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で設計されるため，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化する識別器の獲得は困難である．そのため，進化型多目的最適化手法を用いてファジィ識別器の設計を行う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，ビッグデータなどの大規模なデータに対する機械学習の発展が期待されている．MoFGBMLは高い識別性能を持つが，大規模なデータに適用する際に膨大な計算時間を必要とする問題がある．先行研究では，Island型の並列分散手法をMoFGBMLに適用した計算時間の短縮が提案されている． [2]

従来の並列分散型MoFGBMLでは部分学習用データへの過学習を防ぐために，部分学習用データの交換操作と部分個体群の移住操作が行われているが，その結果，獲得される識別器集合の多様性が劣る課題がある．本研究では，交換操作，移住操作を適用せずに多様性を考慮した識別器の探索を行う．また，識別性能の高い識別器を獲得するため，各部分個体群から抽出した弱識別器で構成されるアンサンブル識別器を設計する．各島では独立な進化型多目的最適化が行われるため，各島から抽出した弱識別器の間には高い多様性が期待できる．これらの弱識別器の多数決によってパターンを識別するアンサンブル識別器を設計し，識別性能の向上を図る．

# 多目的ファジィ遺伝的機械学習

## ファジィ識別器

*n*次元のパターン数xp = (xp1, …, xpn)が*m*個与えられている*M*クラスのパターン識別問題があるとき，xpiは*p*番目のパターンの第*i*次元 (*i* = 1, 2, …, *n*) における属性値を表す．この問題に対して，以下のファジィ集合を前件部条件とするIf-thenルールを用いてファジィ識別器を設計する．

Rule *Rq*: If *xp*1 is *Aq*1 and ... and *xpn* is *Aqn*

then Class *Cq* with *CFq*， (1)

## 多目的ファジィ遺伝的機械学習

GBMLには個体がルール集合を表現するPittsburgh型のアプローチと個体がルールを表現するMichigan型のアプローチがある．本研究で用いる多目的ファジィGBMLは，Pittsburgh型のGBMLを基本として用い，そのルール集合の中のルールを個体として扱うMichigan型のアプローチをルールに対する局所探索として取り入れている．

本研究では，進化型多目的最適化アルゴリズムの代表的な手法の1つであるNSGA-II [] をFGBMLに適用し，識別器の誤識別率とルール数の2つの目的を同時に最小化するような識別器の探索を行う．

Minimize (2)

以下にFGBMLの手順を示す．

# 並列分散型MoFGBMLによるアンサンブル識別器

## 並列分散実装

先行研究 [] において，ファジィGBMLの計算時間短縮のための方法としてIsland型の並列分散実装が提案されている．並列分散実装では，個体群と学習用データを分割した部分集合に対して，複数のCPUコアを用いてMoFGBMLを適用する．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コアに割り当てて進化型多目的最適化を行うことで計算時間の短縮を図る．Island型並列分散実装では，一定間隔で最良個体の異なる島への移住操作と部分学習用データ集合の交換操作が行われる．これらの二つの操作により，部分学習用データへの過学習を防ぎ，汎化性能を向上させることが期待される．各島において学習終了後，全ての個体は一つの個体群に統合される．最後に，全ての個体は式(2)の2目的について評価される．この評価には，NSGA-IIで用いられる非優劣ランキングに基づいて行われる．



図1: 並列分散実装のモデル図

## アンサンブル識別器の設計

## 正規化MOEA/Dの問題点

図2では，2目的最小化問題に対してスカラー化関数としてPBI (Penalty-based Boundary Intersection) を用いたときにMOEA/Dにおける***w*** = (0.2, 0.8) の重みベクトルが獲得する最良の解を示している．また，図2 (a), (b) は，それぞれPBIで設定する必要のあるペナルティパラメータ**を1.0と5.0と指定した時に獲得される解を示している．**の値が小さいとき (図2 (a))，等高線によってできる最大傾斜方向は，ほとんど参照点***r***\*の方向を指している．そのため，Pareto front上に獲得される解が対応する重みベクトルから遠いことがわかる．**の値が大きいとき (図2 (b))，等高線によってできる最大傾斜方向は，重みベクトルに向かう方向に近づく．つまり，重みベクトルから遠い解は許容されない．結果として，獲得される解は，対応する重みベクトルに近いことが期待される．そのため，重みベクトルが一様に分布していれば，一様な解集合を獲得できることから，** = 5.0が多くの論文で頻繁に用いられる．本論文でも，** = 5.0とする．

 

(a) ** = 1.0 (b) ** = 5.0

図2: 重みベクトル ***w*** = (0.2, 0.8) が獲得する解

図3に正規化を行った際に各重みベクトルに対して生成される等高線を示す．正規化MOEA/Dでは，正規化空間内で重みベクトルが一様に分布され，正規化された目的関数値に基づき，スカラー化関数値を求めることで解が評価される．つまり，図3 (b) のように** = 5.0で定まる等高線は正規化空間上で生成される．一方で，これらの等高線は，元の目的関数空間で見ると，各重みベクトルによって異なる角度を持った等高線となる．つまり，重みベクトル上に解を獲得しようと** = 5.0と設定したとしても，重みベクトル上に解を獲得できない等高線を持つベクトルが生成されることで最終的に一様な解集合を獲得できなくなる．

 

(a) 元の目的関数空間 (b) 正規化空間

図3: 正規化MOEA/Dの等高線

# Penalty-based Boundary Intersection for Normalization

## 提案手法

図4では，目的関数毎に異なる範囲の正規化が行われている様子を示している．*f*1に関しては最小値が0, 最大値が1の範囲で正規化が行われ，*f*2に関しては，最小値が0，最大値が2の範囲で正規化が行われる．このように*f*1方向と*f*2方向に関して正規化を行う範囲が異なることで各重みベクトルの等高線の広がり方が変化する．



図4: 目的関数空間における探索領域の変化

本論文では，目的関数毎に異なる範囲の正規化が行われることを考慮し，各重みベクトルの探索領域を変化させることのできるスカラー化関数 (PBI-N: Penalty-based Boundary Intersection for Normalization) を提案する．図5 は，図4と同じ正規化をした状態でPBI-N を用いたときの探索領域を元の目的関数空間と正規化空間に示している．図5 (a)と図4を比べると，重みベクトル***w*** = (0, 1) における探索領域に変化はないが，重みベクトル***w*** = (1, 0) における探索領域は，図4では大きく，図5 (a)では小さくなる．また，図5 (a)における元の目的関数空間内の重みベクトル***w*** = (1, 0)と重みベクトル***w*** = (0, 1)の探索領域の大きさ (等高線の角度) は等しい．提案手法は，このように，等高線の角度を元の目的関数空間内で固定するように正規化空間上の等高線の角度を定める (図5 (b)) ことで重みベクトル上に解を獲得できる．

 

(a) 元の目的関数空間 (b) 正規化空間

図5: PBI-Nによる探索領域の調整

## 数値実験

提案手法 (NMOEA/D-PBI-N) の性能を数値実験により示す．通常のMOEA/Dとスカラー化関数は通常のPBIを用いるが正規化は行う正規化MOEA/D (NMOEA/D) を比較手法として用いる．テスト問題として，DTLZ1 [6] とWFG4 [5] を用いる．DTLZ1はPareto front上の各目的関数値の値域がそれぞれ等しく，一般的には正規化が必要とされない問題である．一方でWFG4はPareto front上の第*i*目的の値域が[0, 2*i*]で示される目的関数間のスケールが異なる問題で正規化を必要とする．表1では，51回施行におけるHypervolumeの平均値を示しており，より大きな値が良い解集合であることを示している．図6では，各アルゴリズムに関して3目的WFG4における中央値を示した施行の解分布を示している．

表1から，通常のMOEA/DとNMOEA/Dを比べたとき， NMOEA/Dの性能が悪化していることがわかる．特に，8目的，10目的問題では，性能の悪化が著しい．これらのことから，正規化を必要とする問題であるかないかに関わらず，MOEA/Dに対し単純に正規化を用いるだけでは性能を悪化させてしまうことがわかる．NMOEA/D-PBI-Nでは，特に8目的，10目的問題で最良の結果を示している．3目的問題に関しても図6 (b), (c)で示している通り，単純に正規化を行っただけのMOEA/D (NMOEA/D) と同じようにPareto front上に一様に分布する解集合を獲得できる．これらのことから，正規化を行ったMOEA/Dに対し，提案手法であるPBI-Nを用いることが有効であることがわかる．また，正規化を必要とする問題であるかないかに関わらず，MOEA/Dに対し単純に正規化を用いるだけでなく，スカラー化関数としてPBI-Nを用いることがより良い性能を示すために必要であることがわかる．

表1: MOEA/DとNMOEA/D, NMOEA/D-PBI-Nの比較

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 問題 | 目的数 | MOEA/D | NMOEA/D | NMOEA/D-PBI-N |
| DTLZ1 | 3 | 1.1163 | **1.0308** | **1.1168** |
|  | 5 | **1.5777** | **0.3836** | 1.5774 |
|  | 8 | 2.1364 | **0.2840** | **2.1368** |
|  | 10 | 2.5925 | **0.2839** | **2.5927** |
| WFG4 | 3 | 0.6853 | **0.6900** | **0.6755** |
|  | 5 | **1.1513** | **1.2695** | 1.2637 |
|  | 8 | 1.1739 | **0.1949** | **1.9447** |
|  | 10 | 1.4580 | **0.2474** | **2.5085** |

  

(a) MOEA/D (b) NMOEA/D (c) NMOEA/D-PBI-N

図6: 3目的WFG4の解集合

# おわりに

本概要では，通常のMOEA/Dに対し，正規化を用いることの問題点を明らかにし，その問題点をもとに新たなスカラー化関数であるPBI-Nを提案した．正規化を必要としないDTLZ1と正規化を必要とするWFG4をテスト問題として通常のMOEA/Dと正規化を行っただけのMOEA/Dと性能比較を行い，提案手法の有効性を示した．これらの性能比較調査から，MOEA/Dに正規化を使用し，PBI-Nを用いることでより良い性能を示すことがわかった．

本論文では，PBI-Nの具体的な定義を示し，DTLZ1-4, WFG4-9，正規化アルゴリズムの調査用に提案された各目的関数間のスケールの違いが大きいRescaled DTLZ1-4, Deceptive Rescaled DTLZ1-4 [4]を用いることで正規化MOEA/D-PBI-Nの性能を調査する．

# 参考文献

1. Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Trans. on Evolutionary Computation*, vo. 11, no. 6, pp. 712-731, 2007.
2. K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part I: Solving problems with box constraints,” *IEEE Trans. on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, August 2014.
3. Y. Yuan, H. Xu, B. Wang, and X. Yao, “A new dominance relation based evolutionary algorithm for many-objective optimization,” *IEEE Trans. on Evolutionary Computation*, vol. 20, no. 1, pp. 16-37, February 2016.
4. H. Ishibuchi, K. Doi, and Y. Nojima, “On the effect of normalization in MOEA/D for multi-objective and many-objective optimization,” *Complex and Intelligent Systems*, vol. 3, no. 4, pp. 279-294, 2017.
5. S. Huband, P. Hingston, L. Barone, and L. While, “A review of multiobjective test problems and a scalable test problem toolkit,” *IEEE Trans. on Evolutionary Computation*, vol. 10, no. 5, pp. 477-506, October 2006.
6. K. Deb, A. Pratap, S. Agrawal, and T. Meyariyan, “A fast elitist non-dominated sorting genetic algorithm: NSGA-II,” *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.