 2. U の更新： シンプレックス法により U を求める．

Step 3. V の更新： クラスタの重心を求め，新たな V とする．

Step 4. 収束したら終了．そうでなければ Step 2. にもどる．

また，最初にクラスタサイズ K またはクラスタ数 c を指定するが，指定しない方の変数は

指定した方の変数を用いて決定する．

c を指定する場合

n 個の個体をサイズが K または $K+1$ の c 個のクラスタに分割する際の n, c, k に関する条件は $Kc \leq n < (K+1)c$ である．また，床関数 $\lfloor a \rfloor$ は定義より $\lfloor a \rfloor \leq a < \lfloor a \rfloor + 1$ という性質をもつ．ここで，$Kc \leq n < (K+1)c$ において，全ての辺を c で割ると，$K \leq \frac{n}{c} < K+1$ となる．したがって，K は $\frac{n}{c}$ の床関数にほかならない．ゆえに，この場合は，$K = \lfloor \frac{n}{c} \rfloor$ とすればよい．

K を指定する場合

K の値によっては，個体を K または $K+1$ のサイズのクラスタに分割することができない場合がある．このとき，E2DCKM の場合と同様に，K は式 3.1 を満たすようにすれば良い．そこで，式 3.1 を満たすまで K を 1 ずつ増加させることにより適切な K を選択する．その後，この K を用いて $c = \lfloor \frac{n}{K} \rfloor$ と設定する．

第4章　提案手法

4.1　ECBO++

ECBO では初期値をランダムに与えてクラスタリングを行っていた．しかしながら，従来の HCM や FCM と同様に，与えられる初期値によっては，計算量・クラスタリング結果ともに悪い結果をもたらす場合がある．k-means++は FCM の初期値選択を改善した手法であるが，これと同様の初期値選択方法を ECBO にも用いることを考える．偏りが少なくなるようクラスタ中心を確率的に選択することにより，計算量やクラスタリング結果の改善につなげる．

ここで，目的関数および制約式は ECBO と同じものを用い，距離尺度もユークリッド 2 乗距離を用いる．すなわち，

$$\begin{aligned}
\text{minimize} \quad & J_{\text{ECBO++}} = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ki}\|x_k - v_i\|^2 \\
\text{s.t.} \quad & \sum_{i=1}^{c} u_{ki} = 1 \quad (k = 1, \cdots, n) \\
& \sum_{k=1}^{n} u_{ki} \geq K \quad (i = 1, \cdots, c) \\
& \sum_{k=1}^{n} u_{ki} \leq K + 1 \quad (i = 1, \cdots, c).
\end{aligned}$$

アルゴリズムは，まず k-means++と同様に Algorithm 3 の Step 1a-c. により，初期値を決定する．その後，この初期値を用いて通常の ECBO を行う．V の最適化の計算オーダーは $O(ncp)$ である．このアルゴリズムを Algorithm 17 に示す.

4.2　L_1 ECBO

ECBO に L_1 距離を導入することによって，L_1FCM のように計算量や外れ値に対するロバスト性が改善されることが期待される．このアルゴリズムを Algorithm 18 に示す.

Algorithm 17 ECBO++

Step 0. 定数 K, c を設定する．

Step 1. 初期クラスタ中心 V を Algorithm 3 の Step 1a-c. により与える．

Step 2. 帰属度 U を更新する： シンプレックス法により最適解を求める．

Step 3. クラスタ中心 V を更新する： クラスタの重心を V とする．

Step 4. 新たな重心 V が前回の重心から変化しなければ終了．そうでなければ Step 2 へ

ここで，制約式は ECBO と同じ物を用い，距離尺度は L_1 距離を用いる．すなわち，

$$\begin{aligned} \text{minimize} \quad & J_{L_1\text{ECBO}} = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ki}\|x_k - v_i\|_1 \\ \text{s.t.} \quad & \sum_{i=1}^{c} u_{ki} = 1 \quad (k = 1, \cdots, n) \\ & \sum_{k=1}^{n} u_{ki} \geq K \quad (i = 1, \cdots, c) \\ & \sum_{k=1}^{n} u_{ki} \leq K + 1 \quad (i = 1, \cdots, c). \end{aligned}$$

また，U の最適化にはシンプレックス法を用いる．V の最適化の際は，距離の尺度を L_1 ノルムに変更したため，L_1FCM と同様に準目的関数 J_{ij} の最適化を行って求める．この V の最適化の計算オーダーは $O(ncp)$ である．

Algorithm 18 L_1 ECBO

Step 0. 定数 K, c を設定する．

Step 1. 初期クラスタ中心 V を，$x_k \in X$ からランダムに選択する．

Step 2. 帰属度 U を更新する： シンプレックス法により最適解を求める．

Step 3. クラスタ中心 V を更新する： J_{ij} の最小化を v_{ij} について解く．

Step 4. 新たな重心 V が前回の重心から変化しなければ終了．そうでなければ Step 3 へ

4.3 Medoid ECBO

k-medoids は個体同士の距離さえ与えればクラスタリングを行うことができ，外れ値に対するロバスト性にも利点がある．このようなクラスタリングを行うように，k-medoids と同様に V の更新を行うように ECBO のアルゴリズムを変更する．ここで，U の最適化については，シンプレックス法を用いる．また，V の最適化の計算オーダーは $O(n^2c)$ である．さらに，目的関数および制約式は基本的に ECBO と同じものを用いるが，距離尺度については一般的な k-medoids Clustering と同様に通常のユークリッド距離を用いる．すなわち，

$$\begin{aligned} \text{minimize} \quad & J_{\text{MECBO}} = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ki}\|x_k - v_i\| \\ \text{s.t.} \quad & \sum_{i=1}^{c} u_{ki} = 1 \quad (k = 1, \cdots, n) \\ & \sum_{k=1}^{n} u_{ki} \geq K \quad (i = 1, \cdots, c) \\ & \sum_{k=1}^{n} u_{ki} \leq K + 1 \quad (i = 1, \cdots, c). \end{aligned}$$

このアルゴリズムを Algorithm 19 に示す.

Algorithm 19 Medoid ECBO

Step 0. 定数 K, c を設定する.

Step 1. 初期クラスタ中心 V を，$x_k \in X$ からランダムに選択する.

Step 2. 帰属度 U を更新する： シンプレックス法により最適解を求める.

Step 3. クラスタ中心 V を更新する： 各クラスタ内で，距離の総和を最小にする個体 x を求め，クラスタ中心とする.

Step 4. 新たな重心 V が前回の重心から変化しなければ終了．そうでなければ Step 2 へ

4.4 Kernel ECBO

ECBO は HCM を基にした手法であるため，線形な分割しか行うことができない．そこでカーネルクラスタリングの手法を導入し，非線形な分割を行うことにより結果を改善することを考える．目的関数および制約式は ECBO と同じだが，距離 d_{ki} には式 2.6 を用いる．すな

わち,

$$
\begin{aligned}
\text{minimize} \quad & J_{\text{KECBO}} = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ki}\|\phi(x_k) - v_i^{\phi}\|^2 \\
& = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ki}\left\{\kappa(x_k,x_k) - \frac{2}{|C_i|}\sum_{x_l\in C_i}\kappa(x_k,x_l) + \frac{1}{|C_i|^2}\sum_{x_l\in C_i}\sum_{x_m\in C_i}\kappa(x_l,x_m)\right\} \\
\text{s.t.} \quad & \sum_{i=1}^{c} u_{ki} = 1 \quad (k = 1,\cdots,n) \\
& \sum_{k=1}^{n} u_{ki} \geq K \quad (i = 1,\cdots,c) \\
& \sum_{k=1}^{n} u_{ki} \leq K+1 \quad (i = 1,\cdots,c).
\end{aligned}
$$

距離行列 δ の更新の計算オーダーは $O(n^3)$ である.

以上のアルゴリズムを Algorithm 20 に示す.

Algorithm 20 Kernel ECBO

Step 0. 定数 K, c を設定する.

Step 1. 初期クラスタ中心 V を，$x_k \in X$ からランダムに選択する.

Step 2. $||\phi(x_k) - v_i^{\phi}||^2$ をすべての k, i で計算し，距離行列 δ とする.

Step 3. 帰属度 U を更新する： シンプレックス法により最適解を求める.

Step 4. 距離行列 δ を更新する.

Step 5. U が前回から変化しなければ終了．そうでなければ Step 3 へ

第5章　数値例

5.1　情報損失関数

ECBO の改善を行った各手法では，目的関数が異なるため，目的関数をもとにして単純に比較を行うことができない．そこで，情報損失関数 IL (Information Loss) [16] を用いて比較を行う．これは K-匿名化を行う際の情報の損失を表す尺度であり，クラスタリングの精度を表す 1 つの目安となる．この値が小さいほど情報損失が小さく，良い分類結果であると言える．数値データのみを扱う場合は，以下の式で表される．ここで，データ集合 $X = \{x_1, \cdots, x_n\}$ とし，$\hat{N}_j(S)$ は集合 $S \subseteq X$ の要素の j 成分の最大値，$\check{N}_j(S)$ は S の要素の j 成分の最小値を表す.

$$\mathrm{IL} = \sum_{i=1}^{c} |C_i| \cdot D(C_i),$$

$$D(C_i) = \sum_{j=1}^{p} \frac{\hat{N}_j(C_i) - \check{N}_j(C_i)}{\hat{N}_j(X) - \check{N}_j(X)}$$

すべての個体がただひとつのクラスタに属する場合に IL は最大値をとり，各個体が一つずつ別々のクラスタに属する場合には最小値 $\mathrm{IL} = 0$ となる．

5.2　人工データ

4 つのデータセットを用いて ECBO と各提案手法でクラスタリングを行った結果の比較により評価を行う．KECBO のカーネル関数にはガウシアンカーネルを用いている．

5.2.1　ノイズ入りデータ

個体数 100 個の正規分布 3 つに加え，ノイズを 15 個ずつ 2 か所に配置した，合計 330 個体を持つデータセットについてクラスタリングを行う．このデータを図 5.1 に示す．このデータに対して，各手法について $K = 110$ で 100 回クラスタリングを行った．

表 5.1 に各手法での最良の IL 値と平均実行時間を示す．また，図 5.2-5.5 に各手法での最良のクラスタリング結果を示す．ここで，凡例の括弧内の数字はクラスタの持つ個体数を表し，KECBO 以外の結果における “center”はクラスタ中心を表す.

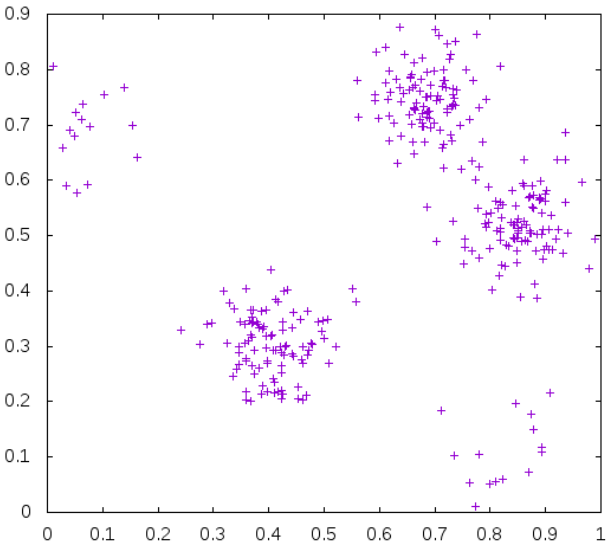

図 5.1: ノイズ入りデータ

IL 値に関しては，ECBO と ECBO++が同じ結果となり，減少がみられたのは Medoid ECBO および KECBO であった．実行時間に関しては，ECBO++, L_1ECBO, Medoid ECBO で減少しており，最も減少しているのは Medoid ECBO であった．

計算量について，L_1ECBO は ECBO と同じオーダー，Medoid ECBO は ECBO よりも大きなオーダーとなっているにもかかわらず，平均実行時間が小さい．これは目的関数の収束が早く，最適化の反復回数が小さく抑えられたからだと考えられる．また，ECBO++も ECBO と同じオーダーだが，こちらも平均実行時間が小さくなっている．これは初期値選択が改善されたことにより，最初から比較的小さな目的関数であったために早い収束になったと考えられる．

ECBO と L_1ECBO，Medoid ECBO のクラスタリング結果について，図 5.2, 5.4, 5.5 でのクラスタ中心に注目すると，ECBO ではクラスタ中心はノイズに引き寄せられ，クラスタのうちの主要な個体の集合の重心からずれてしまっている．対して，L_1ECBO および Medoid ECBO ではクラスタ中心はノイズにはあまり引き寄せられず，個体の主要な集合の中心に位置している．このことから，このデータセットについて，これらの 2 手法は外れ値に対するロバスト性が高い傾向がみられる．

表 5.1: ノイズ入りデータにおける各手法での IL 値および実行時間

手法	ECBO	ECBO++	L_1ECBO	Medoid ECBO	KECBO ($\sigma = 0.3$)
IL (最小)	397.48	397.48	413.37	382.38	378.83
time [ms] (平均)	76.11	68.44	64.82	63.55	641.74

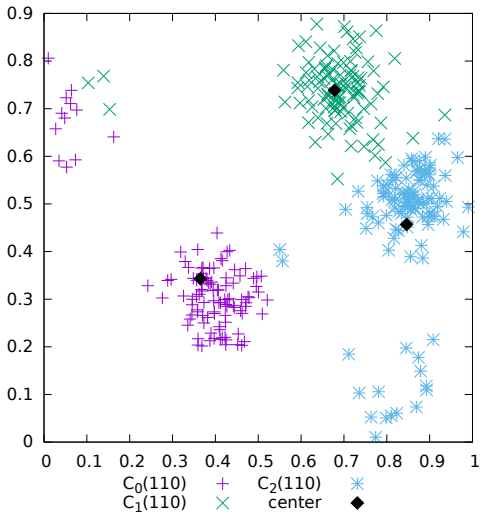

図 5.2: ノイズ入りデータに対する ECBO および ECBO++の結果 (IL= 397.748)

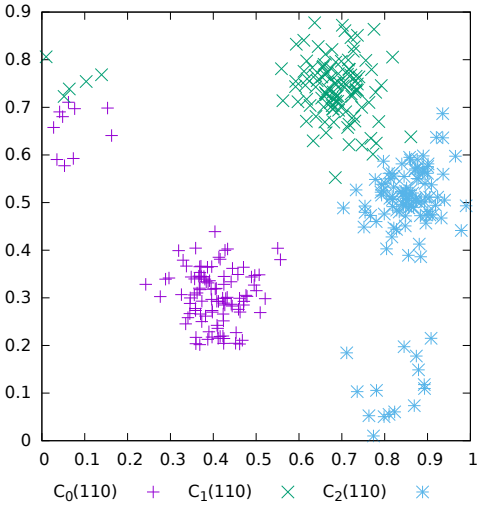

図 5.3: ノイズ入りデータに対する KECBO の結果 (IL= 378.66)

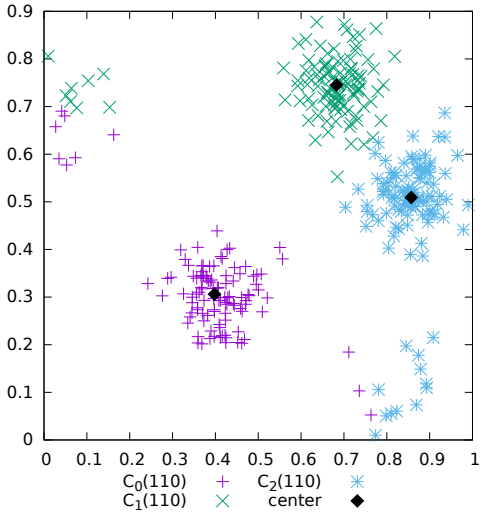

図 5.4: ノイズ入りデータに対する L_1ECBO の結果 (IL= 413.37)

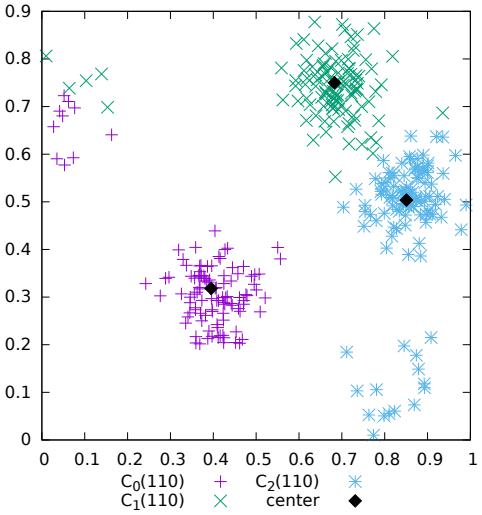

図 5.5: ノイズ入りデータに対する Medoid ECBO の結果 (IL= 382.37)

5.2.2　二重円データ

小さな円の内側に 50 個体，外側の円上に 100 個体の合計 150 の個体を持つデータセットについてクラスタリングを行う．このデータを図 5.6 に示す．このデータに対し，各手法について $c = 2, 3, 5$ で 100 回クラスタリングを行った．

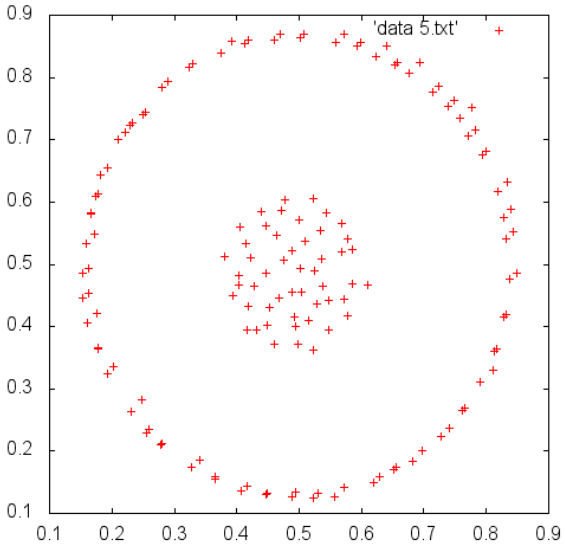

図 5.6: 二重円データ

表 5.2 に各手法で得られた最良の IL 値を示す．ECBO と ECBO++の IL 値はほとんど差は見られない．また，L_1ECBO は $c = 2, 5$ において最も良い値を，$c = 3$ において 2 番目に良い値をとっている．KECBO は $c = 3$ において最も良い値をとっているが，それ以外では最も悪い値をとっている．

平均実行時間を表 5.3 に示す．実行時間が最も短いのは Medoid ECBO，最も長いのは KECBO となった．Medoid ECBO は計算量オーダーが ECBO よりも大きいが，収束が早いため実行時間が小さくなったと考えられる．

$c = 3$ とし，データセットを 50 個ずつ，3 つに分割した時の ECBO と KECBO のクラスタリング結果を図 5.7, 5.8 に示す．ECBO では分割は線形分離の形となり，点対称に近いクラスタ分類となっているのに対し，KECBO では非線形分離となり，小さい円のクラスタと，大きい円を 2 分割したクラスタに分類することができている．また，IL の値も KECBO の方が小さい値となった．

次に $c = 5$ とし，データセットを 30 個ずつ，5 つに分割する場合を考える．この時の ECBO の結果を図 5.9，KECBO の結果を図 5.10，IL 値が最小であった L_1ECBO の結果を図 5.11 に示す．ECBO では小さい円の中心部で 1 つのクラスタを作っており，余った個体が外側の 4 つクラスタに所属している．また，KECBO では大きい円上の 3 つのクラスタが作られ，その余った個体と小さい円の個体でクラスタが作られている．離れた位置に複数の集合のあるク

ラスタは少ないクラスタリングを実現しているが，IL 値は増大してしまっている．最も IL 値が小さくなったのは L_1ECBO で，大きい円上に 2 つのクラスタ，小さい円上に 1 つのクラスタが作られ，残りの個体で 2 つのクラスタを作っている．クラスタ中心が縦や横に並んで配置されたため，縦長や横長の形のクラスタが生成されやすくなっている．そのため，斜めに長いクラスタができず，IL 値が小さくなったと考えられる．

以上の様に，KECBO によるクラスタでは，非線形な分離や，複数の個体の集合からなるクラスタを減らすことができた．しかしながらデータセットの分布によっては，必ずしも IL 値が最小となる分類はできない．

c	2	3	5
ECBO	224.6	185.0	130.7
ECBO++	224.6	185.0	130.4
L_1ECBO	224.5	181.5	127.8
Medoid ECBO	225.4	185.2	131.3
KECBO	237.3	181.2	134.3

表 5.2: 二重円データにおける各手法での最良の IL 値

c	2	3	5
ECBO	12.34	19.94	33.68
ECBO++	13.53	19.73	30.52
L_1ECBO	13.31	19.06	38.51
Medoid ECBO	10.64	16.01	29.58
KECBO	44.42	72.59	136.94

表 5.3: 二重円データにおける各手法での平均実行時間 [ms]

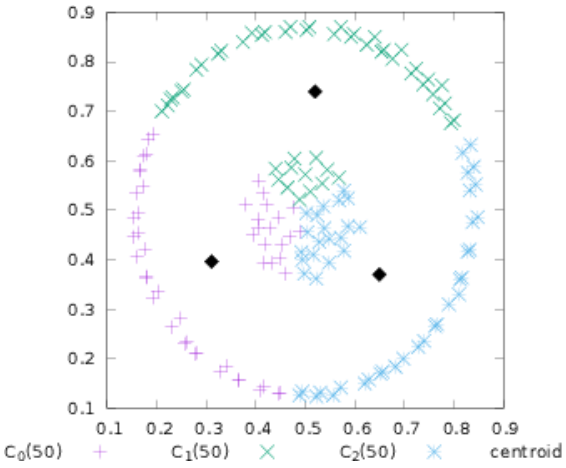

図 5.7: 二重円データに対する ECBO の結果 (IL= 185.00)

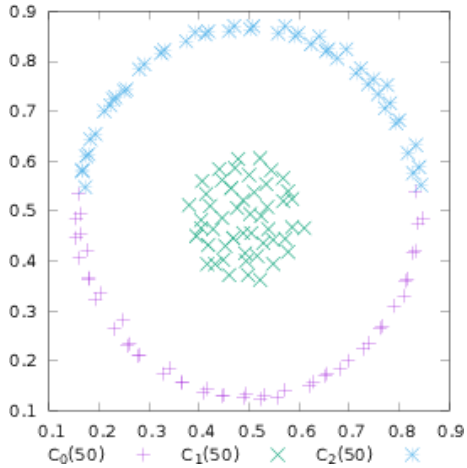

図 5.8: 二重円データに対する KECBO の結果 (IL= 181.23)

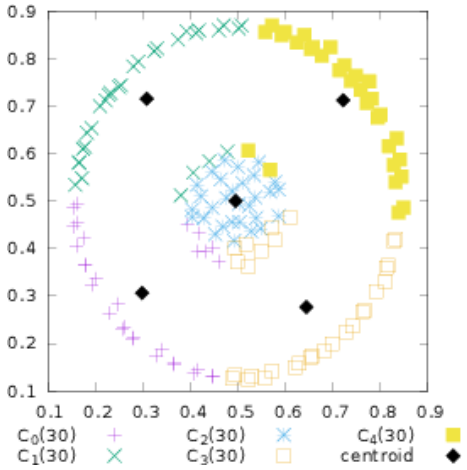

図 5.9: 二重円データに対する ECBO の結果 (IL= 130.65)

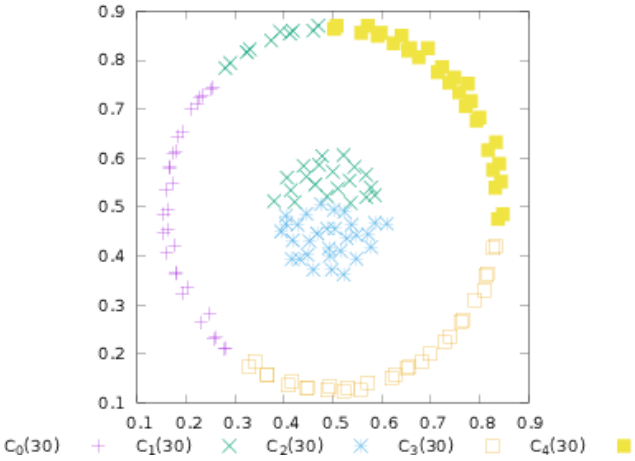

図 5.10: 二重円データに対する KECBO の結果 (IL= 134.32)

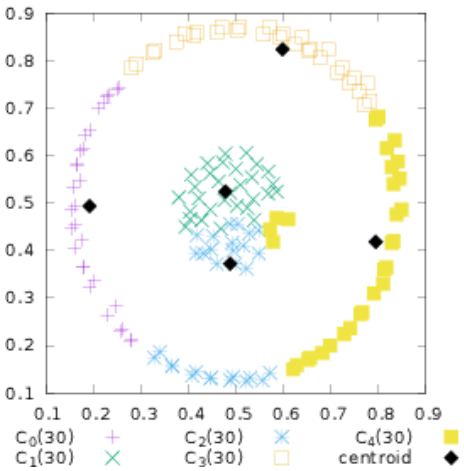

図 5.11: 二重円データに対する L_1ECBO の結果 (IL= 127.77)

5.3 実データ

5.3.1 Iris データ

Iris データ [19] はアヤメの花に関するもので，がく片の幅と長さ，花弁の幅と長さの 4 次元とアヤメの種類のラベルが付いている．個体数は $n = 150$ で，3 種類のあやめのデータが 50 ずつ含まれている．これを各手法について，$K = 50$ で 100 回クラスタリング行い，修正 Rand Index [20] により正解のラベルと比較し，正しい分類になっているかを検証する．比較のため，ECBO 以外にも HCM および FCM によるクラスタリングの結果も求めた．ここで，HCM および FCM は情報損失を考慮していない手法であるため，IL 値は求めなかった．

表 5.4 に各手法での Rand Index の値，IL 値，実行時間を示す.

RI については，ECBO，ECBO++，Medoid ECBO が同じ値となっており，L_1ECBO と KECBO は ECBO よりも改善されている．また，HCM，FCM と比較すると非常に高い値となっている．KECBO は $\sigma = 0.6$ の場合は RI 値が非常に高い値となった．しかしながら，RI 値の分散は比較的大きく，RI 値の最小値も小さな値となっているため，出力される結果は不安定であるといえる．$\sigma = 2.0$ の場合は比較的 RI 値が低いが，HCM や FCM よりも高い値となった．また，RI 値の分散も 0 となり，出力される結果は非常に安定している．

計算時間は ECBO++が最も短く，次いで Medoid ECBO についても短い実行時間となっている．

5.3.2 地図データ

google map [21] の住宅地の地図から住宅を抽出したデータ (図 5.12，$n = 502$) について各手法で 100 回ずつ行った．ここでは，c を $2, 5, 10$ と設定し，KHCM のパラメータは $\sigma = 100$ とした.

各手法について、地図データ (n=502) に対して 100 回クラスタリングを行った際の最良の IL 値と平均実行時間を表 5.5, 5.6 に示す。

また，ECBO と ECBO++について IL 値の平均と分散を表 5.7 に示す.

ECBO と ECBO++の IL 値は最小値，平均値が共にほぼ等しくなっている．しかしながら，IL 値の分散値は僅かに小さくなっており，安定した結果を出すのは ECBO++だといえる．また，実行時間についても，ECBO++のほうが短いという結果となっている.

L_1ECBO は IL 値，実行時間共に ECBO よりも悪化している．Medoid ECBO については，実行時間が短くなっている一方，IL 値は大きくなっている結果も出ている.

KECBO は全ての手法の中で $c = 2, 10$ で最小となっているが，$c = 5$ では最大の値となっている．また，実行時間は非常に長くなっている.

表 5.4: Iris データにおける各手法での値

手法	ECBO	ECBO++	L_1ECBO	Medoid ECBO
RI (最大)	0.893	0.893	0.909	0.893
RI (最小)	0.893	0.893	0.893	0.893
RI (平均)	0.893	0.893	0.903	0.893
RI (分散)	0.000	0.000	5.87×10^{-5}	0.000
IL (最小)	272.4	272.4	273.2	272.4
time [ms] (平均)	15.00	12.91	17.20	14.76

手法	KECBO ($\sigma = 0.6$)	KECBO ($\sigma = 2.0$)	HCM (参考)	FCM ($m = 1.5$, 参考)
RI (最大)	0.961	0.785	0.730	0.716
RI (最小)	0.199	0.785	0.420	0.422
RI (平均)	0.851	0.785	0.672	0.645
RI (分散)	0.010	0.000	0.0098	0.016
IL (最小)	273.7	272.4	-	-
time [ms] (平均)	84.00	55.42	0.24	6.93

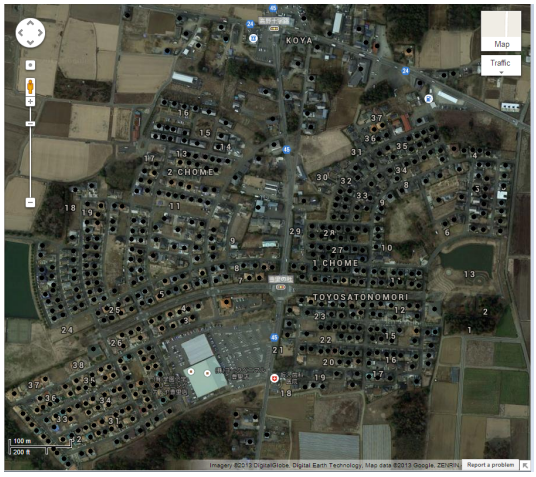

図 5.12: 茨城県つくば市豊里の杜の衛星写真 (google map より)

c	2	5	10
ECBO	759.24	404.90	271.10
ECBO++	759.24	404.90	271.10
L_1ECBO	763.66	411.17	277.64
Medoid ECBO	779.53	407.86	268.26
KECBO	726.97	419.14	267.34

表 5.5: 地図データにおける各手法での最良の IL 値

c	2	5	10
ECBO	130.88	640.83	2420.36
ECBO++	108.55	591.51	2113.53
L_1ECBO	143.64	697.32	2471.65
Medoid ECBO	125.56	593.50	1909.39
KECBO	2950.65	10639.92	27792.31

表 5.6: 地図データにおける各手法の平均実行時間 [ms]

表 5.7: ECBO,ECBO++の IL 値の平均と分散

c	2	5	10
ECBO (平均値)	759.24	407.20	276.32
ECBO++ (平均値)	759.24	407.25	276.31
ECBO (分散値)	0.0	2.02	17.55
ECBO++ (分散値)	0.0	1.86	16.85

第6章　結論

本研究ではECBOに対し，ノイズに対するロバスト性や，分割が線形分離となることなどを改善するため，k-means++, L_1 Fuzzy c-means, k-medoids, Kernel Hard c-means の4手法の考え方やアルゴリズムを利用した手法をそれぞれ提案し，数値例を用いて考察を行った．

ECBOに対してk-means++の初期値選択手法を導入し，適切な初期値を選択するECBO++については，分類結果のIL値の最小値はECBOとほぼ同じとなった．しかし，IL値の分散値についてはECBOよりも小さくなり，比較的安定して良い結果を得ることができることが確認された．また，実行時間についてもECBOよりも短くすることができた．

ノイズに対するロバスト性をECBOに持たせるためL_1距離を用いたL_1ECBOには，L_1 Fuzzy c-meansのクラスタ中心の求め方を利用した．ノイズ入りのデータに対してクラスタリングを行った結果，クラスタ中心はクラスタのうち主要な個体の集合の中心に位置しており，ノイズに対するロバスト性の傾向が確認された．

また，ノイズに対するロバスト性が高い分類をすることや，グラフデータのような個体の距離のみによって表されるデータに対してクラスタリングを行える利点があるため，k-medoidsのクラスタ中心の求め方をECBOに導入したMedoid ECBOを提案した．ノイズ入りのデータに対してクラスタリングを行い，クラスタ中心が主要な個体の集合の中心に位置しており，ノイズに対するロバスト性の傾向を確認できた．更に，今回検証した多くのデータセットにおいて，ECBOよりも実行時間が改善されている．今回はユークリッド空間上の個体データのみを扱ったが，グラフデータへの応用も可能である．

更に，線形分離ではうまく分割できないデータセットを分割するため，ECBOに対してカーネルを導入したKECBOを提案した．二重円データを用いた検証により，内部の円と外部の円へと分割することができており，IL値も最も良い結果を得ることができた．また，多くのデータセットに対して小さなIL値となる結果を出力している一方，データセットやc, Kの値によってはIL値が比較的大きな値を取ることも確認された．

本研究ではECBOに対し，4つの手法の考え方を導入したが，その他のクラスタリング手法からの考え方の導入が今後の研究として考えられる．例えば，ラフクラスタリングは，ハードクラスタリングにおいて個体のクラスタへの帰属を「属する」「属さない」の2値に分割していたのに対し,「属する」「属さない」「属すかどうか不明」の3値に分割するクラスタリングである．クラスタの境界を厳密に決めないことで，所属が識別不能である個体を外し，確実にクラスタに所属するクラスタを取り出すことができる．中でも目的関数の最適化に基づくラフクラスタリング [22] が提案されており，ECBOへの導入もできると考えられる．

また，本研究では数値データのみを対象とした手法を提案したが，ECBOはK-匿名化への

応用に役立つと考えられる．そこで，今後は個人データをはじめとする，カテゴリーデータを含むデータセットを対象とした手法への拡張も必要となる．

利用するデータセットやクラスタリングの目的によって適している手法は異なっており，その選択は難しい問題である．本研究の提案手法は必ずしも汎用的に用いることができる手法ではないが，個体数が均等な分割を得るための手法として強力な選択肢になると考える．

謝辞

本研究を進めるにあたり，システム情報工学研究科 遠藤靖典教授には工学システム学類 4 年次より 3 年間に渡りご指導いただき，心より感謝致します．研究や論文の指導はもとより，就職活動など，私的な活動においても大変参考となるご助言を賜りました．

また，システム情報工学研究科 宮本定明教授，イリチュ（佐藤）美佳教授には学会発表やソフトコンピューティング基礎グループ内での研究発表等，様々な機会に際してご指導いただきました．深く感謝致します．

また，リスク工学専攻の先生方，並びにソフトコンピューティング基礎グループのメンバーの皆様には数多くの貴重なご意見をいただきました．特に，木下尚彦氏には研究や論文の内容に関して様々なご助言を頂きました．ここに深く感謝致します．

最後に，大学生活を支え続けてくれました両親に心より感謝致します．

参考文献

[1] 宮本 定明. クラスター分析入門. 森北出版株式会社, 1999.

[2] J. MacQueen. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp.281-297, 1967.

[3] 宮岸 聖高，市橋 秀友，本多 克宏．K-L 情報量正則化 FCM クラスタリング法．日本ファジィ学会誌 Vol.13，No.4，pp.406-417, 2001.

[4] 緒方 悠人，遠藤 靖典. K-Member Clustering 問題に関する一考察. 第 29 回ファジィシステムシンポジウム (FSS2013), 2013.

[5] 平野 翼，緒方 悠人，木下 尚彦，遠藤 靖典. 最適化に基づくサイズ均等クラスタリングアルゴリズム. 第 40 回ファジィ・ワークショップ講演論文集, pp.23-26, 2013.

[6] D. Arthur, S. Vassilvitskii. k-means++: The Advantages of Careful Seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027-1035, 2007.

[7] K. Jajuga. L_1-norm based fuzzy clustering. Fuzzy Sets and Systems, Vol.39, pp.43-50, 1991.

[8] L. Bobrowski and J. C. Bezdek. c-means clustering with the l_1 and l_∞ norms. Systems, Man and Cybernetics, IEEE Transactions on, Vol.21, No.3, pp.545-554, 1991.

[9] S. Miyamoto, A. Yudi. An efficient algorithm for l_1 fuzzy c-means and its termination. Control and Cybernetics, 24, pp.421-436, 1995.

[10] H.S. Park, C.H. Jun. A simple and fast algorithm for K-medoids clustering. Expert Systems with Applications 36.2, pp.3336-3341, 2009.

[11] L. Kaufman, P.J. Rousseeuw. Finding groups in data: An introduction to cluster analysis. New York: Wiley, 1990.

[12] M. Girolami. Mercer Kernel-Based Clustering in Feature Space. IEEE Trans. on Neural Networks, vol.13, no.3, pp.780-784, 2002.

[13] K. LeFevre, D. DeWitt, R. Ramakrishnan. Incognito: Efficient Full-domain K-Anonymity. In ACM International Conference on Management of Data, 2005.

[14] B.C.M. Fung, K. Wang, P.S. Yu. Top-Down Specialization for Information and Privacy Preservation. In International Conference on Data Engineering, 2005.

[15] K. LeFevre, D. DeWitt, R. Ramakrishnan. Mondrian Multidimensional K-anonymity. In International Conference on Data Engineering, 2006.

[16] J.-W. Byun, A. Kamra, E.Bertino, N. Li. Efficient k-Anonymization Using Clustering Techniques. In DASFAA, pp.188-200, 2007.

[17] J.-L. Lin, M.-C. Wei. An Effcient Clustering Method for k-Anonymization. ACM New York, NY, USA, 2008.

[18] X. He, H.-H. Chen, Y. Chen, Y. Dong, P. Wang, Z. Huang. Clustering-Based K-anonymity. PAKDD, pp.405-417, 2012.

[19] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics 7 (2), pp.179-188, 1936.

[20] L. Hubert, P. Arabie. Comparing Partitions. Journal of Classification 2, pp.193-218, 1985.

[21] Google マップ, https://maps.google.co.jp/

[22] Y. Endo, N. Kinoshita. On Objective-Based Rough c-Means Clustering. Proc. of The 2012 IEEE International Conference on Granular Computing, 2012.