分割・併合機能を有するK-Means アルゴリズムによるクラスタリング

倉橋　和子
奈良女子大学　人間文化研究科
博士前期課程 情報科学専攻

２００７年1月３１日

目 次

図 目 次

表 目 次

1 はじめに

パターン認識とは，認識対象がいくつかのクラスに分類できるとき，観測されたパターンをそれらのクラスのうちのひとつに対応させる処理のことである。パターン認識における最も基本的な課題は，未知の認識対象を計測して得られた特徴ベクトルから，その対象がどのクラスに属するかを判定する識別方法を開発することである。パターン認識は，教師あり分類と教師なし分類の 2 種類に大別できる。

- 教師あり分類
  外的基準を用いて，既知の分類結果から未知の分類パターンの推定を行う。

- 教師なし分類
  外的基準なしに自動的に分類を行う手法である。分類すべき個体の間に定義された類似性や距離にもとづいてグループ分けを行う。

本論文では，教師なし分類の代表的な手法であるクラスタリングを扱う。クラスタリング [1][2][3] とは，データの集まりを外的基準なしに自動的にクラスターに分類を行う方法であり，クラスター分析とも呼ばれ様々な分野で応用されている。クラスタリングの技法は，最短距離法などの階層的クラスタリングと，K-Means 法 [5][8][9] などの非階層的クラスタリングに大きく分けられる。

- 階層的クラスタリング
  1 個の対象だけを含む N 個のクラスターがある初期状態から，クラスター間の距離 (非類似度) 関数にもとづき，最も距離の近い 2 つのクラスターを逐次的に併合する。そして，この併合をすべての対象が 1 つのクラスターに併合されるまで繰り返すことで階層構造を得る。

- 非階層的クラスタリング
  分割のための評価関数を定義し，その評価関数を最適にする分割を探索する。可能な分割の総数は個体数 N に対して指数的であるので，実際は準最適解を求める。

非階層的クラスタリングの代表的な手法として，K-Means 法やファジィ c-平均法 [4][7] などがある。K-Means 法は，セントロイドをクラスターの代表点とし，個体のクラスターへの割り当てと代表点の再計算を交互に繰り返すことで最適解の探索を行うものである。それとは異なるアプローチによる技法に学習ベクトル量子化 (Learning Vector Quantization:LVQ)[6][11] によるクラスタリングがあり，本論文ではこの 2 つのアルゴリズムを扱っている。K-Means は評価関数の最適化にもとづいているのに対して，LVQ は学習概念にもとづいている。

K-Means アルゴリズムや LVQ は，異なる統計分布を有するクラスから構成されるデータに対してクラスタリングを行った場合，必ずしも効果的な分類結果が得られるとは限らない。K-Means アルゴリズムによるクラスタリングを行った時の各決定領域における歪みの和は，各クラスが持つ歪みの和よりも小さくなるが実際は誤りを有する分類結果になる [16][17]。仮

に，それぞれのクラスターの中心が正しく求まったとしても，統計的に異なる分布をもつクラスデータをボロノイ分割すると分類誤りが生じ，K-Means アルゴリズムでは正しく分類することはできない。この問題を解決する方法として，本論文では，分割・併合機能を有する K-Means アルゴリズムによるクラスタリングを新たに提案する。

従来の K-Means アルゴリズムによるクラスタリングを行い，K 個の決定領域を得たとする。得られた各決定領域に対して再び K-Means アルゴリズムを適用し，それぞれの決定領域にクラスターが唯 1 つだけ存在しているか否かを判定する方法を提案する。

さらに、決定領域を部分領域に分割する尺度を導入する。得られた各決定領域において，クラスターが 1 つだけ存在していると判断された場合，そのクラスタリング過程を終了する。しかし，決定領域内に 2 つ以上のクラスターが存在すると判断された場合，分割の尺度にもとづいて決定領域を部分領域に分割する。部分領域のクラスター中心と隣接する決定領域の間のユークリッド距離を比較することによって，1 つの部分領域を除くその他の部分領域は、隣接する適切な決定領域に併合される。

ここまでは線形分離可能なデータに対しての手法であるが，非線形なデータに対しては，カーネル法 [13][14] と呼ばれる非線形な分類境界を効果的に求めることができる技術がある。これは，データ空間から高次元特徴空間への写像を行い，高次元上での内積を表すカーネル関数を用いることで非線形分離を可能にするものである。実際に，円の周りに輪をもつような非線形データに対してカーネル法を用いた K-Means アルゴリズムを適用したときの分類結果を示す。

本論文では，第 2 章において，K-Means と LVQ によるクラスタリングアルゴリズムについて述べる。第 3 章では，分割・併合機能を有する K-Means アルゴリズムによるクラスタリングを新たに提案し，手法の詳細を説明する。第 4 章では，高次元特徴空間におけるカーネル関数を用いた K-Means と LVQ によるクラスタリング手法を述べる。第 5 章において，分割・併合機能を有する K-Means アルゴリズムによるクラスタリング, K-Means,LVQ の 2 つのアルゴリズムを用いた分割・併合によるクラスタリングの分類結果をそれぞれ示し, 提案手法の有効性を確認する。さらに，カーネル法を用いた K-Means による分類結果とその有効性について検証する。最後に，第 6,7 章で，まとめと今後の課題を述べる。

2 K-Means 法

K-Means 法では，クラスターの数をあらかじめ指定し，個体を K 個のクラスターに分割する。分類の基準として，クラスターの中心と各個体との間のユークリッド距離の 2 乗を用いる。

2.1 K-Means アルゴリズム

n 個の個体を $\vec{x}_i = (x_{i1}, \ldots, x_{iD}), i = 1, \ldots, n$ で表わし，この個体の集合を X とする。データを K 個の重なりのないクラス $X_k, k = 1, \ldots, K$ に分類するため，次の目的関数を用いる。

$$J = \min_{\{\vec{c}_k, k=1,\ldots,K\}} \sum_{i=1}^{n} \sum_{\vec{x}_i \in X_k} \|\vec{x}_i - \vec{c}_k\|^2. \tag{1}$$

ここで，$\|\vec{x}_i - \vec{c}_k\|^2 = \sum_{d=1}^{D} (x_{id} - c_{kd})^2$ とし，クラスター中心は $\vec{c}_k = (c_{k1}, \ldots, c_{kD})$ である。式 (1) について最小化を行うため，K-Means アルゴリズムと呼ばれている以下の反復アルゴリズムを用いる。

アルゴリズム **km**

(km1) $\{\vec{c}_k^{(t)}\}$ が与えられた時，それぞれの $\vec{x}_i$ に関して，次式を計算する。

$$\alpha = \arg\min_k \|\vec{x}_i - \vec{c}_k^{(t)}\|^2. \tag{2}$$

このとき，$\vec{x}_i \in X_\alpha^{(t)}$ と決定する。

(km2) $\{X_k^{(t)}\}$ が与えられた時，次式を計算する。

$$\vec{c}_k^{(t+1)} = \frac{1}{n_k^{(t)}} \sum_{\vec{x}_i \in X_k^{(t)}} \vec{x}_i, \ k = 1, \ldots, K. \tag{3}$$

ここで，$n_k^{(t)}$ は $X_k^{(t)}$ に属する個体の数であり，$\vec{c}_k^{(t+1)}$ は $X_k^{(t)}$ の $(k+1)$ 回目のクラスター中心でありクラスターの代表点に相当する。

ある小さい定数を ϵ とし，すべての k について終了条件である $|\vec{c}_k^{(t+1)} - \vec{c}_k^{(t)}| < \epsilon$ を満たすまで，(km1) と (km2) の手順を繰り返す。

$\{\vec{c}_k^{(t)}\}$ の収束値を $\{\vec{c}_k\}$ で表わすと，この値は必ずしも式 (1) の最小化を満たすとは限らない。決定領域 R_k は，

$$R_k = \{\vec{x} \mid \|\vec{x} - \vec{c}_k\|^2 < \|\vec{x} - \vec{c}_i\|^2 \text{ for all } i \neq k\} \tag{4}$$

で表わされる。このとき，

$$\text{if } \ \vec{x} \in R_k, \ \vec{x} \in \text{Class } k \tag{5}$$

が成り立ち，決定領域は $\{\vec{c}_k\}$ を母点とするボロノイ図で表わされる。

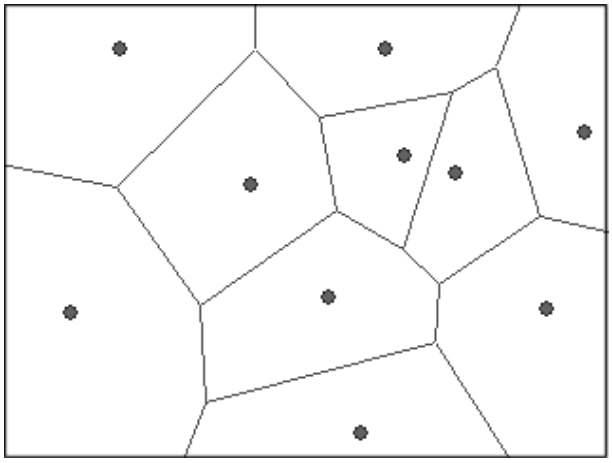

図 1: ボロノイ分割：母点を結ぶ線分の垂直二等分線で構成される

2.2 K-Means アルゴリズムの変形

K-Means アルゴリズムでは n 個の個体すべてをクラスターに割り当てた後，クラスターの中心を更新した。このアルゴリズムを変形し，個体の割り当てが変わるごとにクラスター中心を更新するようなアルゴリズムを考える。

X_p に属する個体 $\vec{x}$ が別のクラスター X_q に最も近く，個体 $\vec{x}$ を X_p から X_q へ移動させるとする。このとき，クラスターは X_p が $X_p - \{\vec{x}\}$，X_q が $X_q \cup \{\vec{x}\}$ と変化する。個体が移動する前のクラスター中心を $\vec{c}_p$，$\vec{c}_q$ とし，移動後をそれぞれ $\vec{c}_p^*$，$\vec{c}_q^*$ とする。このとき，X_p^* に属する個体の数が $N_p - 1$, X_q^* に属する個体の数が $N_q + 1$ となっていることを考慮した評価関数 J の値の変化に基づくアルゴリズム [1] がある。

アルゴリズム **km'**

(km'1) 初期値としてクラスター中心を与えクラスターを生成する。
中心 $\vec{c}_k^{(t)}, k = 1, \ldots, K$ を計算する。

(km'2) 目的関数 J の値が減少しないことを収束条件とし，すべての $\vec{x}_i, i = 1, \ldots, N$ について次の手順を繰り返す。

(km'2-1) $\vec{x}_i$ が X_p に属していると仮定する。$N_p \neq 1$ ならば次を計算する。

$$J_p = \frac{N_p}{N_p - 1}\|\vec{x}_i - \vec{c}_p\|^2$$
$$J_l = \frac{N_l}{N_l + 1}\|\vec{x}_i - \vec{c}_l\|^2,\ l \neq p$$

もし，$J_q \leq J_l, for\ all\ l$ ならば，$\vec{x}_i$ を X_p から X_q へ移動する。

$$\vec{c}_p^* = \vec{c}_p - \frac{1}{N_p - 1}(\vec{x}_i - \vec{c}_p)$$
$$\vec{c}_q^* = \vec{c}_q - \frac{1}{N_q + 1}(\vec{x}_i - \vec{c}_q)$$

によってクラスター中心を再計算し，

$$N_p^* = N_p - 1$$
$$N_q^* = N_q + 1$$

とする。最後に目的関数 J を更新する。

$$J^* = J + J_q - J_p$$

このアルゴリズムは，クラスター X_p が $X_p^* = X_p - \{\vec{x}_i\}$，$X_q$ が $X_q^* = X_q \cup \{\vec{x}_i\}$ と変化することによって，

$$\sum_{\vec{x} \in X_p^*} \|\vec{x} - \vec{c}_p^*\|^2 = \sum_{\vec{x} \in X_p} \|\vec{x} - \vec{c}_p\|^2 - \frac{N_p}{N_p - 1}\|\vec{x}_i - \vec{c}_p\|^2$$
$$\sum_{\vec{x} \in X_q^*} \|\vec{x} - \vec{c}_q^*\|^2 = \sum_{\vec{x} \in X_q} \|\vec{x} - \vec{c}_q\|^2 - \frac{N_q}{N_q + 1}\|\vec{x}_i - \vec{c}_q\|^2$$

と変化することを用いている。

2.3 学習ベクトル量子化によるクラスタリング

学習ベクトル量子化 (LVQ) によるクラスタリングは，競合学習の概念にもとづいており，K-Means アルゴリズムと同じく代表的な手法である。

時間を表す変数を t とし，$t = 1, 2, \ldots$ と離散的な値をとるものとする。$x(t) \in R^p (t = 1, 2, \ldots)$ はある確率分布からとられた信号の無限列であるとする。ベクトル量子化 (VQ) は有限個のコードブックと呼ばれるベクトル $m_k \in R^p, k = 1, \ldots, K$ によって信号の近似を行う方法である。したがって，次式のように $x(t)$ は入力空間での $x(t)$ に最も近いコードブック

ベクトル m_l に近似される。

$$m_l(t) = \arg\min_k \|x(t) - m_k(t)\|. \tag{6}$$

$x(t)$ に適合するコードブック $m_l(t)$ が決定された時，現在の入力にさらに適合するようにコードブックベクトルを更新する。

$$m_l(t+1) = m_l(t) + \alpha(t)[x(t) - m_l(t)]. \tag{7}$$

ここで，$\alpha(t)$ は学習率パラメータと呼ばれ次の条件を満たす。

$$\sum_{t=1}^{\infty} \alpha(t) = \infty, \quad \sum_{t=1}^{\infty} \alpha^2(t) < \infty, \quad t = 1, 2, \ldots \tag{8}$$

$\alpha(t)$ は単調減少関数であり，時間とともに減少していく。たとえば，$\alpha(t) =$ 定数$/t$ とすることもできる。

次に，ベクトル量子化のアルゴリズムをクラスタリングに導入する。まず，ベクトル量子化における無限信号列 $x(t)$ を有限個のデータで表す必要がある。これには個体データを繰り返し用いるか，もしくは，個体の集合からランダムにデータを選択し $x(t)$ に割り当てることが考えられる。また，1つのコードブックベクトルを1つのクラスターの代表点として考え，最近隣法で個体をクラスターに割り当てることでクラスタリングに応用することができる。学習ベクトル量子化にもとづくクラスタリングのアルゴリズムは次のようになる。

アルゴリズム **lvq**

(lvq1) $m_k, k = 1, \ldots, K$ の初期値を設定する。

(lvq2) $t = 1, 2, \ldots$ について収束するまで次の手順を繰り返す。

(lvq2-1) 次を求める。

$$m_l = \arg\min_{1 \le k \le K} \|x(t) - m_k(t)\| \tag{9}$$

(lvq2-2) $m_1(t), \ldots, m_k(t)$ を更新する。

$$\begin{aligned} m_l(t+1) &= m_l(t) + \alpha(t)[x(t) - m_l(t)] \\ m_k(t+1) &= m_k(t), k \neq l \end{aligned}$$

$x(t)$ が表す個体をクラスター X_l に割り当てる。

このアルゴリズムでは，データが時間とともに1つずつ入力されていき，クラスター中心との内積が最も大きいクラスターに割り当てられていく。つまり，入力されたデータと各クラスター中心との位置関係だけでクラスタリングが行われていることになる。この手法のことをオンライン型クラスタリングと呼ぶ。

3　分割・併合機能を有する K-Means アルゴリズム

上で述べた K-Means アルゴリズムを用いて，X に属する個体を分類し，決定領域 $\{R_k\}$ とクラスター中心 $\{\vec{c}_k\}$ を得たと仮定する。もし，それぞれの R_k において，クラスターがただ１つであると決定された場合，クラスタリングの過程を終了する。しかし，R_k に２つ以上のクラスターが存在すると判断された場合，R_k を分割し，１つの適切なクラスターを R_k に残し，その他の誤って分類されているクラスターを隣接する決定領域に併合する。この問題を解決するために，次の方法を提案する。K-Means アルゴリズムを，それぞれの R_k に属する個体に再度適用し，R_k と K-Means アルゴリズムを用いて作成した R_k の部分領域に対して，分割・併合の手順を適用する。

3.1　決定領域の分割

K-Means アルゴリズムを $K = m$ $(2 \leq m \leq M)$ として，R_k に属する個体に適用し，R_k を m 個の部分領域に分割する。ここで，部分領域とそのクラスター中心をそれぞれ $\{R_{k,p}^{(m)}, p = 1, \ldots, m\}$, $\{\vec{c}_{k,p}^{(m)}, p = 1, \ldots, m\}$ とする。通常のクラスタリングでは，M の値は 2，もしくは 3 を用いる。

R_k の 2 乗歪みは，次式で定義される。

$$D_k^{(m=1)} = \sum_{\vec{x}_i \in R_k} \|\vec{x}_i - \vec{c}_k\|^2. \tag{10}$$

$R_{k,p}^{(m)}$ の歪みを，

$$D_{k,p}^{(m)} = \sum_{\vec{x}_i \in R_{k,p}^{(m)}} \|\vec{x}_i - \vec{c}_{k,p}^{(m)}\|^2, \ m = 2, \ldots, M \tag{11}$$

と定義すると，m 個の部分領域をもつ $R_k^{(m)}$ の歪みは次のように表わすことができる。

$$D_k^{(m)} = \sum_{p=1}^{m} D_{k,p}^{(m)}. \tag{12}$$

次に，R_k の分割の尺度として次式を導入する。

$$\rho_k(m) = D_k^{(m)} / D_k^{(m-1)}, \ m = 2, \ldots, M. \tag{13}$$

$\rho_k(m)$ の急激な減少は，R_k が m 個の部分領域に分割されるということを示している。

R_k が m 個のクラスターから構成されると仮定するとき，R_k を $m-1$ 個の部分領域に分割した場合，それぞれの部分領域のクラスター中心は，本来のクラスターの代表点と一致しない。しかし，R_k を m 個の部分領域に分割した場合，そのクラスター中心は，本来のクラ

スターの代表点と一致する可能性が大きい。この時，$D_k^{(m)}$ の値は，$D_k^{(m-1)}$ の値に比べて急激に減少する。この状況を 3 節の分類実験において実証する。

分割の尺度である

$$\rho_k(m^*) = \min_m\{\rho_k(m),\ m = 2, \ldots, M\} \tag{14}$$

を求め，ある値 η について，

$$\rho_k(m^*) < \eta \tag{15}$$

ならば，R_k を m^* 個の部分領域に分割する。そうでなければ，R_k を分割しない。η の値は，実験結果をもとにして定め，η の値が小さいほど，分割の可能性は低くなる。

3.2 部分領域の併合

決定領域 R_k を部分領域 $\{R_{k,p}^{(m^*)}, p = 1, \ldots, m^*\}$ に分割し，$\{R_{k,p}^{(m^*)}\}$ のクラスター中心 $\{\vec{c}_{k,p}^{(m^*)}, p = 1, \ldots, m^*\}$ が K-Means アルゴリズムによって得られたとすると，ただ 1 つの部分領域が，更新された新しい決定領域 R_k となり，その他の $m^* - 1$ 個の部分領域は，隣接する決定領域に併合される。

$\vec{c}_{k,p}^{(m^*)}$ と R_k に隣接する決定領域の境界線までの最小となるユークリッド距離 $d(\vec{c}_{k,p}^{(m^*)})$ を求める。

$$\hat{d}(\vec{c}_{k,p^*}^{(m^*)}) = \max_p\{d(\vec{c}_{k,p}^{(m^*)}),\ p = 1, \ldots, m^*\} \tag{16}$$

で得られた部分領域 $R_{k,p^*}^{(m^*)}$ は，新たな決定領域 R_k となる。$p \neq p^*$ であるその他の部分領域 $R_{k,p}^{(m^*)}$ は，$d(\vec{c}_{k,p}^{(m^*)})$ を満たす隣接する決定領域に併合される。

また，この併合のための距離尺度として，部分領域と隣接する決定領域に含まれるデータ間のユークリッド距離を用いることもできる。この距離の最短を

$$d(R_{k,p}^{(m^*)}) = \min_{\vec{x}_i \in R_{k,p}^{(m^*)}, \vec{x}_j \in R_l, l \neq k} d(\vec{x}_i, \vec{x}_j) \tag{17}$$

とすると，

$$\hat{d}(R_{k,p^*}^{(m^*)}) = \max_p\{d(R_{k,p}^{(m^*)}),\ p = 1, \ldots, m^*\} \tag{18}$$

で得られる部分領域 $R_{k,p^*}^{(m^*)}$ は，新たな決定領域 R_k となり，その他の部分領域 $R_{k,p}^{(m^*)}$ は，$d(R_{k,p}^{(m^*)})$ を満たす隣接する決定領域に併合される。

最後に，この手法によるクラスタリングの有効性を確認するために次の手順を行う。すべての k について，分割の尺度が $\rho_k(m^*) < \eta$ であれば，クラスタリング結果は有効であるとする。そうでなければ，クラスタリング結果を無効とし，その過程を記述してクラスタリングを中止する。

4 高次元特徴空間におけるカーネル関数

線形分離ができないデータに対して，元の特徴空間を高次元特徴空間に写像して線形分離可能にする方法がSVM(Support Vector Machine)[15]において用いられる。無限次元を含む高次元特徴空間を H で表すと，$\Phi : R^p \to H$ の非線形写像を行う。高次元ベクトル空間 H において，

$$K(\vec{x}, \vec{y}) = \langle \Phi(\vec{x}), \Phi(\vec{y}) \rangle_H \tag{19}$$

で表わされるカーネルを用いることで，データに対する特徴ベクトル $\Phi(\vec{x})$ を計算することなく，その内積を求めることができる。主なカーネル関数として次のものがある。

カーネル関数

RBF カーネル

$$K(\vec{x}, \vec{y}) = \exp(-C\|\vec{x} - \vec{y}\|^2) \tag{20}$$

多項式カーネル

$$K(\vec{x}, \vec{y}) = (1 + \langle \vec{x}, \vec{y} \rangle)^d \tag{21}$$

4.1 カーネル関数を用いた K-Means アルゴリズムによるクラスタリング

カーネル関数を K-Means アルゴリズムを用いたクラスタリングに導入する。式 (1) で表わされる K-Means 法の目的関数をクラスター中心を用いることなしに記述すると，高次元空間上での目的関数は

$$J = \sum_{k=1}^{K} \sum_{\vec{x}_i \in X_k} \|\Phi(\vec{x}_i) - \vec{m}_k\|^2 \tag{22}$$

と表せる。ここで，$\vec{m}_k$ は特徴空間上でのクラスター X_k の中心を表している。

$$\vec{m}_k = \frac{\sum_{\vec{x} \in X_k} \Phi(\vec{x})}{n_k}. \tag{23}$$

高次元な写像 Φ を直接計算することは，計算量や記憶量の観点から難しく，カーネル化の際に写像 Φ を明示的に用いてはいけない。そこで，$\|\Phi(\vec{x}_i) - \vec{m}_k\|^2$ をカーネル関数 $K(\vec{x}_i, \vec{x}_j)$ で置き換える。高次元上のデータとクラスター中心の距離を次のように定義すると，

$$D_{ik} = \|\Phi(\vec{x}_i) - \vec{m}_k\|_H^2. \tag{24}$$

式 (22) は，

$$J = \sum_{k=1}^{K} \sum_{\vec{x}_i \in X_k} D_{ik} \tag{25}$$

と表すことができる。

ここで,

$$\begin{aligned} D_{ik} &= \|\Phi(\vec{x}_i) - \vec{m}_k\|^2 \\ &= \langle \Phi(\vec{x}_i) - \frac{\sum_{\vec{x}_j \in X_k} \Phi(\vec{x}_j)}{n_k}, \Phi(\vec{x}_i) - \frac{\sum_{\vec{x}_l \in X_k} \Phi(\vec{x}_l)}{n_k} \rangle \\ &= \langle \Phi(\vec{x}_i), \Phi(\vec{x}_i) \rangle - \frac{2}{n_k} \sum_{\vec{x}_j \in X_k} \langle \Phi(\vec{x}_i), \Phi(\vec{x}_j) \rangle + \frac{1}{n_k^2} \sum_{\vec{x}_j, \vec{x}_l \in X_k} \langle \Phi(\vec{x}_j), \Phi(\vec{x}_l) \rangle \\ &= K(\vec{x}_i, \vec{x}_i) - \frac{2}{n_k} \sum_{\vec{x}_j \in X_k} K(\vec{x}_i, \vec{x}_j) + \frac{1}{n_k^2} \sum_{\vec{x}_j, \vec{x}_l \in X_k} K(\vec{x}_j, \vec{x}_l) \end{aligned} \tag{26}$$

となり，目的関数をカーネル関数を使って表すことができる。これを用いることで，非線形分離が可能なクラスタリングを実現することができる。アルゴリズムは次のようになる。

アルゴリズム **Kkm**

(Kkm1) 個体の集合 X から，初期クラスター中心として K 個のランダムな点 $\vec{y}_j (j = 1, ..., K)$ をとる。個体 $\vec{x}_i$ の最初の割り当てを次式で計算する。

$$\begin{aligned} \alpha &= \arg\min_k \|\Phi(\vec{x}_i) - \Phi(\vec{y}_j)\|^2 \\ &= \arg\min_k K(\vec{x}_i, \vec{x}_i) - 2K(\vec{x}_i, \vec{y}_j) + K(\vec{y}_j, \vec{y}_j). \end{aligned} \tag{27}$$

このとき，$\vec{x}_i \in X_\alpha^{(t)}$ と決定する。

(Kkm2) クラスター間で個体の移動が起こらないことを収束条件とし，すべての個体 $\vec{x}_i$ について次の手順を繰り返す。

(Kkm2-1) 個体 $\vec{x}_i$ を最も近いクラスターに割り当てる。

$$\alpha = \arg\min_k \|\Phi(\vec{x}_i) - \vec{m}_k\|^2 \tag{28}$$

(Kkm2-2) 距離 $\|\Phi(\vec{x}_i) - \vec{m}_k\|^2$ を式 (26) を用いて更新する。

4.2 カーネル関数を用いた変形 K-Means アルゴリズムによるクラスタリング

4.1 のアルゴリズムでは n 個の個体すべてをクラスターに割り当てた後，クラスターの中心を更新した。このアルゴリズムを変形し，個体の割り当てが変わるごとにクラスター中心を更新するようなアルゴリズムを考える。なお，目的関数は式 (25) を用いる。

アルゴリズム **Kkm'**

(Kkm'1) 個体の集合 X から，初期クラスター中心として K 個のランダムな点 $\vec{y}_j (j = 1, ..., K)$ をとる。個体 $\vec{x}_i$ の最初の割り当てを式 (27) で計算する。

(Kkm'2) すべての個体 $\vec{x}_i$ について目的関数が収束するまで次の手順を繰り返す。

(Kkm'2-1) 個体 $\vec{x}_i$ を最も近いクラスターに割り当てる。

$$\alpha = \arg\min_k \|\Phi(\vec{x}_i) - \vec{m}_k\|^2$$

(Kkm'2-2) 個体 $\vec{x}_i$ の割り当てが変化した場合，目的関数を再計算する。

(Kkm'2-2) における目的関数の更新は以下の通りである。便宜上，目的関数値を

$$J_k = \sum_{\vec{x}_i \in X_k} D_{ik} \tag{29}$$

と表わす。ここで,

$$J = \sum_{k=1}^{K} J_k. \tag{30}$$

次に，個体 $\vec{x}_i$ がクラスター X_p から Xq へ移動したとすると，更新後の X_p,X_q, クラスター中心 $\vec{m}_p$,$\vec{m}_q$ はそれぞれ以下のようになる。N_p,N_q はそれぞれのクラスターに属する個体の数とする。

$$\begin{aligned}
X_p^* &= X_p - \{\vec{x}_i\}, \\
X_q^* &= X_q \cup \{\vec{x}_i\}, \\
\vec{m}_p^* &= \frac{N_p}{N_p - 1}\|\Phi(\vec{x}_i) - \vec{m}_p\|^2, \\
\vec{m}_q^* &= \frac{N_q}{N_q + 1}\|\Phi(\vec{x}_i) - \vec{m}_q\|^2.
\end{aligned}$$

更新前の目的関数値を J,J_p,J_q，更新後の目的関数値を J^*,J_p^*,J_q^* とすると次式が成り立つ。

$$J^* = \sum_k J_k^*$$

$$= J_p^* + J_q^* + \sum_{k \neq p,q} J_k$$

ここで,

$$\begin{aligned} J_p^* &= \sum_{\vec{x} \in X_p} \|\Phi(\vec{x}) - \vec{m}_p^*\|^2 - \|\Phi(\vec{x}_i) - \vec{m}_p^*\|^2 \\ &= \sum_{\vec{x} \in X_p} \|\Phi(\vec{x}) - \vec{m}_p + \frac{\Phi(\vec{x}_i) - \vec{m}_p}{N_p - 1}\|^2 - \frac{N_p}{N_p - 1}\|\Phi(\vec{x}_i) - \vec{m}_p\|^2 \\ &= J_p - \frac{N_p}{N_p - 1}\|\Phi(\vec{x}_i) - \vec{m}_p\|^2 \\ &= J_p - \frac{N_p}{N_p - 1} D_{ip} \end{aligned} \tag{31}$$

と導ける。同様に,

$$J_q^* = J_q + \frac{N_q}{N_q + 1} D_{iq}. \tag{32}$$

よって，目的関数値の更新は次のようになる。

$$\begin{aligned} J^* &= J_p^* + J_q^* + \sum_{k \neq p,q} J_k \\ &= J_p - \frac{N_p}{N_p - 1} D_{ip} + J_q + \frac{N_q}{N_q + 1} D_{iq} + \sum_{k \neq p,q} J_k \\ &= J - \frac{N_p}{N_p - 1} D_{ip} + \frac{N_q}{N_q + 1} D_{iq} \end{aligned} \tag{33}$$

4.3 カーネル関数を用いた学習ベクトル量子化によるクラスタリング

LVQ によるクラスタリングにカーネルを用いることで，非線形なデータの分類を行うことができるクラスタリングアルゴリズムに拡張する。

高次元特徴空間上での LVQ は次式のように表せる。

$$\vec{m}_l(t) = \arg\min_k \|\Phi(\vec{x}_h) - \vec{m}_k(t)\| \tag{34}$$

$$\vec{m}_l(t+1) = \vec{m}_l(t) + \alpha(t)[\Phi(\vec{x}_h) - \vec{m}_l(t)] \tag{35}$$

ここで $\vec{m}$ は特徴空間上でのクラスター中心である。カーネルを利用する場合，特徴空間におけるクラスター中心 $\vec{m}$ を明示的に用いることができない。そこで，時刻 t における高次元特徴空間上でのデータとクラスター中心の距離を非類似度として次のように定義する。

$$D_{ik}(t) = \|\Phi(\vec{x}_i) - \vec{m}_k(t)\|^2 \tag{36}$$

この非類似度とカーネル関数を用いて計算を進める必要がある。式 (34) は，次のように表すことができる。

$$D_{il}(t) = \arg\min_k D_{ik}(t) \tag{37}$$

クラスター中心を更新するということは，クラスター中心と個体との距離が更新されることと同じであると考えられる。よって式 (35) は，

$$\begin{aligned} D_{ik}(t+1) &= \|\Phi(\vec{x}_i) - \vec{m}_k(t+1)\|^2 \\ &= \langle\Phi(\vec{x}_i), \Phi(\vec{x}_i)\rangle - 2\langle\Phi(\vec{x}_i), \vec{m}_l(t+1)\rangle + \langle\vec{m}_l(t+1), \vec{m}_l(t+1)\rangle \end{aligned} \tag{38}$$

と置き換えることができる。この式に，式 (35) を代入すると次のようになる。簡単のため，$\alpha = \alpha(t)$ と略記する。

$$\begin{aligned} D_{ik}(t+1) &= \langle\Phi(\vec{x}_i), \Phi(\vec{x}_i)\rangle - 2\{(1-\alpha)\langle\Phi(\vec{x}_i), \vec{m}_l(t)\rangle + \alpha\langle\Phi(\vec{x}_i), \Phi(\vec{x}_h)\rangle\} \\ &+ \{(1-\alpha)^2\langle\vec{m}_l(t), \vec{m}_l(t)\rangle + 2\alpha(1-\alpha)\langle\Phi(\vec{x}_h), \vec{m}_l(t)\rangle\} \\ &+ \alpha^2\langle\Phi(\vec{x}_h), \Phi(\vec{x}_h)\rangle \end{aligned} \tag{39}$$

これを距離についてまとめると

$$\begin{aligned} D_{il}(t+1) &= (1-\alpha)D_{il}(t) - \alpha(1-\alpha)D_{hl}(t) \\ &+ \alpha\{K(\vec{x}_i, \vec{x}_i) - 2K(\vec{x}_i, \vec{x}_h) + K(\vec{x}_h, \vec{x}_h)\} \end{aligned} \tag{40}$$

という距離に関する更新式を得ることができる。カーネル関数を用いた LVQ によるクラスタリングアルゴリズムを示す。

アルゴリズム **Klvq**

(Klvq1) $D_{ik}, i = 1, \ldots, n, k = 1, \ldots, K$ の初期値を設定しする。

(Klvq2) $t = 1, 2, \ldots$ について収束するまで次の手順を繰り返す。

(Klvq2-1) 次を求め，個体 $\vec{x}_i$ をクラスター X_l に割り当てる。

$$D_{il}(t) = \min_k D_{ik}(t)$$

(Klvq2-2) D_{il} を式 (40) を用いて更新する。

5　実験

5.1　分割・併合機能を有する K-Means アルゴリズム

図 2 に示すような，3 つのクラスからなるデータについて実験を行う。クラス 1 は，平均 $(x_1, x_2) = (0, 0)$，分散 $(x_1, x_2) = (0.1, 0.1)$ のガウス密度関数 [10] による擬似乱数である。クラス 2 は，平均 $(x_1, x_2) = (5, 0)$，分散 $(x_1, x_2) = (2, 2)$ のガウス密度関数による擬似乱数である。クラス 3 は，平均 $(x_1, x_2) = (1, 4)$，分散 $(x_1, x_2) = (0.2, 0.2)$ のガウス密度関数による擬似乱数である。

各クラスのセントロイドは，それぞれ $(0.0861, -0.113)$，$(4.98, 0.163)$，$(1.10, 4.04)$ である。x_1 と x_2 も統計的に独立であるとする。

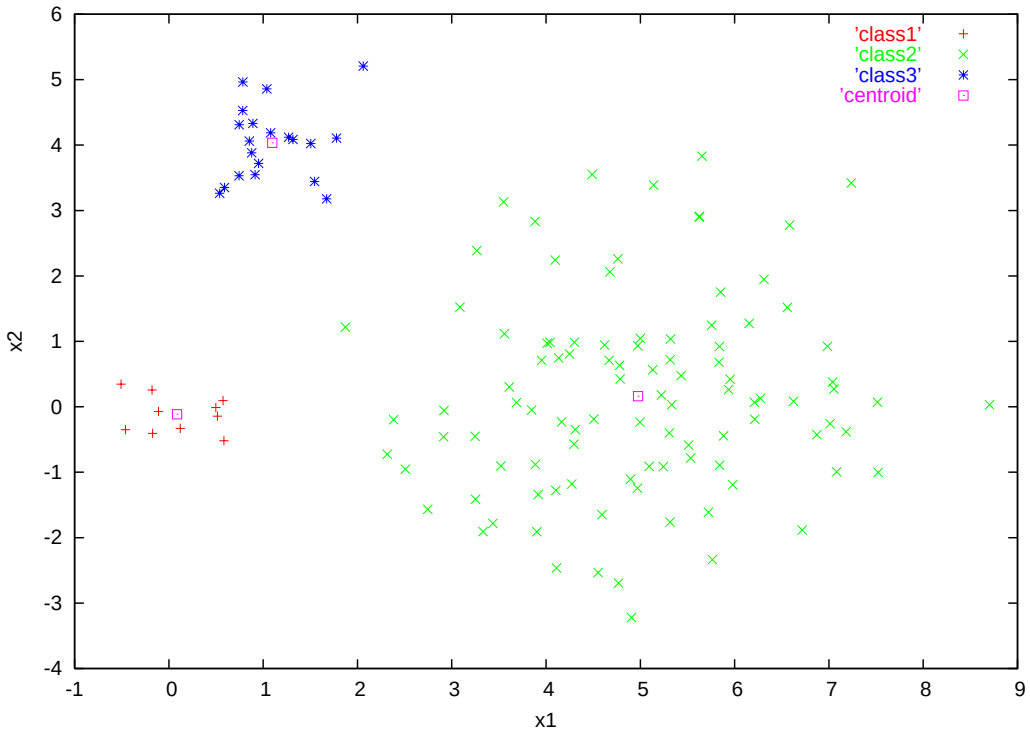

図 2: 3 つのクラスから構成されるデータ

図 3 は K-Means アルゴリズムによる分類結果を示している。決定領域は，”line1,line2,line3” で示される 3 つの直線で分けられる。ここで，決定領域 R_1 とは line1 と line3 によって作られる領域のことである。また，クラスター中心として，$\vec{c}_1 = (1.67, -0.383)$, $\vec{c}_2 = (5.36, 0.146)$, $\vec{c}_3 = (1.55, 3.86)$ を得る。それぞれのクラスが統計的に異なる分布を有する時，K-Means アルゴリズムによる典型的な誤りのある分類結果を得ることになり，これは図 3 より明らかである。ここで，130 個の個体のうちの 17 個の個体が誤った分類になっている。クラスタリングの結果は，表 1 にまとめている。

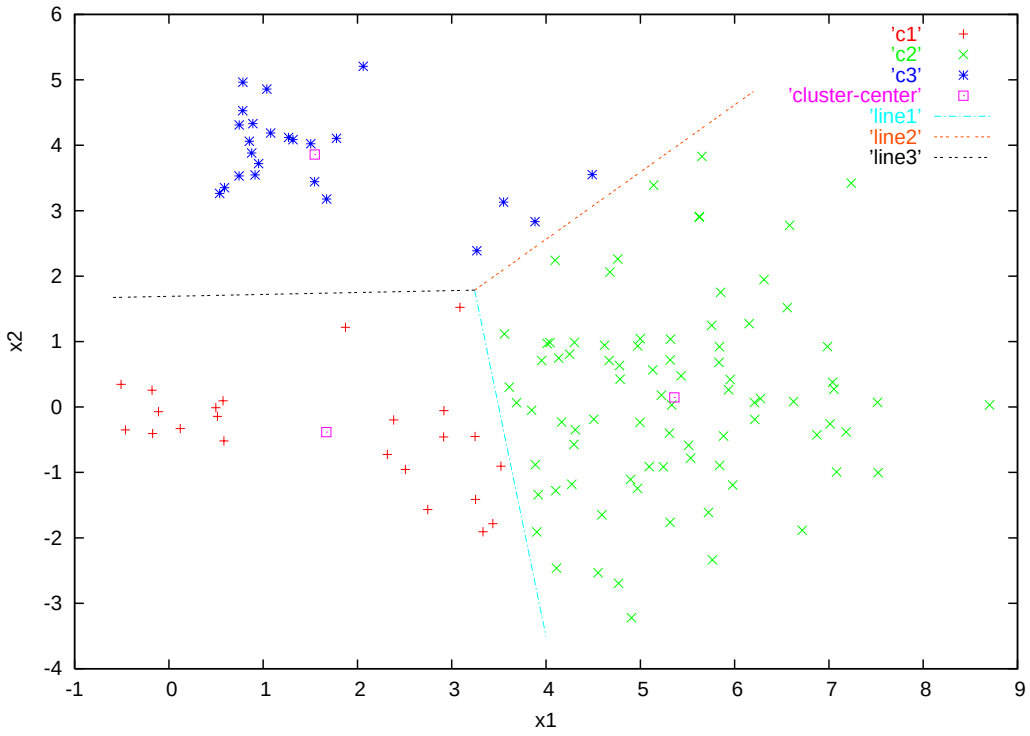

図 3: $K = 3$ での K-Means アルゴリズムによるクラスタリング

表 1: K-Means アルゴリズムによるクラスタリング結果と決定領域

決定領域	データ数	クラスター中心
R_1	23	$\vec{c}_1 = (1.67, -0.383)$
R_2	83	$\vec{c}_2 = (5.36, 0.146)$
R_3	24	$\vec{c}_3 = (1.55, 3.86)$

表 2 は，K-Means アルゴリズムによるクラスタリングで得られた各決定領域の歪みを D_k，それぞれのクラスに正しいセントロイドを与えてボロノイ分割を行った時の各領域の歪みを D_c，３つのクラスを誤りなく完全に分類した時の各領域の歪みを D_p として比較したものである。K-Means アルゴリズムによるクラスタリングで得られた各決定領域における歪みの和は，３つのクラスを完全に正しく分類した時の歪みの和よりも小さくなるが，クラスタリング結果は図 3 の様に誤りを含む分類になっている。仮にそれぞれのクラスのセントロイドが正しく求まった場合でも，図 2 のように統計的に分布の異なるクラスをボロノイ分割で分離すると分類誤りが生じ，K-Means アルゴリズムでは効果的な分類をすることはできない。そこで各決定領域に対して，分割・併合を行い正しく分類する必要がある。

部分領域の数 M=3 を最大値として選び，K=m で再び K-Means アルゴリズムを適用することによって，それぞれの決定領域を m 個の部分領域に分割する。$m = 2, 3$ とした時の，決定領域 $\{R_k\}$ の K-Means アルゴリズムによるクラスタリング結果を表 3,4 に示す。

表 2: 各領域における歪みの比較

決定領域	D_k	D_{c}	D_p
R_1	64.3	24.6	2.43
R_2	168.6	349.5	401.0
R_3	39.1	23.8	9.52
和	373.0	397.8	412.9

表 3: $m = 2$ での K-Means アルゴリズムによるクラスタリング

部分領域	データ数	クラスター中心
$R_{1,1}^{(m=2)}$	10	$\vec{c}_{1,1}^{(m=2)} = (0.0861, -0.113)$
$R_{1,2}^{(m=2)}$	13	$\vec{c}_{1,2}^{(m=2)} = (2.89, -0.590)$
$R_{2,1}^{(m=2)}$	45	$\vec{c}_{2,1}^{(m=2)} = (4.72, -0.579)$
$R_{2,2}^{(m=2)}$	38	$\vec{c}_{2,2}^{(m=2)} = (6.16, 1.05)$
$R_{3,1}^{(m=2)}$	20	$\vec{c}_{3,1}^{(m=2)} = (1.10, 4.04)$
$R_{3,2}^{(m=2)}$	4	$\vec{c}_{3,2}^{(m=2)} = (3.80, 2.98)$

表 4: $m = 3$ での K-Means アルゴリズムによるクラスタリング

部分領域	データ数	クラスター中心
$R_{1,1}^{(m=3)}$	10	$\vec{c}_{1,1}^{(m=3)} = (0.0861, -0.113)$
$R_{1,2}^{(m=3)}$	2	$\vec{c}_{1,2}^{(m=3)} = (2.48, 1.37)$
$R_{1,3}^{(m=3)}$	11	$\vec{c}_{1,3}^{(m=3)} = (2.96, -0.947)$
$R_{2,1}^{(m=3)}$	29	$\vec{c}_{2,1}^{(m=3)} = (4.70, -1.10)$
$R_{2,2}^{(m=3)}$	34	$\vec{c}_{2,2}^{(m=3)} = (5.15, 1.46)$
$R_{2,3}^{(m=3)}$	20	$\vec{c}_{2,3}^{(m=3)} = (6.78, -0.218)$
$R_{3,1}^{(m=3)}$	19	$\vec{c}_{3,1}^{(m=3)} = (1.05, 3.97)$
$R_{3,2}^{(m=3)}$	1	$\vec{c}_{3,2}^{(m=3)} = (2.06, 5.21)$
$R_{3,2}^{(m=3)}$	4	$\vec{c}_{3,2}^{(m=3)} = (3.80, 2.98)$

式 (10) と (12) から求めた，$\{R_k\}$ と $\{R_k^{(m)}\}$ における歪み $\{D_k^{(m)}\}$ を表 5 に示す。表 6 は，式 (13) によって与えられる分割の尺度を表わしている。表 6 の $\{\rho_k^{(m)}\}$ の値に注目すると，$\rho_{k=1}^{(m=2)}$ と $\rho_{k=3}^{(m=2)}$ の値が小さいことがわかる。式 (15) において $\eta \approx 0.4$ とすると，m^*=2 となり，決定領域 R_1 と R_3 は，それぞれ 2 つの部分領域に分割されるべきであると考えられる。

表 5: $\{R_k\}$, $\{R_k^{(m)}\}$ における歪み $\{D_k^{(m)}\}$

	$D_k^{(m=1)}$	$D_k^{(m=2)}$	$D_k^{(m=3)}$
$k = 1$	64.3	18.7	9.18
$k = 2$	269.6	172.5	110.5
$k = 3$	39.1	11.1	0.28

表 6: 分割の尺度 $\{\rho_k^{(m)}\}$

	$\rho_k^{(m=2)}$	$\rho_k^{(m=3)}$
$k = 1$	0.290	0.492
$k = 2$	0.640	0.641
$k = 3$	0.283	0.781

表 7: 部分領域のクラスター中心と決定領域との距離とその最短距離 $\{d(\bar{c}_{k,p}^{(m^*=2)})\}$

部分領域	R_1	R_2	R_3	$d(\bar{c}_{k,p}^{(m^*=2)})$
$R_{1,1}^{(m^*=2)}$		3.39	1.81	$d(\bar{c}_{1,1}^{(m^*=2)}) = 1.81$
$R_{1,2}^{(m^*=2)}$		0.688	2.36	$d(\bar{c}_{1,2}^{(m^*=2)}) = 0.688$
$R_{3,1}^{(m^*=2)}$	2.31	3.11		$d(\bar{c}_{3,1}^{(m^*=2)}) = 2.31$
$R_{3,2}^{(m^*=2)}$	1.18	0.434		$d(\bar{c}_{3,2}^{(m^*=2)}) = 0.434$

表 8: 部分領域と決定領域のデータ間の距離とその最短距離 $\{d(R_{k,p}^{(m^*=2)})\}$

部分領域	R_1	R_2	R_3	$d(R_{k,p}^{(m^*)})$
$R_{1,1}^{(m^*=2)}$		3.04	3.09	$d(R_{1,1}^{(m^*=2)}) = 3.04$
$R_{1,2}^{(m^*=2)}$		0.366	0.882	$d(R_{1,2}^{(m^*=2)}) = 0.366$
$R_{3,1}^{(m^*=2)}$	1.97	2.60		$d(R_{3,1}^{(m^*=2)}) = 1.97$
$R_{3,2}^{(m^*=2)}$	0.882	0.628		$d(R_{3,2}^{(m^*=2)}) = 0.628$

次に，式 (16) により $\{R_{k,p}^{(m^*=2)}, k = 1, p = 1, 2\}$ と $\{R_{k,p}^{(m^*=2)}, k = 3, p = 1, 2\}$ のうちの適切な部分領域を，隣接する決定領域に併合する。部分領域と決定領域との距離，そして $\{d(\bar{c}_{k,p}^{(m^*=2)})\}$ を表 7 に示す。この表から，$R_{1,2}^{(m^*=2)}$ は R_2 に併合され，$R_{3,2}^{(m^*=2)}$ は R_2 に併合されるべきである。また表 8 は，併合の尺度として，式 (17) で定義される部分領域と隣接する決定領域のデータ間の距離を用いたもので，この表からも同様に，$R_{1,2}^{(m^*=2)}$ は R_2 に併合され，$R_{3,2}^{(m^*=2)}$ は R_2 に併合されるべきであることがわかる。

図 4 は分割・併合を行ったときの，K-Means アルゴリズムによる分類結果を表わしている。決定領域 R_1 は line11 により，2 つの部分領域に分割され，右側の部分領域は決定領域 R_2 に併合される。決定領域 R_3 は line33 により，2 つの部分領域に分割され，右側の部分領域は決定領域 R_2 に併合される。

クラスタリングの最終結果を図 5 に示す。これらのデータに対して，誤りのない，正しい分類結果が得られたことが示されている。

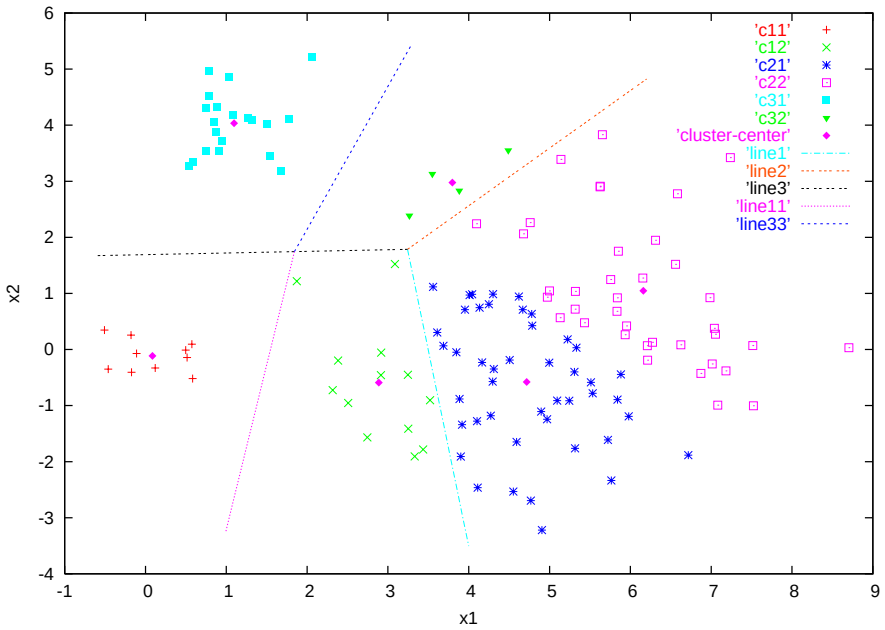

図 4: K=3,m^*=2 の分割・併合機能を有する K-Means アルゴリズムによるクラスタリング

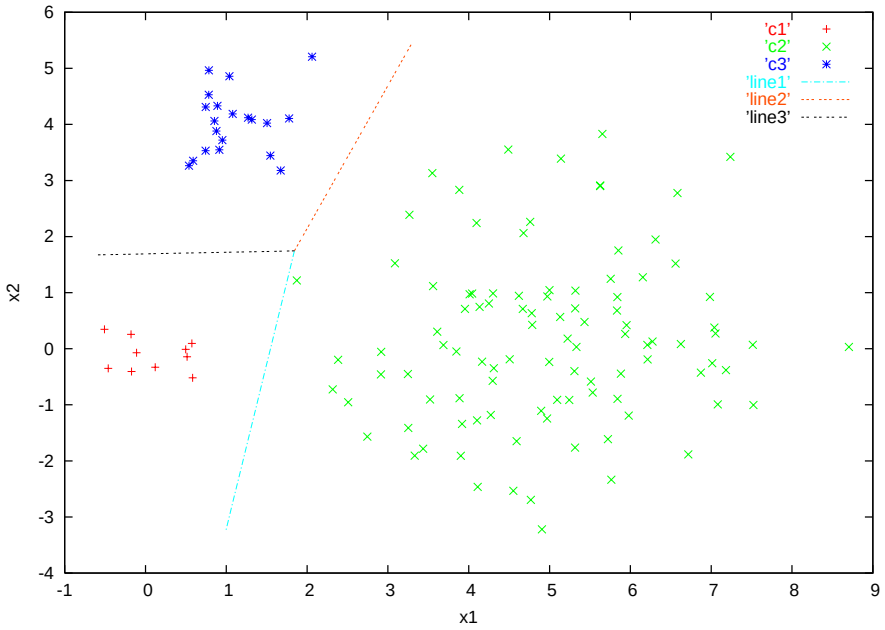

図 5: クラスタリングの最終結果

5.2 LVQ によるクラスタリングと K-Means アルゴリズムを用いた分割・併合

ここまで K-Means アルゴリズムによる分割・併合の手順を用いたクラスタリングを扱ったが，LVQ などその他のアルゴリズムを本手法に適用することも可能である。つまり，決定領域を求める時，また決定領域を部分領域に分割する際に，異なるアルゴリズムを用いることもできる。この節では，LVQ を適用して求めた決定領域に対して K-Means アルゴリズムを用いて分割・併合を行った例を示す。

図 2 のデータに対して LVQ を用いてクラスタリングを行うと，図 6 のような分類結果を得た。クラスタリングの結果は表 9 にまとめている。図 6 より，130 個の個体のうち 11 個体が誤り分類であることがわかる。前節で述べたように，LVQ を用いて比較的本来のクラスター中心に近いセントロイドが求まったとしても，統計分布の異なるクラスデータをボロノイ分割するとやはり分類誤りが生じる。したがって，LVQ によるクラスタリングで得られた分類結果に対しても分割・併合を行うことによって正しく分類する必要がある。

ここで注意すべき点として，通常 LVQ を用いてクラスタリングを行った場合，その分類結果はクラスター中心を母点としたボロノイ分割ではない。よって，領域を分ける境界線は直線で描くことができない。境界を求める手法として最近傍識別 [12] などがあるが，ここでは境界線の描画は行っていない。

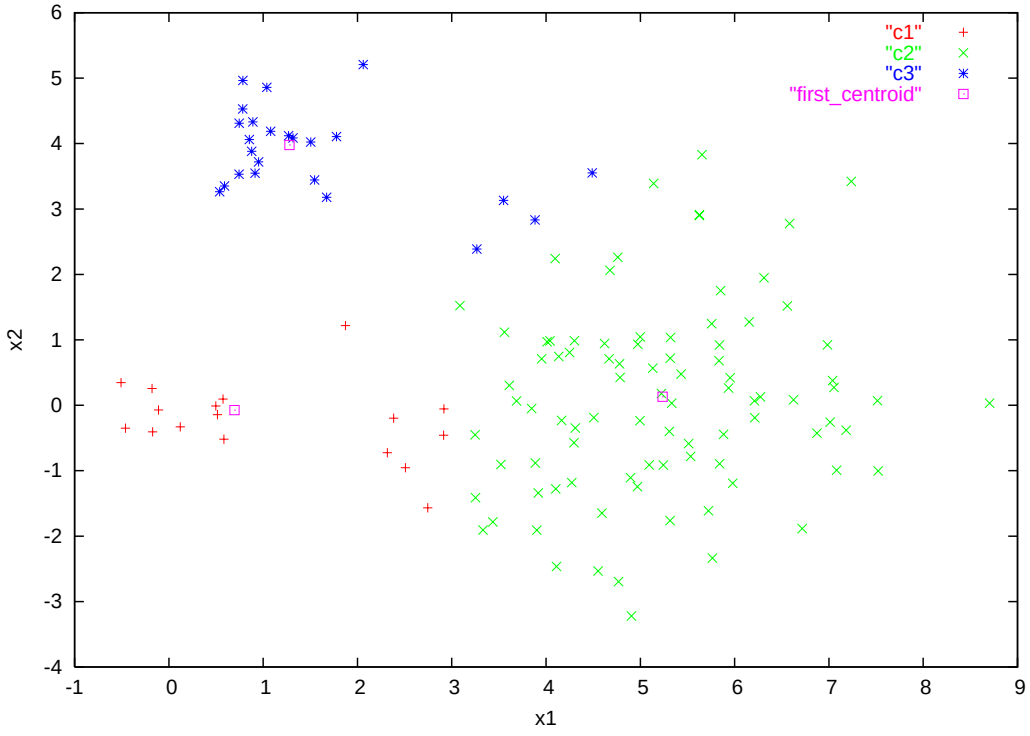

図 6: $K = 3$ での LVQ によるクラスタリング

表 9: LVQ によるクラスタリング結果と決定領域

決定領域	データ数	クラスター中心
R_1	17	$\vec{c}_1 = (0.695, -0.073)$
R_2	89	$\vec{c}_2 = (5.23, 0.131)$
R_3	24	$\vec{c}_3 = (1.27, 3.97)$

表 10: $m = 2$ での LVQ によるクラスタリング

部分領域	データ数	クラスター中心
$R_{1,1}^{(m=2)}$	10	$\vec{c}_{1,1}^{(m=2)} = (0.0861, -0.113)$
$R_{1,2}^{(m=2)}$	7	$\vec{c}_{1,2}^{(m=2)} = (2.52, -0.391)$
$R_{2,1}^{(m=2)}$	50	$\vec{c}_{2,1}^{(m=2)} = (4.54, -0.662)$
$R_{2,2}^{(m=2)}$	39	$\vec{c}_{2,2}^{(m=2)} = (6.09, 1.03)$
$R_{3,1}^{(m=2)}$	20	$\vec{c}_{3,1}^{(m=2)} = (1.10, 4.04)$
$R_{3,2}^{(m=2)}$	4	$\vec{c}_{3,2}^{(m=2)} = (3.80, 2.98)$

表 11: $\{R_k\}$, $\{R_k^{(m)}\}$ における歪み $\{D_k^{(m)}\}$

	$D_k^{(m=1)}$	$D_k^{(m=2)}$	$D_k^{(m=3)}$
$k = 1$	35.63	7.82	5.31
$k = 2$	306.8	190.9	123.41
$k = 3$	41.17	11.07	8.64

表 12: 分割の尺度 $\{\rho_k^{(m)}\}$

	$\rho_k^{(m=2)}$	$\rho_k^{(m=3)}$
$k = 1$	0.220	0.681
$k = 2$	0.627	0.653
$k = 3$	0.271	0.786

部分領域の数 M=3 を最大値として選び，K=m で LVQ によって得られた決定領域に，K-Means アルゴリズムを適用することによって，それぞれの決定領域を m 個の部分領域に分割する。$m = 2$ とした時の，決定領域 $\{R_k\}$ の K-Means アルゴリズムによるクラスタリング結果を表 10 に示す。また，決定領域 $\{R_k\}$ と部分領域 $\{R_k^{(m)}\}$ における歪み $\{D_k^{(m)}\}$ を表 11 に示す。表 12 は，式 (13) によって与えられる分割の尺度を表わしている。前節と同様に $\eta \approx 0.4$ と設定した時，この表の $\{\rho_k^{(m)}\}$ の値から m^*=2 が最適であり，決定領域 R_1 と R_3 は，それぞれ 2 つの部分領域に分割されるべきであるとわかる。この状況は図 7 で表わされる。

次に，適切な部分領域を，隣接する決定領域に併合する。前に述べたように，LVQ によるクラスタリングでは直線の境界線が求められないことから，併合のための距離尺度として，部分領域のクラスター中心から隣接する決定領域の境界線までの距離を用いることはできない。したがって，式 (17) で定義される部分領域と隣接する決定領域に属するデータ間の距離を利用することになる。これを表 13 にまとめている。この表から $R_{1,2}^{(m^*=2)}$ は R_2 に併合され，$R_{3,2}^{(m^*=2)}$ は R_2 に併合されるべきであることがわかる。

LVQ によるクラスタリングで得た決定領域を K-Means アルゴリズムを用いて分割・併合を行った時のクラスタリングの最終結果を図 8 に示す。前節と同様，正しい分類結果が得られたことが示されている。

表 13: 部分領域と決定領域のデータ間の距離とその最短距離 $\{d(R_{k,p}^{(m^*=2)})\}$

部分領域	R_1	R_2	R_3	$d(R_{k,p}^{(m^*)})$
$R_{1,1}^{(m^*=2)}$		2.66	3.09	$d(R_{1,1}^{(m^*=2)}) = 2.66$
$R_{1,2}^{(m^*=2)}$		0.333	1.82	$d(R_{1,2}^{(m^*=2)}) = 0.333$
$R_{3,1}^{(m^*=2)}$	1.97	2.17		$d(R_{3,1}^{(m^*=2)}) = 1.97$
$R_{3,2}^{(m^*=2)}$	1.82	0.628		$d(R_{3,2}^{(m^*=2)}) = 0.628$

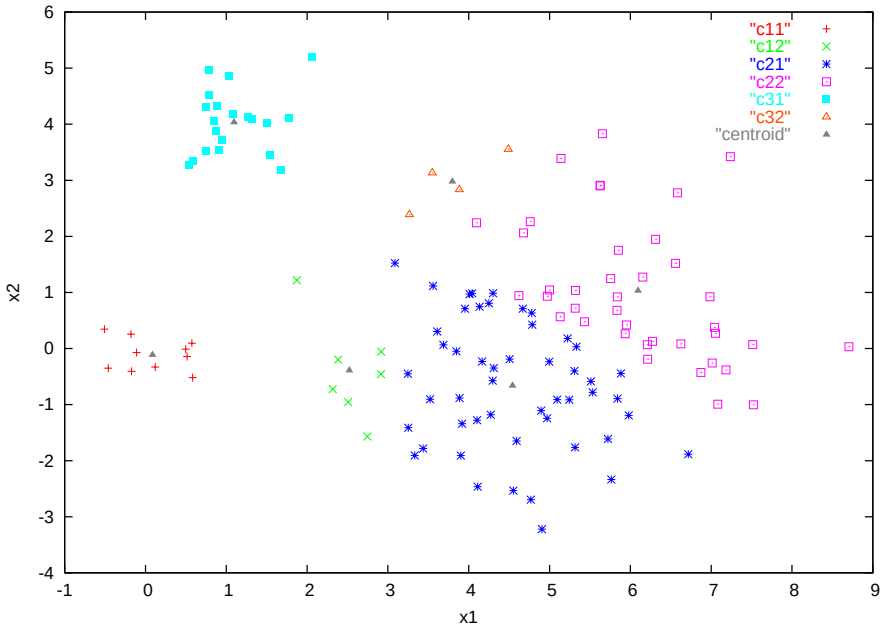

図 7: K=3,m^*=2 の LVQ によるクラスタリングと K-Means アルゴリズムによる分割・併合

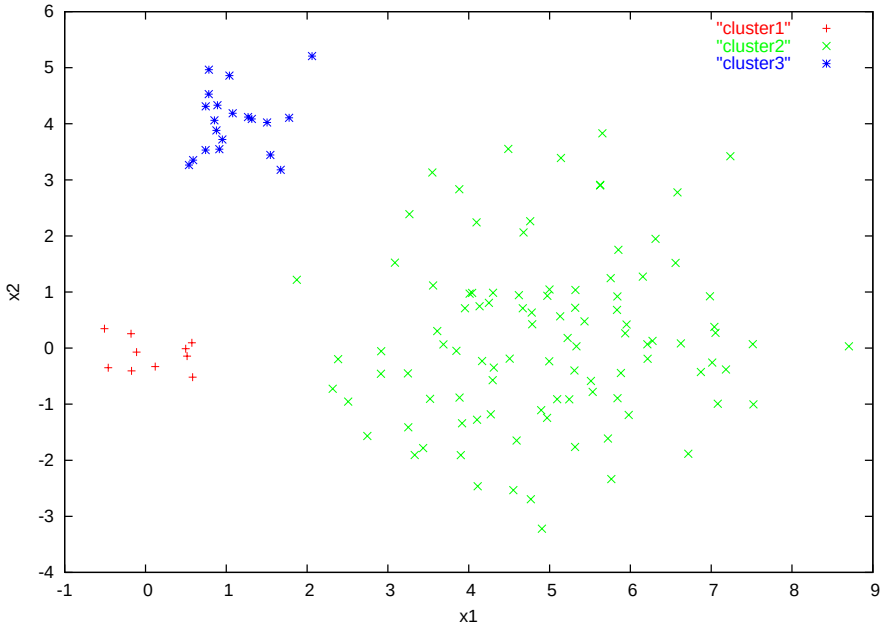

図 8: LVQ と K-Means アルゴリズムを用いた分割・併合によるクラスタリングの最終結果

5.3 カーネル関数を用いた K-Means アルゴリズムによるクラスタリング

図 9 に示すような 2 つのクラスからなるデータについて実験を行う。円 (ball) の周りに輪 (ring) があり，それぞれのクラスは 200 個の個体からなっている。また，カーネル関数を用いない従来の K-Means アルゴリズムによる分類結果を図 10 に，カーネル関数を用いた K-Means アルゴリズムによる分類結果を図 11 に示す。これらの図から，線形分離ができないデータに対して，カーネル関数を用いることで分類可能になるということがわかる。本研究で用いるカーネル関数は，定数 $C = 0.1$ の RBF カーネルである。

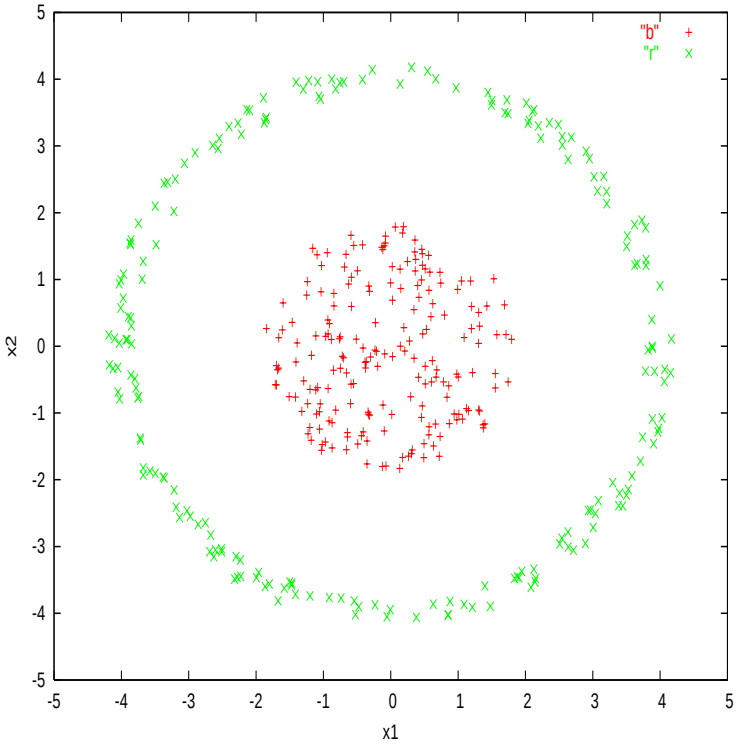

図 9: 2 つのクラスから構成される非線形データ

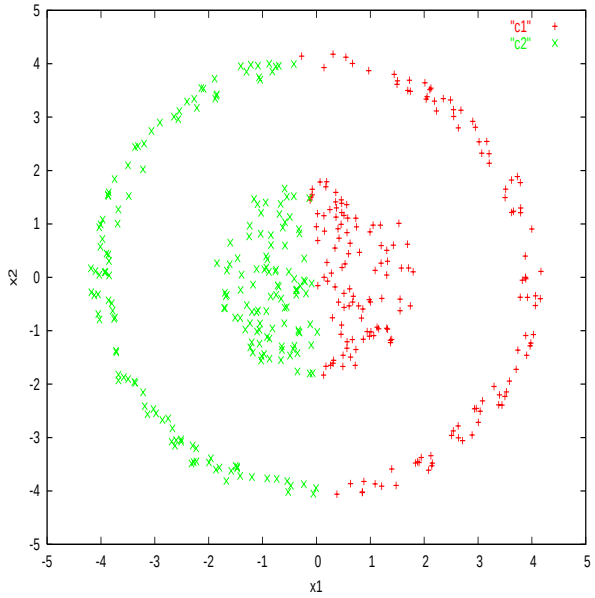

図 10: K-Means アルゴリズムによる分類結果

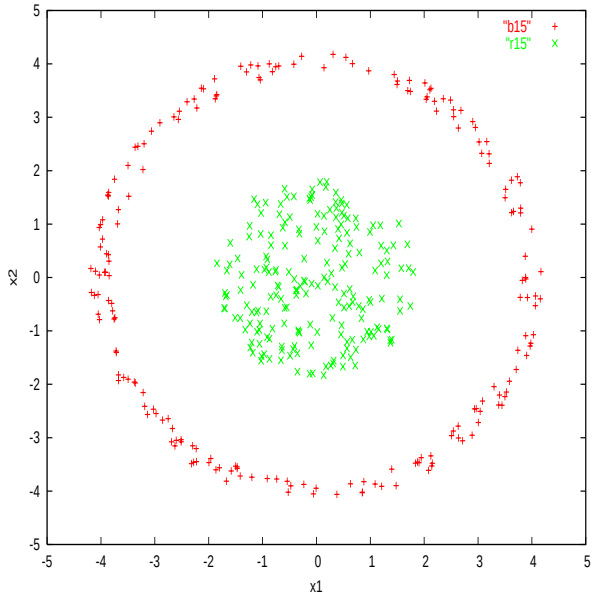

図 11: カーネル関数を用いた K-Means アルゴリズムによる分類結果

次に，400 個の個体からランダムに 2 つの個体を選び各クラスターの初期値に設定する。この操作を 100 回繰り返したときのクラスタリング結果について，完全に分類できた例と分類誤りを含む例を図 13-図 18 に示す。

これらの結果について考察すると，各クラスターにおける初期値の選び方が分類結果に大きく影響していることがわかる。２つのクラスターの初期値の与え方として，内側の円の上に２つとも初期値がある場合，内側の円と外側の輪の上に１つずつ初期値がある場合，外側の輪の上に両方ある場合の 3 つのパターンをそれぞれ図 13,14，図 15,16，図 17,18 に示している。同じような位置に初期値を与えたとしても，その分類結果は大きく異なっており，２つの初期値が両方とも輪の上にある場合は，正しく分類されることはなかった。高次元空間上でのデータの位置関係が明らかではないことから，初期値と分類結果の関係性は一概には言えない。一般には，高次元特徴空間に写像することで各クラスターが概念的に図 12 のような位置関係になることで，線形分離が可能になると考えられているが，高次元上での詳しい分布はわかっていない。高次元特徴空間上でのデータの分布を明らかにする必要があると考えられる。

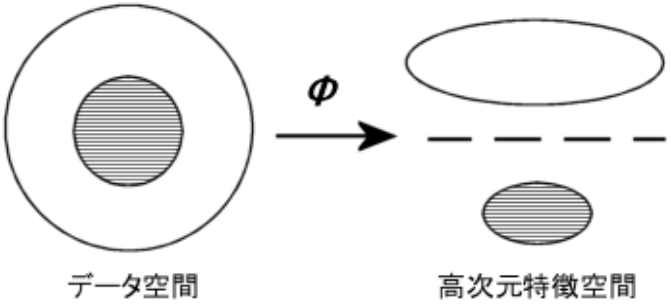

図 12: 高次元特徴空間上での概念的なクラスター位置

カーネル関数を用いた K-Means アルゴリズム (Kkm) とカーネル関数を用いた変形 K-Means アルゴリズム (Kkm') に対して，それぞれ初期値をランダムに与える操作を 100 回繰り返したときのクラスタリング結果において，各アルゴリズムにおける成功率，平均繰り返し回数を表 14 にまとめる。この 2 つのアルゴリズムは目的関数は同じであるが，その最適化の手法の違いが数値に反映していると考えられる。

表 14: Kkm,Kkm' によるクラスタリング結果

method	成功率	繰り返し回数
Kkm	0.21	13.23
Kkm'	0.24	6.94

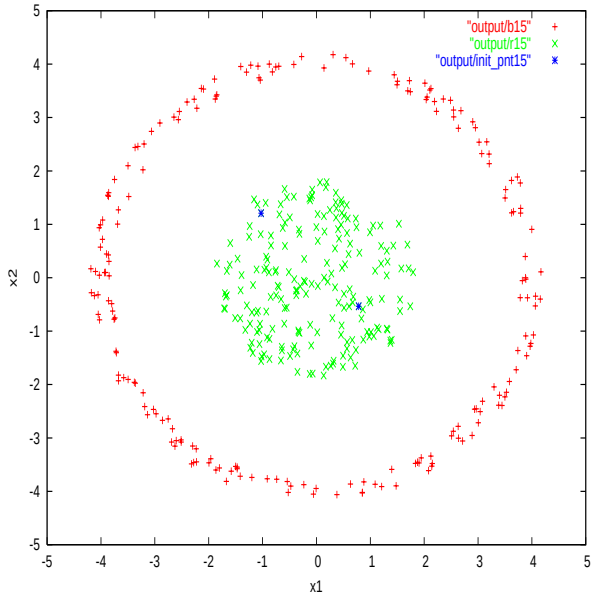

図 13: 分類成功例：Kkm, 初期値 (-1.027533,1.208392),(0.784579,-0.534775)

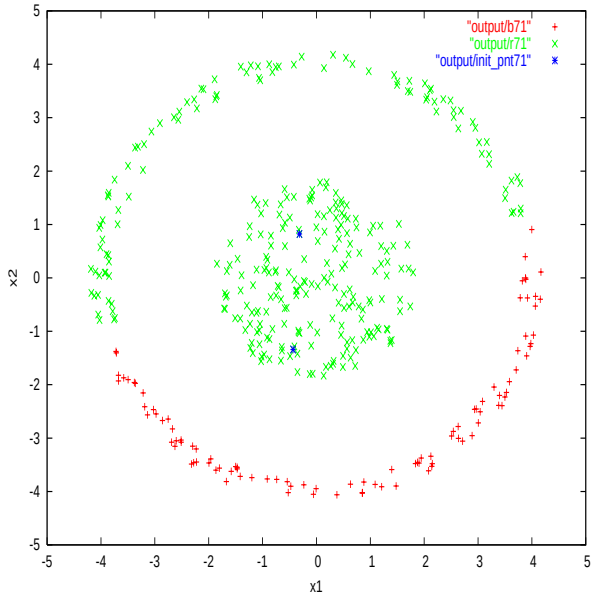

図 14: 分類誤り例：Kkm, 初期値 (-0.433101,-1.341578),(-0.314275,0.821440)

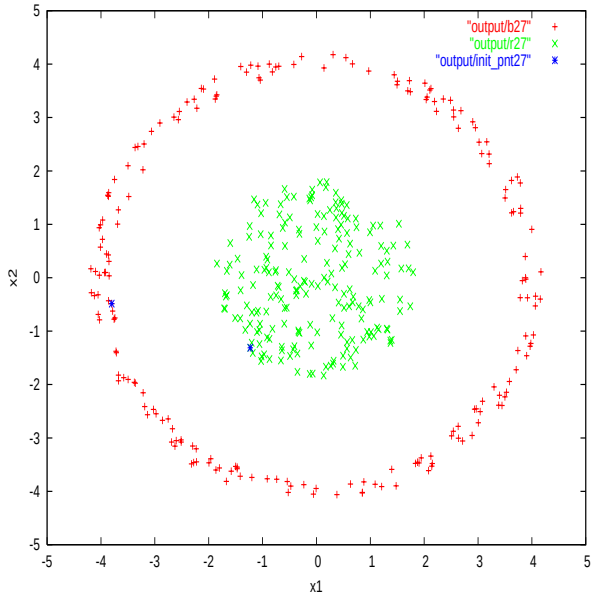

図 15: 分類成功例：Km, 初期値 (-1.228715,-1.312461),(-3.800445,-0.485205)

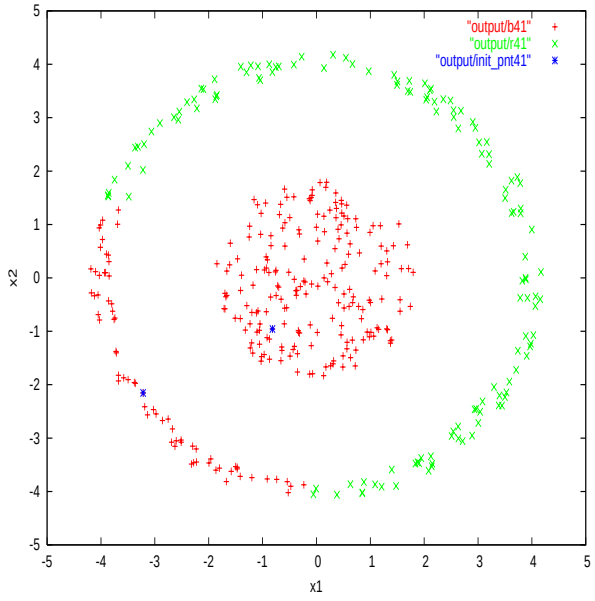

図 16: 分類誤り例：Km, 初期値 (-3.217651,-2.154126),(-0.816030,-0.957352)

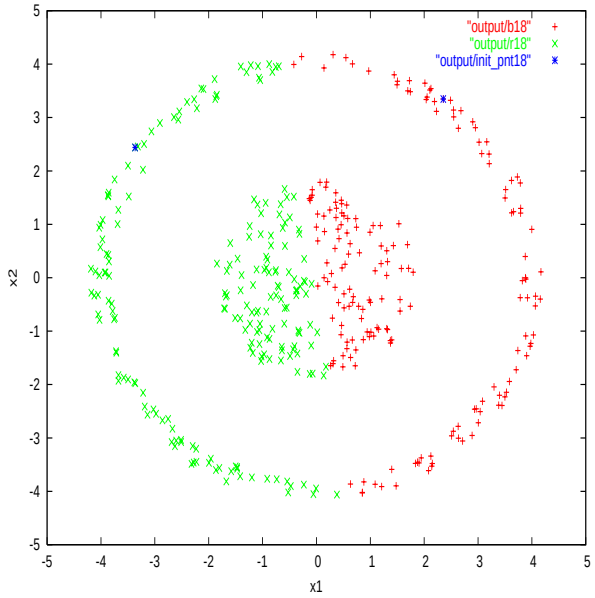

図 17: 分類誤り例：Km, 初期値 (2.355734,3.346941),(-3.366276,2.438362)

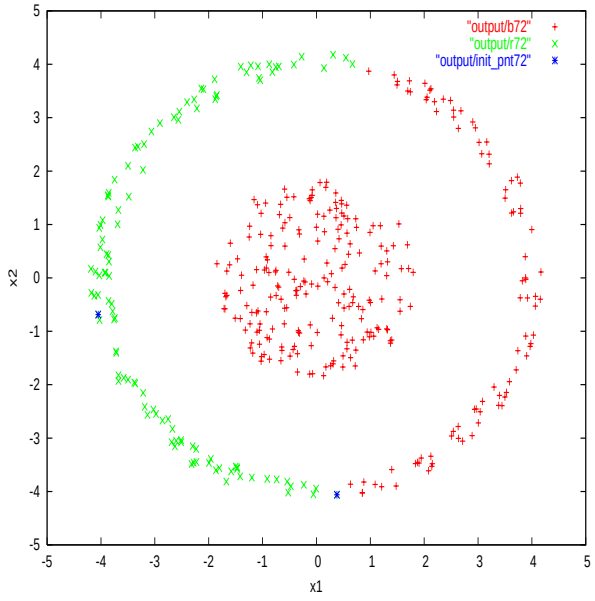

図 18: 分類誤り例：Km, 初期値 (0.380646,-4.061833),(-4.050100,-0.687073)

6　考察

本論文では，統計的に異なる分布を有するクラスのデータに対して，従来の K-Means アルゴリズムによるクラスタリングでは誤りを含む分類結果になることを示し，分割・併合機能を有する K-Means アルゴリズムを適用することにより分類の改善をはかった。しかし，K-Means アルゴリズムや LVQ などを用いたクラスタリングは，それぞれのクラスの統計分布に偏りがなく，正しく分類することが可能なデータに対しても，クラスター中心の初期値の選び方によっては，誤りのある分類を得る可能性がある。このような場合についても，提案手法にもとづいて得られた決定領域を分割・併合することにより正しく分類できる可能性を有する。

カーネル法を用いたクラスタリングにおいて，本論文では K-Mean アルゴリズムを用いて実験を行ったが，LVQ への拡張も可能であると考えており，カーネル関数を用いた LVQ による分類実験も行う予定である。また，カーネル関数の性質やパラメータの決定の仕方についての考察や，実データに対しての適用も必要であると考えている。

7　まとめ

従来の K-Means アルゴリズムによる分類性能を向上させるために，分割・併合機能を有する K-Means アルゴリズムによる，新たなクラスタリング手法を提案した。分割の尺度と併合のための距離尺度を導入し，従来の K-Means アルゴリズムでの典型的な分類誤りを示すデータに対し，提案手法にもとづいた分類実験を行った。そして，信頼性のある効果的な分類結果が得られたことを示した。また，カーネル法を用いたクラスタリングでは，非線形なデータに対してカーネル関数を用いた K-Means アルゴリズムを適用し，線形分離可能になるということを示した。将来的な課題として，更に様々なデータに適用することによって，本手法の性能を明らかにしたいと考えている。

謝辞

本研究を通じて，直接ご指導賜りました森井藤樹教授に，深くお礼申し上げます。

参考文献

[1] Duda R.O., Hart P.E., Stork D.G., *Pattern Classification* (2nd Edition), John Wiley & Sons, INC., 2001.

[2] Jain A.K., Dubes R.C., *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[3] Gordon A.D., *Classification* (2nd Edition), Chapman & Hall/CRC, 1999.

[4] Bezdek J.C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, NY, 1981.

[5] MacQueen J., "Some Methods for Classification and Analysis of Multivariate Observations," Proc. 5th Berkeley Symp. on Math. Stat. and Prob. 1, Univ. of California Press, Berkeley and Los Angeles, pp. 281-297, 1967.

[6] Linde Y., Buzo A., Gray R.M., "An Algorithm for Vector Quantizer Design," IEEE Trans. Commun., Vol.28, pp. 84-95, 1980.

[7] 宮本定明, クラスター分析入門:ファジィクラスタリングの理論と応用, 森北出版, 東京, 1999.

[8] Tarsitano A., "A Computational Study of Several Relocation Methods for K-Means Algorithm," Pattern Recognition, Vol.36, pp.2955-2966, 2003.

[9] Yu J., "General C-Means Clustering Model", IEEE Trans. PAMI., Vol.27, No.8, pp.1197-1211, 2005.

[10] Press W.H., Flannery B.P., Teukolsky S.A., Vetterling W.T., *Numerical Recipes in C*, Cambridge University Press, 1988.

[11] T.Kohonen, *Self-Organizing Maps* (2nd Edition), Springer, Berlin, 1997.

[12] 和田俊和, "空間分割による最近傍識別の高速化", 情報処理学会論文誌, Vol.46, pp.912-918, 2005.

[13] M.Girolami, "Mercer kernel based clustering in feature space", IEEE Trans. on Neural Networks, Vol.13, No3, pp.780-784, 2002.

[14] S.Miyamoto, Y.Nakayama, “Algorithms of hard c-means clustering using kernel functions in support vector machines”, J. of Advanced computational Intelligence and Intelligent Informatics, Vol.1.7, No.1, pp.19-24, 2003.

[15] V.Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[16] F.Morii, K.Kurahashi, “Clustering by the K-Means Algorithm Using a Split and Merge Procedure”, Proceedings of SCIS&ISIS, SA-F2-6, pp.1767-1770, 2006.

[17] 倉橋和子, 森井藤樹, “分割・併合機能を有する K-Means アルゴリズムによるクラスタリング”, 電子情報通信学会技術研究報告, PRMU, vol.106, No.470, pp.67-71, 2007.