98-01064

進化型計算メカニズムを用いた大規模データの簡略化に関する研究

石渕　久生　大阪府立大学大学院工学研究科教授

1　研究目的

離散探索空間内での最適化手法である遺伝的アルゴリズムは、組合せ最適化や機械学習など、様々な分野へ適用されている。近年、遺伝的アルゴリズムを用いたパターン選択により大規模データの簡略化を行う方法が提案されている。パターン選択の目的は、高い識別能力を持つ典型的な少数のパターンを選択することである。選択されたパターン集合は、最近傍識別器における参照点集合として用いられる。したがって、パターン選択は、参照点集合の識別能力の最大化と選択されるパターン数の最小化を行う問題として定式化されている。さらに、遺伝的アルゴリズムを用いて、パターン選択と属性選択を同時に行う方法も提案されている[1]。この方法では、参照点集合の識別能力の最大化、選択されるパターン数の最小化および選択される属性数の最小化が遺伝的アルゴリズムを用いて行われる。本研究の目的は、遺伝的アルゴリズムによりパターン選択と属性選択を同時に行う方法の改良および性能評価である[2]。具体的には、参照点集合の識別能力の定義方法、選択された参照点集合の性能評価、選択された参照点集合をニューラルネットワークの学習用データとして用いることの有効性などを調べた。

2　遺伝的アルゴリズムの適用方法

2.1　コード化

遺伝的アルゴリズムは、cクラスn次元パターン識別問題に対して与えられているm個のパターン$\mathbf{x}_p=(x_{p1},\ldots,x_{pn})$, $p=1,2,\ldots,m$ から少数のパターンと属性を選び出すために用いられる。ここで、x_{pi}はp番目のパターンにおける第i属性の属性値を示している。属性とパターンの全体集合を、それぞれ、$F_{\mathrm{ALL}}=\{f_1,\ldots,f_n\}$と$P_{\mathrm{ALL}}=\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$で表すことにしよう。ここで、$f_i$は第$i$属性を表す記号である。また、選択された属性集合とパターン集合をPとFで表すことにする。ここで、$P\subseteq P_{\mathrm{ALL}}$と$F\subseteq F_{\mathrm{ALL}}$である。このとき、最近傍識別器で用いられる参照点集合は、$S=(F, P)$と表すことができる。

本研究で用いる遺伝的アルゴリズムでは、参照点集合$S=(F, P)$は次のような長さ$(n+m)$のビット列として表現される。

$$S = a_1 a_2 \cdots a_n s_1 s_2 \cdots s_m \tag{1}$$

ここで、a_iは第i属性の選択（$a_i=1$）あるいは非選択（$a_i=0$）、s_pはp番目のパターン$\mathbf{x}_p$の選択（$s_p=1$）あるいは非選択（$s_p=0$）を表している。したがって、式(1)に示す長さ$(n+m)$のビット列から属性集合Fとパターン集合Pが次のように得られる。

$$F = \{f_i \mid a_i = 1,\ i = 1,2,\ldots,n\} \tag{2}$$

$$P = \{\mathbf{x}_p \mid s_p = 1,\ p = 1,2,\ldots,m\} \tag{3}$$

2.2　適応度関数

参照点集合$S(F, P)$を持つ最近傍識別器では、未知パターン$\mathbf{x}$の最近傍参照点$\mathbf{x}_p$は、参照点集合から次式により選び出される。

$$d_F(\mathbf{x}_{\hat{p}}, \mathbf{x}) = \min\{d_F(\mathbf{x}_p, \mathbf{x}) \mid \mathbf{x}_p \in P\} \tag{4}$$

ここで、$d_F(\mathbf{x}_p, \mathbf{x})$はパターン$\mathbf{x}_p$とパターン$\mathbf{x}$の間の距離であり、選択されている属性集合$F$を用いて次のように定義される。

$$d_F(\mathbf{x}_p, \mathbf{x}) = \sqrt{\sum_{i \in F} (x_{pi} - x_i)^2} \tag{5}$$

未知パターン$\mathbf{x}$は、その最近傍参照点$\mathbf{x}_p$により識別される。すなわち、未知パターンは最近傍参照点と同じクラスであると識別される。

属性選択とパターン選択では、属性数とパターン数が最小化され、参照点集合の識別能力が最大化される。すなわち、次のような３目的最適化問題として定式化される。

$$\text{Minimize} \quad |F|, \text{minimize} \quad |P|, \text{and maximize} \quad g(S) \tag{6}$$

ここで、$|F|$は属性集合Fに含まれる属性数、$|P|$はパターン集合Pに含まれるパターン数、$g(S)$は参照点集合$S=(F,\ P)$の識別能力の指標である。本研究では、参照点集合の識別能力は、最近傍識別器として参照点集合を用いた場合での識別結果により定義することにする。

最も直接的に参照点集合$S=(F,\ P)$の識別能力を定義する方法は、与えられているm個のパターンの識別を参照点集合により行うことである。すなわち、参照点集合から構成される最近傍識別器を用いて、m個のパターンの識別を行い、正しく識別されたパターン数を参照点集合の識別能力の指標として用いる方法である。このとき、個々のパターン$\mathbf{x}_q$(q=1, 2,..., m)の最近傍参照点$\mathbf{x}_p$は、参照点集合の中から次式により選択される。

$$d_F(\mathbf{x}_{\hat{p}}, \mathbf{x}_q) = \min\{d_F(\mathbf{x}_p, \mathbf{x}_q) | \mathbf{x}_p \in P\} \tag{7}$$

式(7)に基づく最近傍識別により正しく識別されるパターン数を$g_A(S)$とする。従来の多くの研究におけるパターン選択問題は、$g_A(S)=m$という制約のもとで$|P|$を最小化する問題として定式化されている。すなわち、与えられたm個のパターンを正しく識別することのできる最小のパターン集合を求める問題として定式化されている。

一方、参照点集合の汎化能力の最大化を目的とした研究では、識別能力の定義に工夫が試みられている。最も良く用いられる方法は、個々のパターンの識別を行う場合に、そのパターンを最近傍参照点として選び出さないという方法である。すなわち、個々のパターンのパターン識別は、常に他のパターンにより行われることになる。具体的には、パターン$\mathbf{x}_q$の最近傍参照点$\mathbf{x}_p$は、次式により選択されることになる。

$$d_F(\mathbf{x}_{\hat{p}}, \mathbf{x}_q) = \begin{cases} \min\{d_F(\mathbf{x}_p, \mathbf{x}_q) | \mathbf{x}_p \in P\}, & \text{if} \quad \mathbf{x}_q \notin P \\ \min\{d_F(\mathbf{x}_p, \mathbf{x}_q) | \mathbf{x}_p \in P - \{\mathbf{x}_q\}\}, & \text{if} \quad \mathbf{x}_q \in P \end{cases} \tag{8}$$

式(8)に基づく最近傍識別により正しく識別されるパターン数を$g_B(S)$とする。

これらの二つの定義は、参照点集合に含まれるパターンの識別だけが異なり、参照点集合に含まれていないパターンの識別は全く同じである。さらに、参照点集合に含まれるパターン数すなわち遺伝的アルゴリズムにより選択されるパターン数は非常に少ないので、二つの定義の違いも小さいと考えられる。しかし、実際には、どちらの定義を用いるかで選択される参照点集合は大きく異なる結果となった。

参照点集合$S=(F,\ P)$の適応度は、式(6)で定式化した３目的最適化問題に含まれる三つの目的関数の加重和として、次式のように設定される。

$$fitness(S) = W_g \cdot g(S) \ - \ W_F \cdot |F| \ - \ W_P \cdot |P| \tag{9}$$

ここで、W_gとW_FおよびW_Pは正の重みである。また、識別性能の指標$g(S)$としては、上述の$g_A(S)$または$g_B(S)$が用いられる。

2.3　基本アルゴリズム

式(9)で定義されている適応度関数の最大化を行うために、遺伝的アルゴリズムを用いることにする。まず、初期個体群として、長さ$(n+m)$のビット列をランダムにN_{pop}個生成する。ここで、N_{pop}は個体群に含まれる個体の数である。次に、選択と交叉および突然変異によりN_{pop}個のビット列を生成する。生成されたビット列は、現在の個体群に含まれるビット列に加えられ、$2N_{pop}$個の個体を含む個体群が形成される。次世代の個体群は、$2N_{pop}$個の個体の中から適応度が高い順に選ばれたm個の個体により構成される。このような世代更新は、事前に設定されている終了基準を満たすまで続けられる。このような遺伝的アルゴリズムは、次のように書くことができる。

Step 1 (初期個体群の生成):

長さ$(n+m)$のビット列をランダムにN_{pop}個生成する。

Step 2 (遺伝操作):

以下の手続きを$N_{pop}/2$回繰り返し、N_{pop}個のビット列を生成する。

1) 現在の個体群からランダムに２個のビット列を選択する。
2) 選択された２個のビット列に交叉操作を適用する。数値実験では、一様交叉を用いた。
3) 交叉操作により生成されたビット列に突然変異操作を適用する。

Step 3 (世代更新):

新たに生成されたN_{pop}個のビット列を現在の個体群に加え、$2N_{pop}$個のビット列を含む個体群を構成する。この中から高い適応度を持つビット列から順にN_{pop}個を選び出し、次世代の個体群とする。

Step 4 (終了判定):

事前に設定されている終了条件が満たされていなければStep 2へ戻る。数値実験では、アルゴリズムの繰り返し数（すなわち世代数）を終了条件として用いた。

2.4 数値計算例を用いた説明

図１に示す簡単な例題を用いて本研究における属性とパターンの同時選択手法を説明する。図１は各クラスから30個のパターンが得られている２クラス２次元パターン識別問題であり、識別境界線は全てのパターンを用いた最近傍識別器による識別を表している。この例題に対して、以下のようなパラメータを用いて、前節の遺伝的アルゴリズムを適用した。

ビット列の長さ：62（２個の属性と60個のパターン）

個体群のサイズ：N_{pop}=50

交叉確率：1.0

突然変異確率：0.01

終了条件：1000世代

重みの値：W_g=10; W_F= 1; W_P=1

識別能力の指標：$g_A(S)$

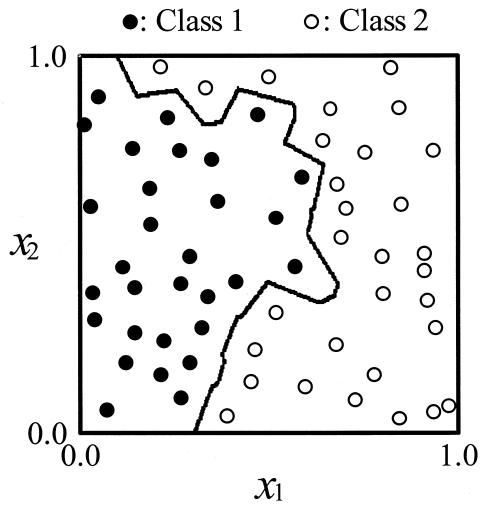

図1　与えられたパターン

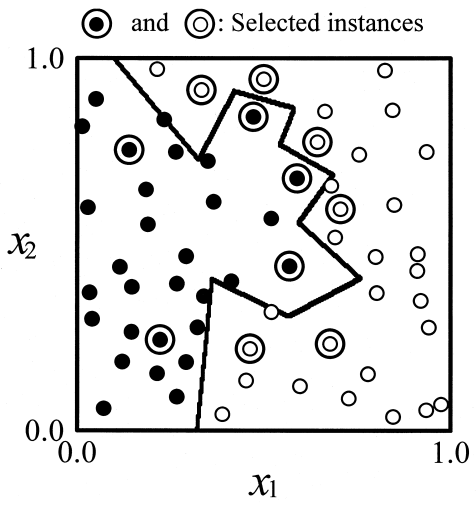

図2　選択されたパターン

遺伝的アルゴリズムは図２に示す11個のパターンを選択した。図２より、与えられた全てのパターンが正しく識別されていることが分かる。これは、学習用パターンに対する識別能力を測る指標である$g_A(S)$に対して大きな重み（W_g=10）を与えているからである。

汎化能力を推定する$g_B(S)$を用いる場合や重みの値が小さい場合は、常に100%の正答率が得られるわけではない。参照点集合の識別能力を$g_B(S)$で定義した場合での遺伝的アルゴリズムにより選択された９個のパターンを図３に示す。図３では、１個のパターンが誤識別されている。また、$g_A(S)$に対する重みを0.5に設定した場合での数値実験結果を図４に示す。図４では、３個のパターンが参照点集合として選択され、参照点集合によるパターン識別の結果として３個のパターンが誤識別されている。この場合では、正答パターン数に関する重み（W_g=0.5）が選択されるパターン数に関する重み（W_P=1）よりも小さいため、誤識別されるパターン数が増えても選択されるパターン数を減らす方が、適応度関数の値が大きくなる。このように、３種類の重みの設定値に応じて、様々な参照点集合が得られる。各評価基準の重みを自由に設定できることは、遺伝的アルゴリズムに基づく手法の利点の一つである。

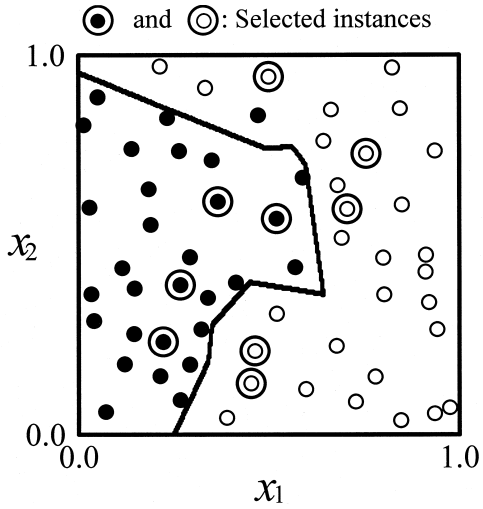

図3　$g_B(S)$を用いた場合の結果

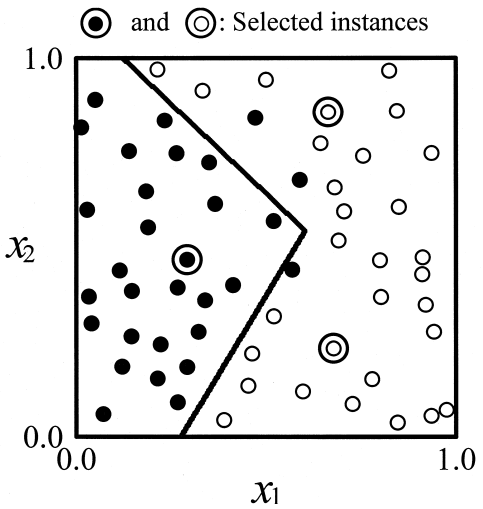

図4　W_g=0.5と設定した場合の結果

2.5　方向性を持つ突然変異確率の導入

遺伝的アルゴリズムを用いた参照点集合の探索では、参照点集合は長さ($n+m$)のビット列で表現される。したがって、探索空間のサイズすなわち探索空間に含まれるビット列の総数は、2^{n+m}個となる。このサイズは、大規模データに適用する場合では、非常に大きなものとなる。通常のパターン識別問題では、パターン数が属性数よりも明らかに多いので、ここでは、パターン数を効率的に減らす方法を議論する。

遺伝的アルゴリズムで新しいビット列を生成する操作は、交叉操作と突然変異操作である。交叉操作は、親個体に含まれているビットの値を交換するだけであるので、選択されるパターン数の総数に変化は生じない。一方、突然変異操作は、選択されるパターン数が減ることを妨げる効果がある。このことを簡単な計算を用いて示すことにする。いま、突然変異前のビット列に含まれているパターン数をm_1、含まれていないパターン数をm_0とする（$m_0+m_1=m$)。突然変異操作により、m_1個のパターンの中で平均して$p_m \cdot m_1$個のパターンが記号列から取り除かれ、m_0個のパターンの中で平均して$p_m \cdot m_0$個のパターンが記号列に加えられる。ここで、p_mは突然変異確率である。したがって、突然変異後の記号列に含まれるパターン数の期待値は、次のように計算される。

$$\hat{m}_1 = m_1 - p_m \cdot m_1 + p_m \cdot m_0 . \tag{10}$$

少数のパターンが大規模データから選択される場合では、m_1の値はm_0やmよりもずっと小さなものとなる。例えば、1000個のパターンから10個のパターンが選択されているような状況を考えて見よう（すなわち、m=1000、m_1=10、m_0=990)。この場合では、突然変異後のビット列に含まれるパターン数の期待値は次のように計算される。

P_m=0.1のとき、$\hat{m}_1$=108

P_m=0.01のとき、$\hat{m}_1$=19.8

P_m=0.001のとき、$\hat{m}_1$=10.98

このような計算により、大きな突然変異確率は、選択されるパターン数の減少を妨げることが分かる。突然変異確率と選択されるパターン数との関係を示すために、1000個のパターンを持つ２クラス２次元パターン識別問題に遺伝的アルゴリズムを適用した。各クラス500個のパターンは、正規分布$N(\mu_k, \Sigma_k)$を用いて生成した。ここで、kはクラスを表す記号であり、平均値ベクトルμ_kと共分散行列Σ_kは次のように設定した。

$$\mu_1 = (0,1),\ \mu_2 = (1,0),\ \Sigma_1 = \Sigma_2 = \begin{pmatrix} 0.3^2 & 0 \\ 0 & 0.3^2 \end{pmatrix} \tag{11}$$

パターン選択に関連した突然変異確率として、上述の３種類の値（すなわち、0.1と0.01および0.001）を用いて数値実験を行なった。これらの突然変異確率は、長さ1002のビット列の後半の1000個のビットに適用される。属性選択に関連した最初の２ビットに対する突然変異確率も含めて、他のパラメータに関しては図２の数値実験と同じ値を用いた。各突然変異確率に対して10回の数値実験を行なった結果を表１に示す。なお、計算時間はPentium II 400MHzのプロセッサを持つパソコンで測られたものである。この表から、高い突然変異確率を用いた場合では、選択されるパターン数を減らせていないことが分かる。また、選択されるパターン数が多ければ多いほど、長い計算時間か必要であることも分かる。これは、参照点集合に含まれるパターン数が多いほど、パターン識別に時間が必要となるからである。

表1　10回の数値実験結果の平均

突然変異確率	正答パターン数	選択パターン数	計算時間（分）
0.1	1000	349	317
0.01	1000	32	110
0.001	999	18	86

選択されるパターン数を効率的に減少させるために、本研究では、突然変異確率に方向性を与えることにする。すなわち、パターン選択に関連したビットに対しては、0から1への突然変異よりも1から0への突然変異に大きな確率を与えることにする。なお、属性選択に関連したビットに対しては、通常の突然変異確率P_mを用いた。表1と同じ問題に突然変異確率に方向性を与えた遺伝的アルゴリズムを適用した。具体的には、パターン選択に関する突然変異確率を$P_m(1\to0)=0.1$および$P_m(0\to1)=0.001$と設定し、属性選択に関する突然変異確率は$P_m=0.1$とした。10回の数値実験により、表2のような結果が得られた。表1に示す通常の突然変異確率を用いた場合と比較すると、選択されるパターン数が半分以下になっていることが分かる。

表2　突然変異確率に方向性を与えた場合での数値実験結果

正答パターン数	選択パターン数	計算時間（分）
997	7	51

3　性能評価

3.1　実験に用いたデータ集合

人工的に生成された2種類のパターン識別問題と4種類の実データを用いた。数値実験では、属性値を単位区間[0,1]内の実数値に変換した後で、遺伝的アルゴリズムを個々のデータ集合に適用した。数値実験で用いたデータ集合を簡単に説明すると以下のようになる。

1) 人工データI

正規分布を用いて、2次元パターン空間[0,1]×[0,1]内に2クラス問題を生成した。各クラスに対して50個のパターンを生成するために、正規分布$N(\mu_k, \Sigma_k)$の中心ベクトルμ_kと共分散行列Σ_kを次のように設定した。

$$\mu_1=(0,1),\ \mu_2=(1,0),\ \Sigma_1=\Sigma_2=\begin{pmatrix}0.4^2 & 0\\ 0 & 0.4^2\end{pmatrix}. \tag{12}$$

2) 人工データII

人工データと同様に正規分布を用いて2クラス問題を生成した。データIと比較するとデータIIでは正規分布の分散が大きいため、クラス間の重なりが大きくなっている。具体的には、正規分布$N(\mu_k, \Sigma_k)$の中心ベクトルμ_kと共分散行列Σ_kを次のように設定した。

$$\mu_1=(0,1),\ \mu_2=(1,0),\ \Sigma_1=\Sigma_2=\begin{pmatrix}0.6^2 & 0\\ 0 & 0.6^2\end{pmatrix}. \tag{13}$$

3) アヤメデータ

アヤメデータは、パターン識別問題として最も良く用いられているものである。4次元3クラス問題であるアヤメデータでは、各クラスに50個のパターンが与えられている。クラス1（Iris setosa）は、他の二つのクラスから完全に分離している。他の二つのクラス（Iris versicolorとIris virginica）の間には小さな重なりがある。

4) 盲腸データ

盲腸データは、106人の患者に対して与えられている8種類の属性値を用いて、各患者が盲腸であるかどうかを診断する問題である。オリジナルのデータでは8種類の属性を持っているが、一つの属性は欠落値を持つため、本研究では7次元2クラス問題として盲腸データを取り扱った。

5) 癌データ

癌データは、286人の患者に対して与えられている9種類の属性値を用いて、各患者が癌であるかどうかを診断する問題である。9次元2クラス問題として取り扱われる癌データは、クラス間の重なりが大きく、正答率が80%を超える識別器

を構築することは困難である。

6) ワインデータ

ワインデータは178個のサンプルに対して与えられている13種類の属性値を用いて、各サンプルを３種類のワインに分類する問題である。高次元パターン識別問題ではあるが、クラス間の重なりが小さいため、高い識別能力を持つ識別器を構築することができる。

3.2 学習用データに対する性能

まず、学習用データに対する性能評価を行った結果を示す。本節の数値実験では、各データ集合に含まれる全てのサンプルを学習用データとして用い、次のようなパラメータ設定で遺伝的アルゴリズムを適用した。

個体群のサイズN_{pop}：50

交叉確率：1.0

突然変異確率：p_m=0.1, $p_m(1\rightarrow0)$=0.1, $p_m(0\rightarrow1)$=0.01

終了条件：500世代

重みの値：W_g=5; W_F=1; W_P=1

識別能力の指標として学習用データに対する正答数である$g_A(S)$を用いた場合での数値実験結果を表３に示す。なお、表３は30回の数値実験の平均であり、括弧内の値は選択前の属性数およびパターン数である。また、汎化能力を推定するための指標である$g_B(S)$を用いた場合での数値実験結果を表４に示す。表３と同様に表４も30回の数値実験の平均である。これらの表から、少数のパターンが遺伝的アルゴリズムにより選択されていることが分かる。また、２種類の人工データに対する数値実験結果の比較から、クラス間の重なりが小さいほど選択されるパターン数も少ないことが分かる。このことは、クラス間の重なりが小さいワインデータと重なりが大きい癌データに対する数値実験結果の比較からも明らかである。一方、属性選択に関しては、識別に不必要な属性は取り除かれているようである。例えば、アヤメデータでは$\{f_3, f_4\}$が重要な属性として知られているが、この組合せは、表３では30回中29回の数値実験で選択されている。表４でも、$\{f_3, f_4\}$は16回の数値実験で選択され、残りの14回の数値実験では、$\{f_2, f_3, f_4\}$が選択されている。

表３と表４との比較からは、表３の方が選択されるパターン数と正答パターン数が共に多いことが分かる。これは、$g_A(S)$が学習用データに対する性能を直接的に評価しているためである。特に、クラス間の重なりが大きい人工データIIや癌データでは、非常に多数のパターンが表３では選択されている。一方、クラス間の重なりが小さいワインデータでは、表３と表４の違いは小さい。これらの数値実験から、クラス間の重なりが大きいほど、識別性能を評価する指標の選択が最終的に得られる参照点集合に大きな影響を与えていることが分かる。

表3　学習用データに対する結果（$g_A(S)$を用いた場合）

データ集合	属性数	パターン数	正答率
人工データ I	1.9 (2)	14.5 (100)	96.7%
人工データ II	1.8 (2)	31.0 (100)	94.4%
アヤメ	2.0 (4)	6.1 (150)	99.4%
盲腸	3.3 (7)	16.0 (106)	97.5%
癌	5.1 (9)	54.3 (286)	89.2%
ワイン	6.3 (13)	5.9 (178)	100%

表4　学習用データに対する結果（$g_B(S)$を用いた場合）

データ集合	属性数	パターン数	正答率
人工データ I	2.0 (2)	6.2 (100)	92.3%
人工データ II	1.8 (2)	12.3 (100)	80.9%
アヤメ	2.6 (4)	7.6 (150)	94.2%
盲腸	3.2 (7)	4.4 (106)	91.8%
癌	2.9 (9)	27.2 (286)	81.3%
ワイン	6.6 (13)	7.3 (178)	99.9%

3.3 評価用データに対する性能

本節では、与えられたデータの一部を未知データとして、属性選択とパターン選択には用いないという数値実験を実行することにより、選択された参照点集合の汎化能力を調べた。なお、２種類の人工データに対しては未知データを自由に生成することができるので、100個の学習用データに加えて1000個の未知データを生成することにより数値実験を行った。アヤメデータと盲腸データに対しては、１個のデータのみを未知データとして隠しておくという手法（LV1）を用いた。また、癌データとワインデータに対しては、データ集合を10個の部分集合に等分し、１個の部分集合を未知データとして隠しておくという手法（10CV）を用いた。

前節と同じパラメータを用いて数値実験を行なった結果を表５に示す。比較のため、表５では、属性選択とパターン選択を行う前のデータを用いた場合での最近傍識別器の性能も示している。この表から、汎化能力の指標である$g_B(S)$を用いた場合では、多くのデータ集合に対して、属性選択とパターン選択により識別能力が向上していることが分かる。この向上は、クラス間の重なりが大きい人工データIIや癌データなどで顕著である。一方、学習用データに対する正答数である$g_A(S)$を用いた場合では、アヤメデータと癌データを除いて、識別能力が低下していることも分かる。このような結果から、未知データに対する識別能力の向上を目的とした場合では、$g_B(S)$を用いた属性選択とパターン選択が有効であることが分かる。

表5　評価用データに対する数値実験の結果

データ集合	選択前	$g_A(S)$	$g_B(S)$
人工データⅠ	81.3%	80.2%	84.7%
人工データⅡ	60.6%	60.2%	64.8%
アヤメ	95.3%	96.9%	94.2%
盲腸	80.2%	77.0%	85.7%
癌	65.3%	68.3%	73.6%
ワイン	95.3%	94.8%	96.5%

3.4 属性選択の効果

前節では、属性選択とパターン選択を同時に行うことにより、選択された参照点集合の汎化能力が向上することを明らかにした。本節では、属性選択の効果を調べるために、パターン選択のみを行う遺伝的アルゴリズムを用いた数値実験結果を示す。前節の数値実験と同じ条件で、パターン識別だけを行った参照点集合の汎化能力を調べた。なお、識別性能の評価には、汎化能力を推定する$g_B(S)$を用いた。数値実験結果を表６に示す。表６では、比較のため、パターン選択を行う前の結果および属性選択とパターン選択を同時に行った場合の結果も示している。表６より、属性選択とパターン選択を同時に行った場合と比較すると、パターン選択だけでは汎化能力の向上が少ないことが分かる。

表6　属性選択の必要性を調べた数値実験の結果

データ集合	選択前	パターン選択	同時選択
人工データⅠ	81.3%	84.5%	84.7%
人工データⅡ	60.6%	64.7%	64.8%
アヤメ	95.3%	95.0%	94.2%
盲腸	80.2%	83.2%	85.7%
癌	65.3%	69.6%	73.6%
ワイン	95.3%	95.0%	96.5%

4　ニューラルネットワークの学習への応用

ここでは、選択された参照点集合をニューラルネットワークの学習用データとして用いることの有効性について検討する。ニューラルネットワークの汎化能力には、構造の選択、学習パラメータの選択、学習の終了基準の選択など、様々な要因が関係しているため、学習用データの選択のみの影響を厳密に議論することは難しい。そのため、学習用パターンの選択が学習後のニューラルネットワークの汎化能力を向上させる可能性があることを示唆するだけにする。

汎化能力を調べた3.3節の数値実験で選択された参照点集合を学習用データとして用いてニューラルネットワークの学習を行い、評価用データを用いてニューラルネットワークの汎化能力の評価を行った。数値実験結果の一例として、癌データに対する結果を図５に示す。全ての学習用データを用いた結果（白丸）では、明らかに過学習が観察される。また、識別能力を測る最初の定義である$g_A(S)$を用いて選択された参照点集合を用いた結果（黒丸）では、汎化能力が全体的に低下している。しかし、２番目の定義である$g_B(S)$を用いて選択された参照点集合を用いた場合では、過学習は観察されず、汎化能力も向上している。他のデータ集合に対する数値実験結果でも、最近傍識別器として高い汎化能力を持つ参照点集合を学習用データとして用いた場合では、ニューラルネットワークの汎化能力も向上するという傾向が観察された。

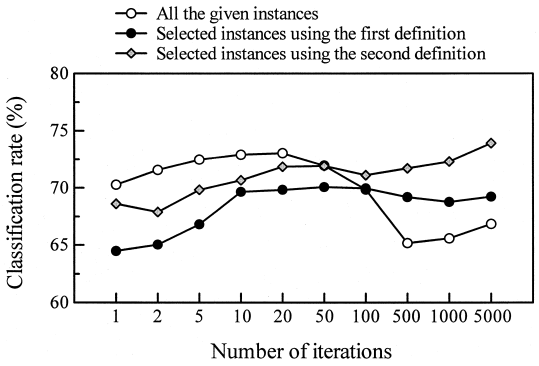

図5　異なる学習用データを用いた場合でのニューラルネットワークの汎化能力の比較

5　まとめ

本研究では、遺伝的アルゴリズムを用いて属性選択とパターン選択を同時に行う方法の改良および性能評価を行った。数値実験により、参照点集合の汎化能力を推定する指標である$g_B(S)$を用いた場合では、遺伝的アルゴリズムにより選択された参照点集合は高い汎化能力を持つという結果が得られた。また、汎化能力の向上は、クラス間の重なりが大きい場合に顕著であるという結果も得られた。さらに、遺伝的アルゴリズムによる学習用データの選択により、ニューラルネットワークの汎化能力を向上させることができるという結果も観察された。

参考文献

[1] H. Ishibuchi and T. Nakashima, “ Evolution of Reference Sets in Nearest Neighbor Classification, ” *Proceedings of the Second Asia-Pacific Conference on Simulated Evolution and Learning* (Canberra, Australia, November 23-26, 1998).

[2] H. Ishibuchi and T. Nakashima, “ Pattern and Feature Selection by Genetic Algorithms in Nearest Neighbor Classification, ” *Journal of Advanced Computational Intelligence* (in press).

＜発　表　資　料＞

題　　名	掲載誌・学会名等	発表年月
Pattern and Feature Selection by Genetic Algorithms in Nearest Neighbor Classification	Journal of Advanced Computational Intelligence	2000年（掲載予定）