並列分散型多目的ファジィ遺伝的機械学習を用いたアンサンブル識別器設計

第 8 グループ　面﨑 祐一

1. はじめに

ファジィ識別器は言語的に解釈可能なルール集合で構成されるため，識別器がどのようにデータを識別しているのかが解釈可能であるという特徴をもつ．しかし，識別性能と解釈性能との間にはトレードオフの関係があるため，どちらも同時に最適となる識別器の獲得は困難である．そこで，ファジィ識別器の設計に進化型多目的最適化手法を用いた多目的ファジィ遺伝的機械学習 (Multiobjective Fuzzy Genetics-Based Machine Learning: MoFGBML) [1] が提案されている．

近年では，あらゆる場所にインターネットが繋がるユビキタス化が進み，大規模なデータが獲得されるようになり，これらの有効利用が期待されている．しかし，MoFGBML は高い識別性能を維持しつつ解釈性に優れた識別器を獲得可能だが，大規模なデータに対して膨大な計算時間を必要とする問題がある．文献 [1] では，学習用データと個体群を分割する島型の並列分散実装を MoFGBML に適用することで計算時間の短縮を実現している．ここでは，部分学習用データへの過学習を防ぐために，部分個体群の移住操作が適用される．その結果，トレードオフ曲線に沿った識別器の獲得が困難であるという課題がある．

本研究では，移住操作は行わない並列分散実装を行う．これにより，各部分個体群は独立に MoFGBML が適用されるため，効率よくトレードオフ曲線に沿った識別器を獲得できる．また，単一の識別器と比較して識別性能の高さが期待されるアンサンブル機構を導入し，各部分個体群から弱識別器を抽出するアンサンブル識別器の設計を行う．

2. 識別性能と解釈性能の 2 目的最適化

2.1. ファジィ識別器

n 次元 M クラスのパターンが m 個与えられたパターン識別問題に対して，ファジィ集合を条件部とする以下の If-then ルールを用いてファジィ識別器を設計する．ある未知パターンは $\mathbf{x} = (x_1, \ldots, x_n)$ のように表され，x_i は第 i 次元 $(i = 1, 2, \ldots, n)$ における属性値を表す．

$$\text{Rule } R\text{: If } x_1 \text{ is } A_1 \text{ and } \ldots \text{ and } x_n \text{ is } A_n \text{ then Class } C \text{ with } CF \qquad (1)$$

$\mathbf{A} = (A_1, \ldots, A_n)$は条件部ファジィ集合，$C$ は結論部クラス，CF はルールの重みを表す．本研究では，条件部ファジィ集合として，2, 3, 4, 5 分割の三角型のファジィ集合 14 種と，メンバシップ値として必ず 1 を返す ”don’t care” の合計 15 種類のファジィ集合を同時に用いる．また，学習用データを用いてルール重みと結論部クラスを決定する．未知パターンの推論は，適合度とルール重みの積が最大となるルールを勝者とする単一勝利ルールによって行う．

2.2. 多目的ファジィ遺伝的機械学習

本研究では，ファジィ識別器の誤識別率の最小化と複雑性 (ルール数) 最小化の 2 つの目的を用いる．代表的な進化型多目的最適化アルゴリズム (Evolutionary Multi-objective Optimization Algorithm: EMOA) である NSGA-II [2] を FGBML に適用し，獲得されるファジィ識別器の 2 目的最適化を図る．以下に MoGBML の手順を示す．

Step 1: 学習用データから初期個体群を生成し，初期個体群の評価を行う．
Step 2: 現個体群から遺伝的操作（交叉，突然変異操作）によって子個体群を生成する．
Step 3: 子個体を評価し，現個体群の世代更新を行う．
Step 4: 終了条件を満たさない場合，Step 2 へ戻る．
Step 5: 得られた個体群から，誤識別率最小化，ルール数最小化の 2 目的において互いに非劣な個体を選択する．

3. 並列分散型 MoFGBML によるアンサンブル識別器

3.1. 並列分散実装

並列分散実装では，学習用データに含まれるパターンの正解クラスの比が変わらないようにして学習用データを分割する．また，個体群サイズを島数で分割し，部分個体群と部分学習用データのペアをそれぞれ一つの CPU コアに割り当てて EMOA を適用する．文献 [1] では，島間の最良個体の移住操作と部分個体群の移住操作が一定間隔で行われる．これらの移住の方向は反対方向で行われる．これにより，部分個体群の部分学習用データへの過学習を防ぎ，得られる識別器の汎化性能を向上させることが期待される．学習終了後，各部分個体群は一つの個体群に統合される．最後に，全ての個体は誤識別率とルール数の 2 目的最小化における非優劣ランキングに基づいて評価される．

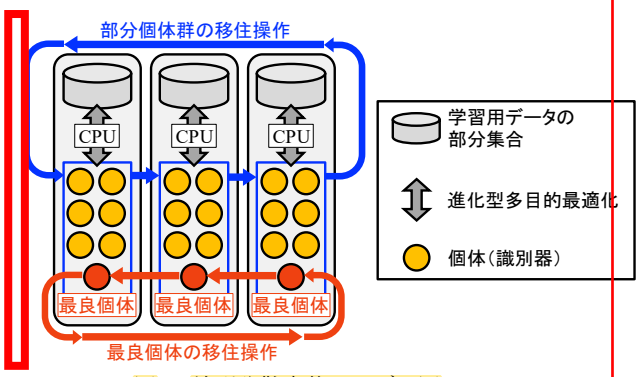


図 1: 並列分散実装のモデル図 [1]

3.2. アンサンブル識別器の設計

単一の識別器には，識別が困難となる特徴を持ったパターンが存在する場合がある．そのようなパターン群に対する単一の識別器の識別結果には分散が生じ，汎化性能が低くなる．そこで，複数の弱識別器を用いたアンサンブル識別器を設計することで，単一の識別器では識別が困難なパターン群への識別結果の分散を減少させることにより，汎化性能の向上が実現される [3]．このとき，識別が困難なパターン群を少なくするため，用いる弱識別器の間には多様性が求められる．

本研究では，弱識別器の多様性向上のため，移住操作を適用しない並列分散型 MoFGBML によるアンサンブル識別器を設計する．移住操作を行わないことで，各島において独立に EMOA が適用されることで各部分個体群の間には高い多様性が期待できる．これにより得られた各部分個体群から最良の識別器を抽出し弱識別器とする．ここでは，弱識別器の多数決によるアンサンブル識別器を設計する．多数決の方法として，全ての弱識別器が 1 票を有する単純多数決と各弱識別器の識別性能 (識別率) を重みとして与えた重み付け多数決を用いる．多数票が同票となった場合，多数票の中からランダムに選択し，識別結果とする．

4. 数値実験

4.1. 数値実験設定

数値実験では，各部分個体群から単一な弱識別器を抽出する場合と，各部分個体群から非劣な弱識別器集合を抽出する場合の 2 種類のアンサンブル識別器を用いる．それぞれに対して，移住操作を適用しない場合 (提案手法) と適用する場合について実験を行う．移住操作を行う場合は，最良個体，部分個体群ともに 50 世代間隔で移住を行う．これは，文献 [4] においてアンサンブル識別器の識別性能が良い結果を示した世代間隔を用いている．また，島数を変更し，アンサンブル識別器の識別性能への影響を調べる．学習用データには UCI Machine Learning Repository で提供されている中から Phoneme (パターン数: 5404，属性数: 5，クラス数: 2) と Satimage (パターン数: 6435，属性数: 36，クラス数: 6) の 2 種類の実世界データを用いた．それぞれのデータの連続値を持つ属性においては，各属性値の最小値と最大値を用いて [0, 1] の範囲で正規化する．それぞれの実験では以下の設定を共通して用いる．

試行回数	: 30 (10-fold cross-validation × 3)
終了条件	: 50,000
個体群サイズ	: 300
EMOA	: NSGA-II
島数	: 3, 5, 7, 9

4.2. 数値実験結果

4.2.1. 単一の弱識別器で構成されるアンサンブル識別器

本節では，各部分個体群から，全学習用データの誤識別率が最小の個体を 1 つ弱識別器として抽出したアンサンブル識別器について実験を行う．加えて，全ての部分個体群を統合し，全学習用データの誤識別率が最小の個体を 1 つ選択した単一識別器と比較を行う．多数決には，単純多数決 (simple) と，部分学習用データの識別率を重みとするサブデータ重み多数決 (sub)，全学習用データの識別率を重みとする全データ重み多数決 (all) の 3 つの方法を用いる．数値実験を 30 回試行して得られた評価用データの誤識別率の平均を表 1 および表 2 に示す．移住操作設定が同じ範囲で，アンサンブル識別器による誤識別率が単一識別器による誤識別率よりも低くなった結果を太字，各行において誤識別率が最も低くなった結果を下線 (と赤字) で示す．

表 1: 評価用データ誤識別率 [%] (Phoneme，単一弱識別器)

島数	移住操作間隔: なし				移住操作間隔: 50 世代			
	単一識別器	simple	sub	all	単一識別器	simple	sub	all
3	16.38	**15.52**	**15.53**	**15.53**	15.44	**15.37**	**15.37**	**15.37**
5	17.09	**16.27**	**16.27**	**16.27**	16.19	**16.16**	**16.17**	**16.16**
7	17.89	**16.75**	**16.76**	**16.75**	17.06	17.20	17.20	17.20
9	18.14	**17.13**	**17.14**	**17.14**	17.28	17.36	17.35	17.36

表 2: 評価用データ誤識別率 [%] (Satimage，単一弱識別器)

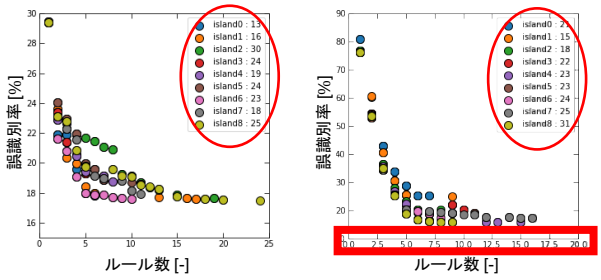
島数	移住操作間隔: なし				移住操作間隔: 50 世代			
	単一識別器	simple	sub	all	単一識別器	simple	sub	all
3	14.34	**13.26**	**13.24**	**13.16**	13.57	13.61	13.61	13.61
5	15.03	**14.05**	**14.00**	**13.98**	13.96	**13.94**	**13.95**	**13.95**
7	16.05	**14.26**	**14.23**	**14.27**	14.31	**14.18**	**14.16**	**14.16**
9	16.48	**14.25**	**14.21**	**14.24**	14.58	**14.51**	**14.49**	**14.50**

表 1, 2 より，提案手法で得られたアンサンブル識別器の全ての誤識別率が太字で示されている．よって，提案手法で得たアンサンブル識別器の汎化性能が，単一識別器よりも高いことが分かる．また，島数が 7 および 9 の場合，提案手法で得たアンサンブル識別器の各誤識別率は，移住操作を適用した単一識別器の誤識別率よりも低いことが分かる．特に，島数が 9 の場合では，データセットにかかわらず提案手法による結果が下線で示されている．このことより，島数が大きい場合では，提案手法が最も有効となることが予測される．

Phoneme では，提案手法において単純多数決と重み付け多数決で大きな差は見られないが，Satimage では，重み付け多数決の方が単純多数決よりも低い誤識別率が得られた．これは，Satimage は Phoneme よりもクラス数が多いため，単純多数決において同票になる可能性が高く，ランダム依存性が強いことによる識別性能の低下が原因だと考えられる．

4.2.2. 非劣解集合で構成されるアンサンブル識別器

本節では，各部分個体群から，2 目的において非劣な識別器集合を弱識別器集合として抽出したアンサンブル識別器について実験を行う．数値実験を 30 回試行して得られた評価用データの誤識別率の平均を調べると，全ての場合において表 1, 2 の提案手法で得られたアンサンブル識別器の誤識別率よりも劣っていた．ここで，本節で用いられた非劣な弱識別器集合のトレードオフ曲線に沿った分布を調べると，図 2 のようにルール数最小化に偏って分布していた．弱識別器集合に，図 2 のようなルール数最小化への偏りがあるため，アンサンブル識別器の誤識別率最小化の劣化が生じたと考えられる．



(a) Phoneme (b) Satimage

図 2: 非劣な弱識別器集合の 2 目的における分布 (島数: 9)

5. おわりに

本研究では，移住操作を適用しない島型の並列分散実装を MoFGBML に適用し，得られた各部分個体群から弱識別器を抽出するアンサンブル識別器を設計した．数値実験として，移住操作を適用した並列分散実装の識別性能と比較し，提案手法の汎化性能の高さを検証した．加えて，島数を変更した場合のアンサンブル識別器の識別性能への影響を調べ，島数が大きい場合の提案手法の有効性を検証した．

本手法で得られる非劣な弱識別器集合には，ルール数最小化に偏った分布が見られた．今後の課題として，トレードオフ曲線における分布の偏りを考慮した弱識別器の選択方法を考えることが挙げられる．

参考文献

[1] Y. Nojima, Y. Takahashi, and H. Ishibuchi, "Application of parallel distributed implementation to multiobjective fuzzy genetics-based machine learning," *Lecture Notes in Computer Science 9011: Intelligent Information and Database Systems – ACIIDS 2015*, Part I, pp. 462-471, Springer, Berlin, March 2015.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[3] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, pp. 1-39, 2010.

[4] H. Ishibuchi, M. Yamane, and Y. Nojima, "Ensemble fuzzy rule-based classifier design by parallel distributed fuzzy GBML algorithms," *Proc. Of 9th International Conference on Simulated Evolution and Learning – SEAL 2012*, pp. 93-103, Hanoi, Vietnam, December 16-19, 2012.