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

# 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，識別性能の高さと解釈性能の高さ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．先行研究 [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．文献 [1] の島型MoFGBML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．

本研究では文献 [1] で行われていた移住操作は適用せずに学習を行う．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．

# 識別性能と解釈性能の2目的最適化

## 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**x** = (*x*1, …, *xn*) のように表され，*xi*は第*i*次元 (*i* = 1, 2, …, *n*) における属性値を表す．

Rule *R*: If *x*1 is *A*1 and ... and *xn* is *An*

then Class *C* with *CF*  (1)

**A** = (*A*1, …, *An*)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．

## 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．

# 並列分散型MoFGBMLによるアンサンブル識別器

## 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正解ラベルの比を保つように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]

## 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，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．単純多数決と比較して，重み付け多数決では多数決結果が同票となる可能性が低くなることが期待される．

# 数値実験

## 数値実験設定

本研究で設計したアンサンブル識別器と [1] の並列分散実装で得られた識別器の識別性能を比較した．このとき，[1] で行う交換間隔と移住間隔はともに50世代とした．また，並列分割数を変更し，アンサンブル識別器の識別性能への影響を調べた．数値実験にはKEEL-data set repositoryにより提供されている以下の2種類の実世界データを用いた（表1）．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．以下に数値実験における各種設定を示す．

|  |  |  |
| --- | --- | --- |
| 試行回数 | : | 30 (10-fold cross-validation x 3) |
| 終了条件 | : | 50,000 |
| 個体群サイズ | : | 300 |
| EMOA | : | NSGA-II |
| 並列分割数 | : | 3, 5, 7, 9 |
| 移住操作間隔 | : | 50世代間隔，なし |
| 交換操作間隔 | : | 50世代間隔，なし |

表1: 使用するデータセット

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Patterns | Attributes | Classes |
| Phoneme | 5404 | 5 | 2 |
| Satimage | 6435 | 36 | 6 |

## 数値実験結果

### 単一弱識別器によるアンサンブル識別器

本節では，各島の部分個体群から，全学習用データに対する誤識別率が最小の個体を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った．並列分散実装で得られた全ての個体群の中から学習用データに対して誤識別率が最も低くなる識別器を単一に選択した場合と，アンサンブル識別器の誤識別率を比較した．重み付け多数決では，割り当てられた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と，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の2種類の方法で実験した．

交換操作と移住操作を適用しない並列分散実装を30回試行し，得られたアンサンブル識別器の誤識別率の平均を表2，表3に示す．また，交換操作と移住操作を適用した場合の結果を表4, 5に示す．誤識別率が最も低くなった結果を太字で表している．

表2: 評価用データ誤識別率 [%]  
(Phoneme，単一弱識別器，交換・移住操作なし)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 島数 | 単一識別器 | 単純 多数決 | サブデータ  重み多数決 | 全データ 重み多数決 |
| 9 | 18.14 | **17.13** | 17.14 | 17.14 |
| 7 | 17.89 | **16.75** | 16.76 | **16.75** |
| 5 | 17.09 | **16.27** | **16.27** | **16.27** |
| 3 | 16.38 | **15.52** | 15.53 | 15.53 |

表3: 評価用データ誤識別率 [%]  
(Satimage，単一弱識別器，交換・移住操作なし)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 島数 | 単一識別器 | 単純 多数決 | サブデータ  重み多数決 | 全データ 重み多数決 |
| 9 | 16.48 | 14.25 | **14.21** | 14.24 |
| 7 | 16.05 | 14.26 | **14.23** | 14.27 |
| 5 | 15.03 | 14.05 | 14.00 | **13.98** |
| 3 | 14.34 | 13.26 | 13.24 | **13.16** |

表4: 評価用データ誤識別率 [%]  
(Phoneme，単一弱識別器，交換・移住操作50世代間隔)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 島数 | 単一識別器 | 単純 多数決 | サブデータ  重み多数決 | 全データ 重み多数決 |
| 9 | **17.28** | 17.36 | 17.35 | 17.36 |
| 7 | **17.06** | 17.20 | 17.20 | 17.20 |
| 5 | 16.19 | **16.16** | 16.17 | **16.16** |
| 3 | 15.44 | **15.37** | **15.37** | **15.37** |

表5: 評価用データ誤識別率 [%]  
(Satimage，単一弱識別器，交換・移住操作50世代間隔)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 島数 | 単一識別器 | 単純 多数決 | サブデータ  重み多数決 | 全データ 重み多数決 |
| 9 | 14.58 | 14.51 | **14.49** | 14.50 |
| 7 | 14.31 | 14.18 | **14.16** | **14.16** |
| 5 | 13.96 | **13.94** | 13.95 | 13.95 |
| 3 | **13.57** | 13.61 | 13.61 | 13.61 |

表2, 3, 4, 5より，アンサンブル識別器を設計することで単一識別器よりも識別性能が高くなることが分かる．また，島数が増えるにつれて誤識別率は高くなっているが，交換操作，移住操作を適用しないアンサンブル識別器では，分割数が少ない場合の単一識別器よりも誤識別率が低いことが分かる．重み付け多数決について，クラス数が少ないデータセット (Phoneme) では，単純多数決と重み付け多数決に大きな差は見られないが，クラス数が多いデータセット (Satimage) では，重み付け多数決による識別性能の向上が見られた．

### 非劣解集合弱識別器によるアンサンブル識別器

本節では，全学習用データの誤識別率最小化とルール数最小化の2目的に対して，非優越ランキング [2] に基づいた非劣な識別器集合を弱識別器とする．このとき，島ごとに非劣な弱識別器を全て用いて多数決を行った結果を表6, 7に示す．

表6: 評価用データ誤識別率 (Phoneme, 非劣弱識別器) [%]

|  |  |  |  |
| --- | --- | --- | --- |
| 島数 | 単純多数決 | サブデータ 重み多数決 | 全データ 重み多数決 |
| 9 | 18.31 | 18.27 | 18.29 |
| 7 | 17.64 | 17.50 | 17.51 |
| 5 | 17.10 | 17.11 | 17.14 |
| 3 | 16.17 | 16.17 | 16.17 |

表7: 評価用データ誤識別率 (Satimage, 非劣弱識別器) [%]

|  |  |  |  |
| --- | --- | --- | --- |
| 島数 | 単純 多数決 | サブデータ  重み多数決 | 全データ 重み多数決 |
| 9 | 17.00 | 16.37 | 16.49 |
| 7 | 16.71 | 16.06 | 16.12 |
| 5 | 15.88 | 15.39 | 15.41 |
| 3 | 13.95 | 13.65 | 13.70 |

表6, 7では，表2, 3と比較して，非劣解集合で弱識別器を構成することによる識別性能の向上は見られなかった．これは，ルール最小化に偏った非劣解が弱識別器として抽出されていたからであると考えられる．

# おわりに

本研究では，交換操作，移住操作を行わない並列分散実装をMoFGBMLに適用し，アンサンブル識別器を設計した．数値実験として，誤識別率が最小である識別器を単一に選択した場合と比較し，アンサンブル識別器の有効性を検証した．

今後の課題としては，MoFGBMLで獲得された非劣解集合を有効に利用するため，非劣解集合の分布に応じ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.