メンバーシップ関数の種類によるファジィ識別器への影響調査

第 8 グループ 瀧川 弘毅



三角形型

ガウシアン型

台形型

均等区間型

図2: 3分割の各メンバーシップ関数

# はじめに

識別性能が高い識別器が多く研究されているが，それらはもののブラックボックス化しており識別の判断理由を説明できない．ファジィ識別器では言語的に解釈可能なルール群から構成されているので，どのように識別しているかが解釈可能であるという特徴を持つ．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．

実世界では識別されるデータセットも様々なものがあり，データセットに対して必要な識別器も変わってくる．データセットに応じて，適切なメンバーシップ関数を用いた識別器を生成することで識別器の性能向上を目指す．本研究では前段階としてメンバーシップ関数を変更して識別器を生成し，識別器の性能への影響を調査した．これにより，識別器を生成する前にデータセットの特徴を知っていれば最適なメンバーシップ関数の選択ができ，より良い識別器を得られる．

## 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．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．

## メンバーシップ関数

適応させる条件部集合への帰属度を表現する関数．本研究では三角形型，ガウシアン型，台形型，均等区間型の4種類のメンバーシップ関数を用いる．各型について，2, 3, 4, 5分割のファジィ集合14種と，メンバー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．均等区間型はクリスプ集合であり，ファジィ型ではない．ファジィ型にすることで1つのルールが覆う値域が広がり，識別境界に曖昧性を持たすことができる．

**図1: ファジィ型メンバーシップ関数の値域**

## 多目的最適化

本研究では，ファジィ識別器の誤識別率の最小化とルール数最小化の2つの目的を用いる．本研究では代表的な進化的多目的最適化アルゴリズムであるNSGA- IIをファジィ識別器に適応し2目的最適化を図る．誤識別率が低ければ識別性能が高く，ルール数が少なければ複雑性が少なくなり計算コストが少なく解釈性能が高い識別器と言える．

# 数値実験

実験は以下の設定で行った．

|  |  |  |
| --- | --- | --- |
| 試行回数 | : | 30 (10-fold cross-validation 3) |
| 最大ルール数 |  | 60 |
| 個体群サイズ | : | 60 |
| 世代数 |  | 1000 |
| EMOA | : | NSGA-II |

データセットには以下を用いた．

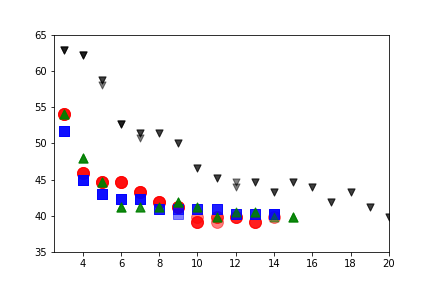
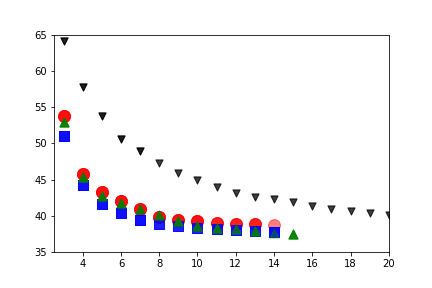
|  |  |  |  |
| --- | --- | --- | --- |
|  | パターン数 | 属性数 | クラス数 |
| iris | 150 | 4 | 3 |
| wine | 178 | 13 | 3 |
| phoneme | 5404 | 5 | 2 |
| sonar | 208 | 60 | 2 |
| yeast | 1484 | 8 | 10 |

三角形型，ガウシアン型，台形型，均等区間型の4種類のメンバーシップ関数を用いて生成された識別器を上のデータセットから学習を行い．その識別性能を比較した．

# 数値実験結果

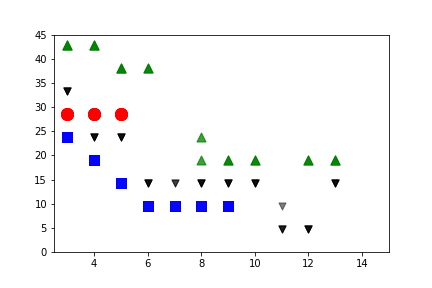
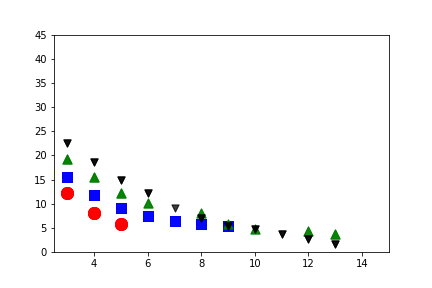
各試行での学習用データ誤識別率最小個体の平均値を表にした．

**表１: 学習用データ誤識別率**



Yeast:学習用データ

Yeast:評価用データ



Sonar:学習用データ

Sonar:評価用データ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Iris | Wine | Phoneme | Sonar | yeast |
| 三角形型 | 0.72 | 0.00 | 16.12 | 6.126 | 38.66 |
| ガウシアン型 | 0.74 | 0.06 | 16.18 | 4.861 | 37.89 |
| 台形型 | 0.49 | 0.00 | 15.07 | 3.882 | 37.72 |
| 区間型 | 0.10 | 0.00 | 16.01 | 1.941 | 40.04 |

**表2: 評価用データ誤識別率**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Iris | Wine | Phoneme | Sonar | yeast |
| 三角形型 | 6.22 | 7.86 | 17.40 | 22.30 | 42.54 |
| ガウシアン型 | 4.89 | 5.46 | 17.42 | 23.82 | 42.92 |
| 台形型 | 5.33 | 5.80 | 16.05 | 22.16 | 42.25 |
| 区間型 | 5.33 | 9.89 | 17.46 | 25.23 | 43.87 |

誤識別率の評価からは台形型メンバーシップ関数の識別器の性能が良いように思われる．

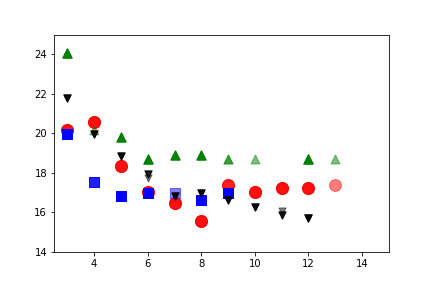
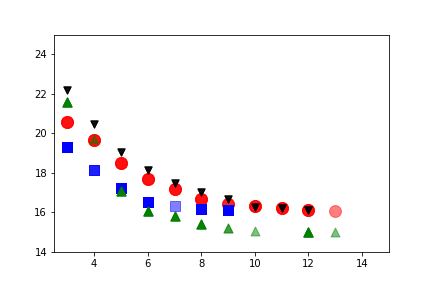
次に各試行での学習用データ誤識別率最小個体のルール数を表にした．

**表3: ルール数**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Iris | Wine | Phoneme | Sonar | yeast |
| 三角形型 | 5.10 | 5.00 | 13.50 | 8.767 | 12.70 |
| ガウシアン型 | 4.03 | 3.57 | 11.33 | 9.133 | 10.83 |
| 台形型 | 5.37 | 5.90 | 16.07 | 11.17 | 14.93 |
| 区間型 | 6.33 | 8.00 | 17.53 | 15.43 | 21.50 |

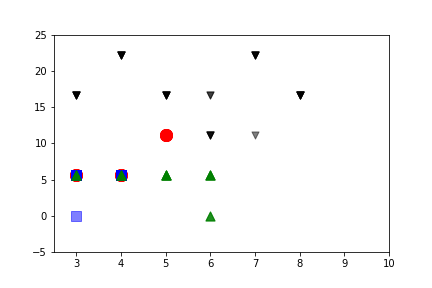
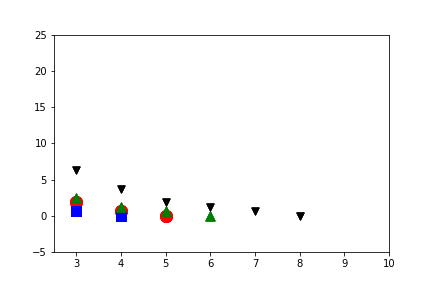
台形型を見てみると，ルール数が多くなっている．これより，台形型の誤識別率が良いのはルール数が多いことが要因であり，識別器の性能自体は高いとは言えないことがわかる．

次に各試行に対してルール数を横軸に，データ誤識別率を縦軸にとってプロットした．



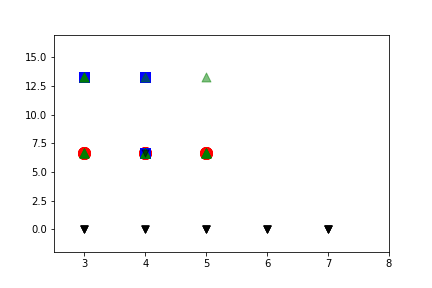
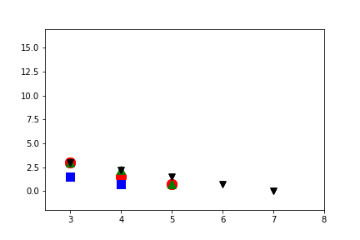
Phoneme:学習用データ

Phoneme:評価用データ



Wine:学習用データ

Wine:評価用データ



Iris:学習用データ

Iris:評価用データ

●三角形型 ■ガウシアン型 ▲台形型 ▼区間型

**図3: ルール数とデータ誤識別率**

IrisやWineなどのクラス境界がはっきりしていると考えられるデータセットでは区間型で十分識別ができるのでファジィ型でメンバーシップ関数を変えても大きな影響は見られなかった．また，台形型などは学習用データに対して過剰に適合し，評価用データでは良い結果を出せず汎用性に欠ける識別器となった．逆にガウシアンや三角形型では評価型で良い結果を出した．これはメンバーシップ関数が覆う値域が広いためファジィルールの汎用性が高いことが理由として考えられる．

# おわりに

本研究では，メンバーシップ関数を変更して識別器を生成し，その識別性能への影響を調査した．その結果，メンバーシップ関数による影響が認められ，メンバーシップ関数が覆う値域が広い関数は汎用性能を高める影響を与えていたことがわかった．

今後の課題として，データセットの特徴を解析してどのような特徴に対してどのメンバーシップ関数が適しているのかなどを調べる必要がある．

# 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.