

Fuzzy Markup Languageを用いたファジィシステムの開発

第35回FSS2019 企画セッション
SOFT-CR連携ファジィ学問塾

○面崎祐一¹, 増山直輝¹, 能島裕介¹, 石渕久生²

¹大阪府立大学, ²南方科技大学

Fuzzy Markup Language (FML)とは？ 2

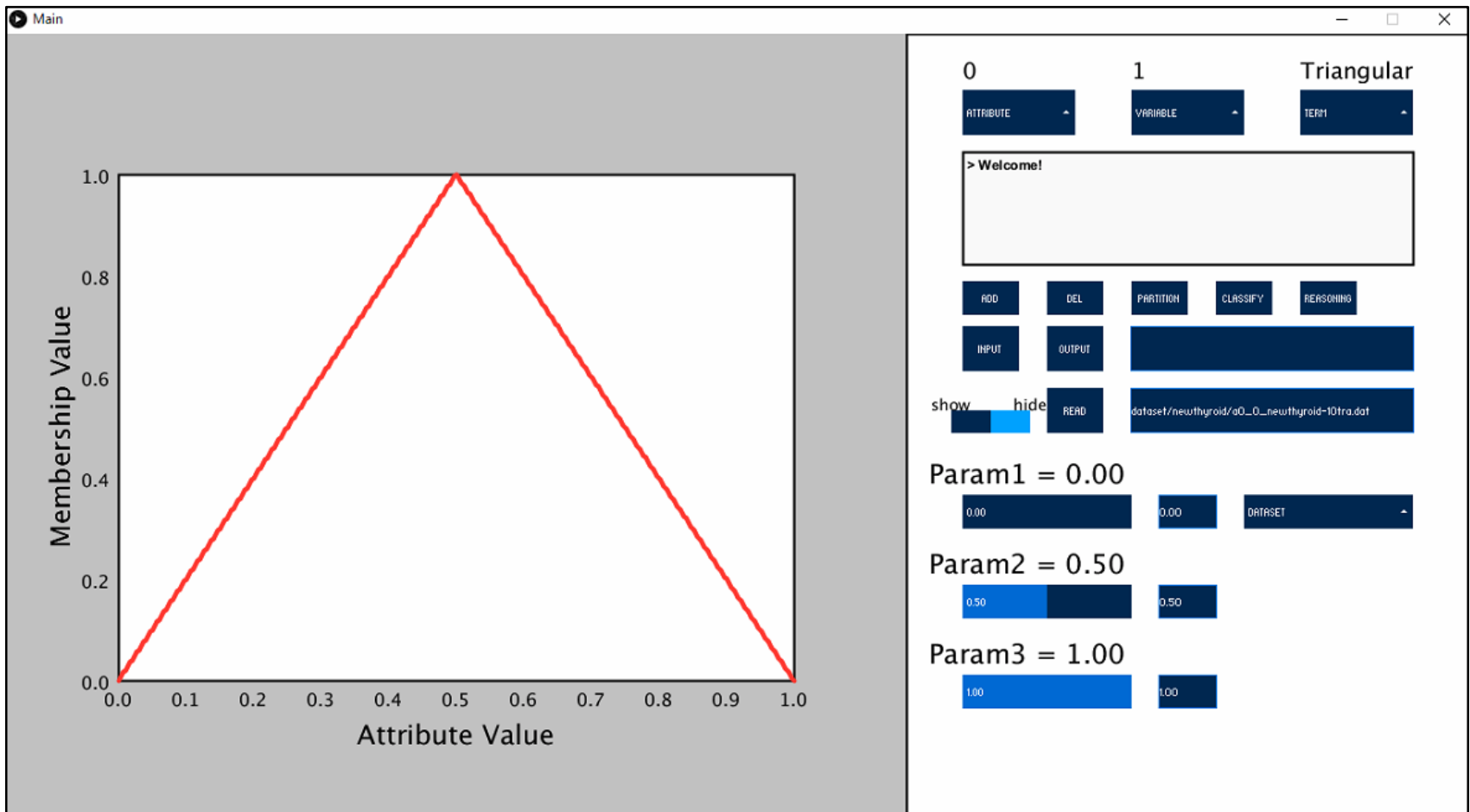
- ファジィシステムに関する標準を定義する IEEE Standard 1855-2016
- 開発環境に依存しないXMLベースの言語仕様
- JFMLというオープンソースライブラリの提供



JFMLを用いたGUIツールの開発

3

FML初学者のFML導入を支援するGUIツールを開発.



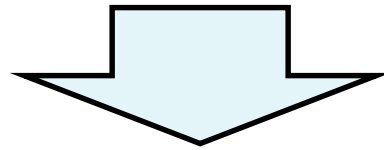
1. ファジィシステム研究の背景
2. ファジィ識別器
3. Fuzzy Markup Language
4. FMLの導入を支援するGUIツール
5. 数値実験
6. まとめ

ファジィシステム研究の背景

AIに求められる意思決定の透明性

5

医療や金融の分野では、意思決定における判断根拠の説明責任が求められる。



ファジィシステムでは、ファジィ集合で表現される言語ラベルによって意思決定が行われるため、判断根拠を解釈可能な言語で表現することが可能である。

Answer 1: SAFE

because x_1 is small and x_2 is medium

Answer 2: DANGER

because x_1 is medium and x_2 is large

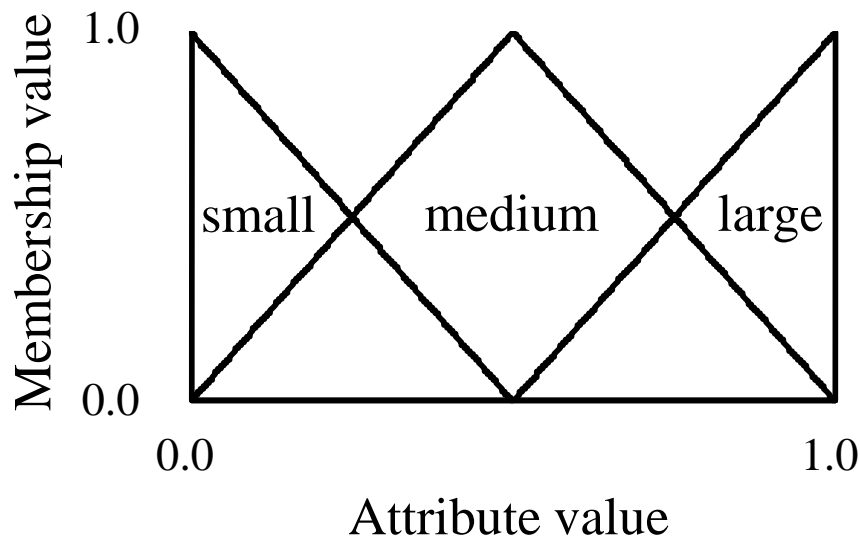
ファジィ識別器

条件部ファジィ集合

6

Rule R_q : If x_1 is A_{q1} and ... and x_n is A_{qn}
条件部
then Class C_q with CF_q

例: $A_q = (\text{small}, \text{medium}, \text{large})$



R_q : q 番目のルール

\mathbf{x} : 入力パターン

A_q : 条件部ファジィ集合

C_q : 結論部クラス

CF_q : ルール重み

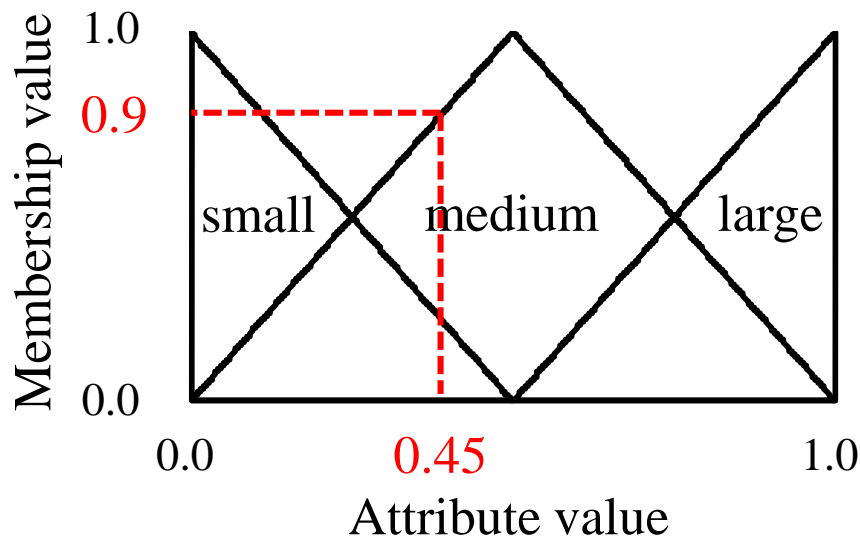
ファジィ識別器

条件部ファジィ集合

7

Rule R_q : If x_1 is A_{q1} and ... and x_n is A_{qn}
条件部
then Class C_q with CF_q

例: $\mathbf{A}_q = (\text{small}, \text{medium}, \text{large})$



例: $x_1 = 0.45$

$A_{q1} = \text{medium}$

x_1 is medium

帰属度を出力する
メンバーシップ関数

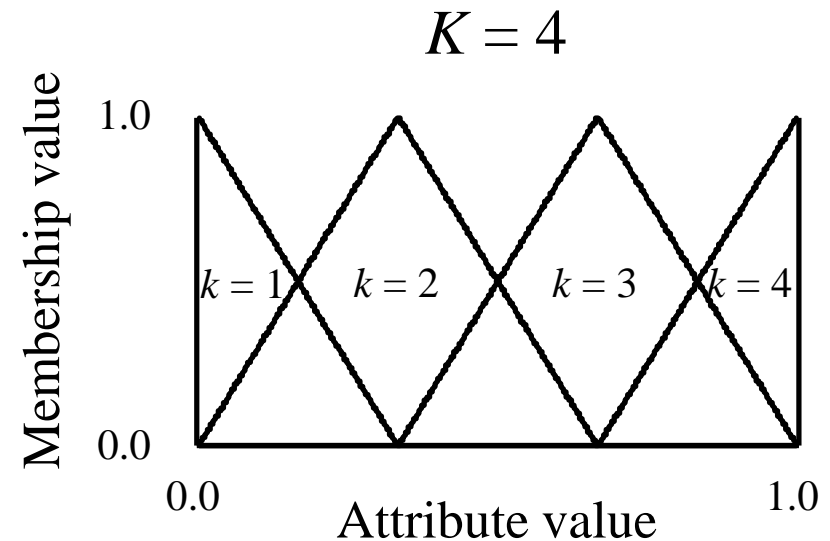
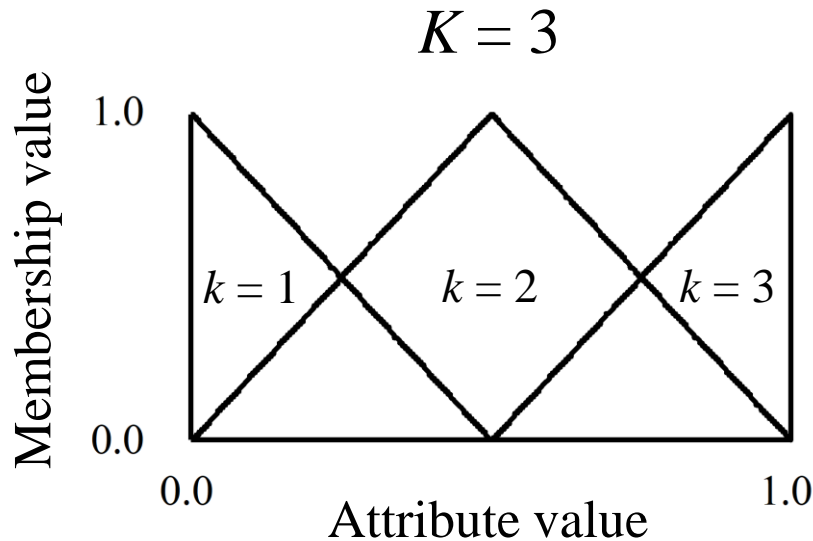
$$\mu_{A_{q1}}(0.45) = 0.9$$

ファジィ識別器

メンバシップ関数

8

条件部ファジィ集合への帰属度を表現する関数



$$\mu_{A_i}(x_i) = \max \left\{ 1 - \frac{|a - x_i|}{b}, 0 \right\}, \quad \begin{aligned} a &= \frac{k - 1}{K - 1}, \\ b &= \frac{1}{K - 1} \end{aligned}$$

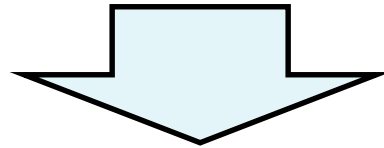
ファジィ識別器

メンバシップ関数

9

条件部ファジィ集合への帰属度を表現する関数

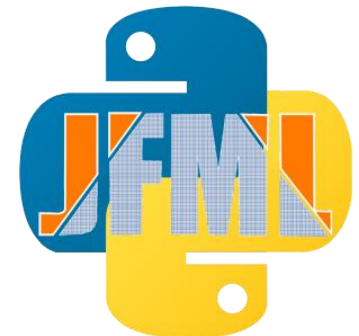
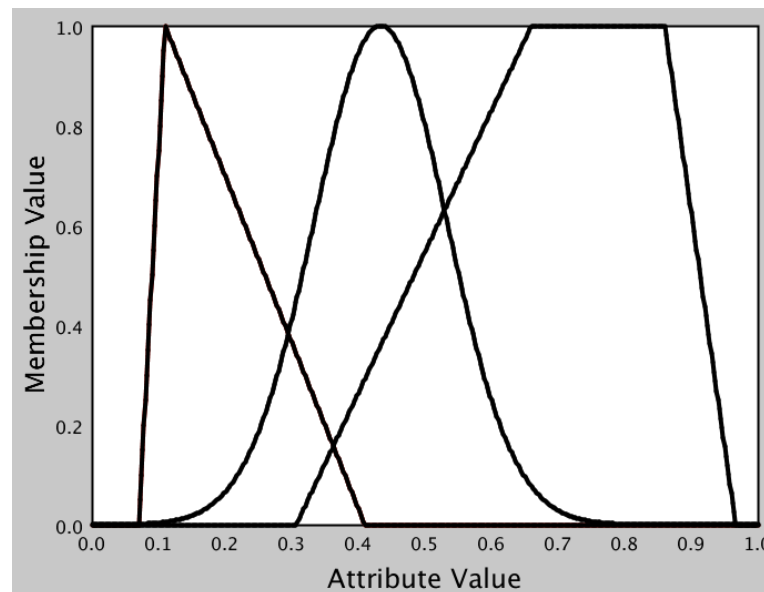
データセットの特徴によって最適な言語ラベルの形状は異なると予想される. 自然言語を定量化したファジィ集合には**多様な形状の設計**が求められる.



問題点

新しい形状のメンバシップ関数を定義する際に,
数学的な定式化とプログラムへの実装が必要である.

- 様々な形状のメンバシップ関数の提供
- Java Fuzzy Markup LanguageというJavaベースのオープンソースライブラリ
- XMLファイルの入出力が可能
- Py4JFMLというPython版も存在



Fuzzy Markup Language

FMLにおけるメンバシップ関数の定義

11

- 様々な形状がJFMLで実装されている
- 複数のパラメータの設定で定義される

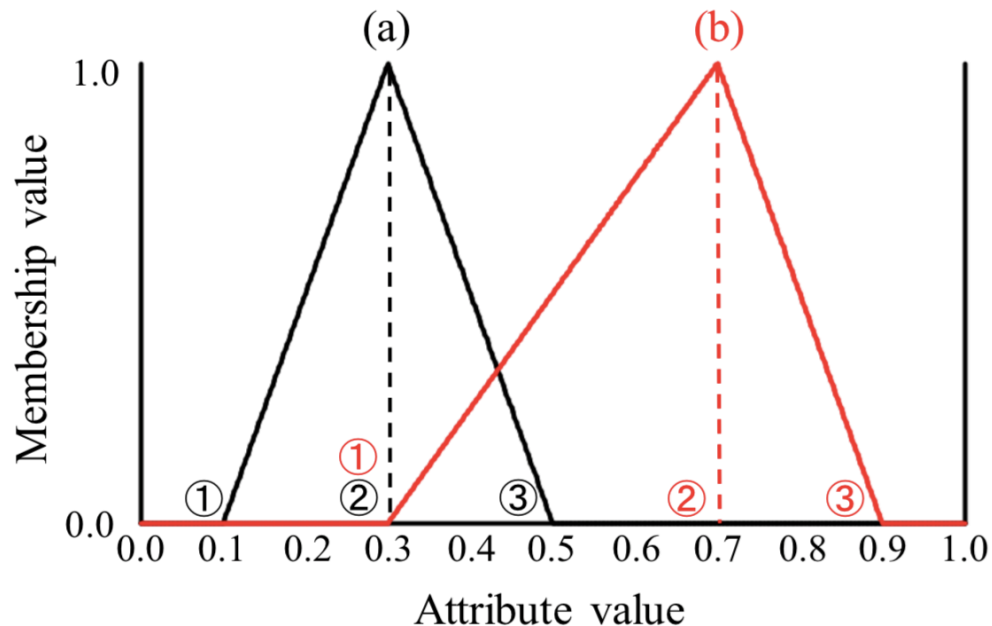
(a) `<triangularShape param1="0.1" param2="0.3" param3="0.5" />`

(b) `<triangularShape param1="0.3" param2="0.7" param3="0.9" />`

形状名

各パラメータ

Triangular

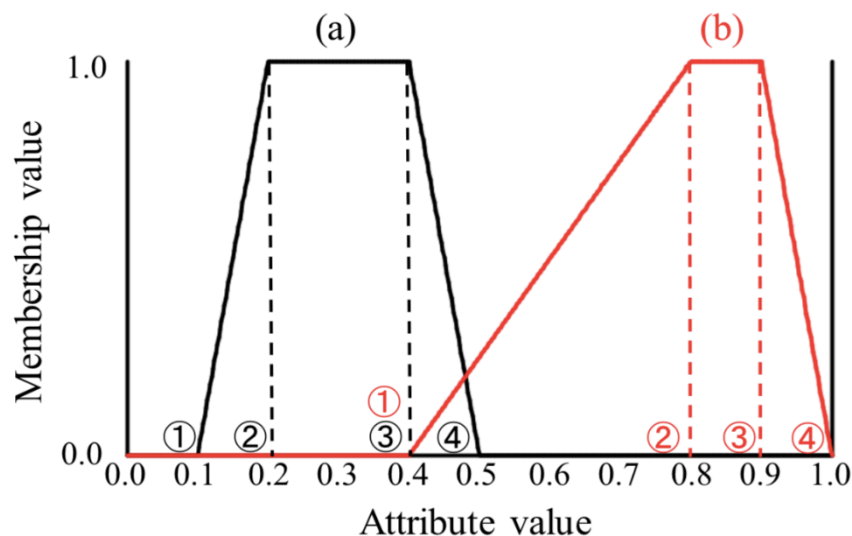


FMLにおけるメンバーシップ関数の定義

- 様々な形状がJFMLで実装されている
- 複数のパラメータの設定で定義される

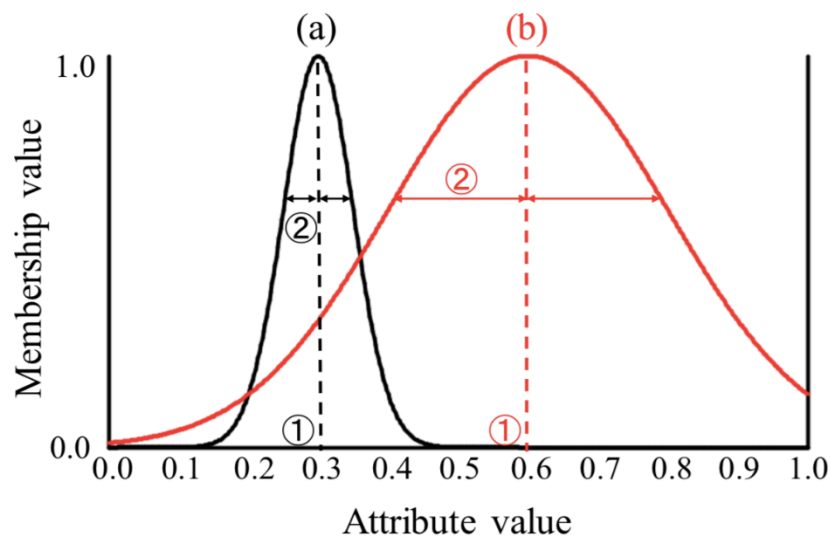
Trapezoid

4点で定義



Gaussian

正規分布の平均と分散



FMLにおけるメンバシップ関数の定義

- 様々な形状がJFMLで実装されている
- 複数のパラメータの設定で定義される

利点

様々なメンバシップ関数が実装されており、**多様な形状**を簡単に使用できる

問題点

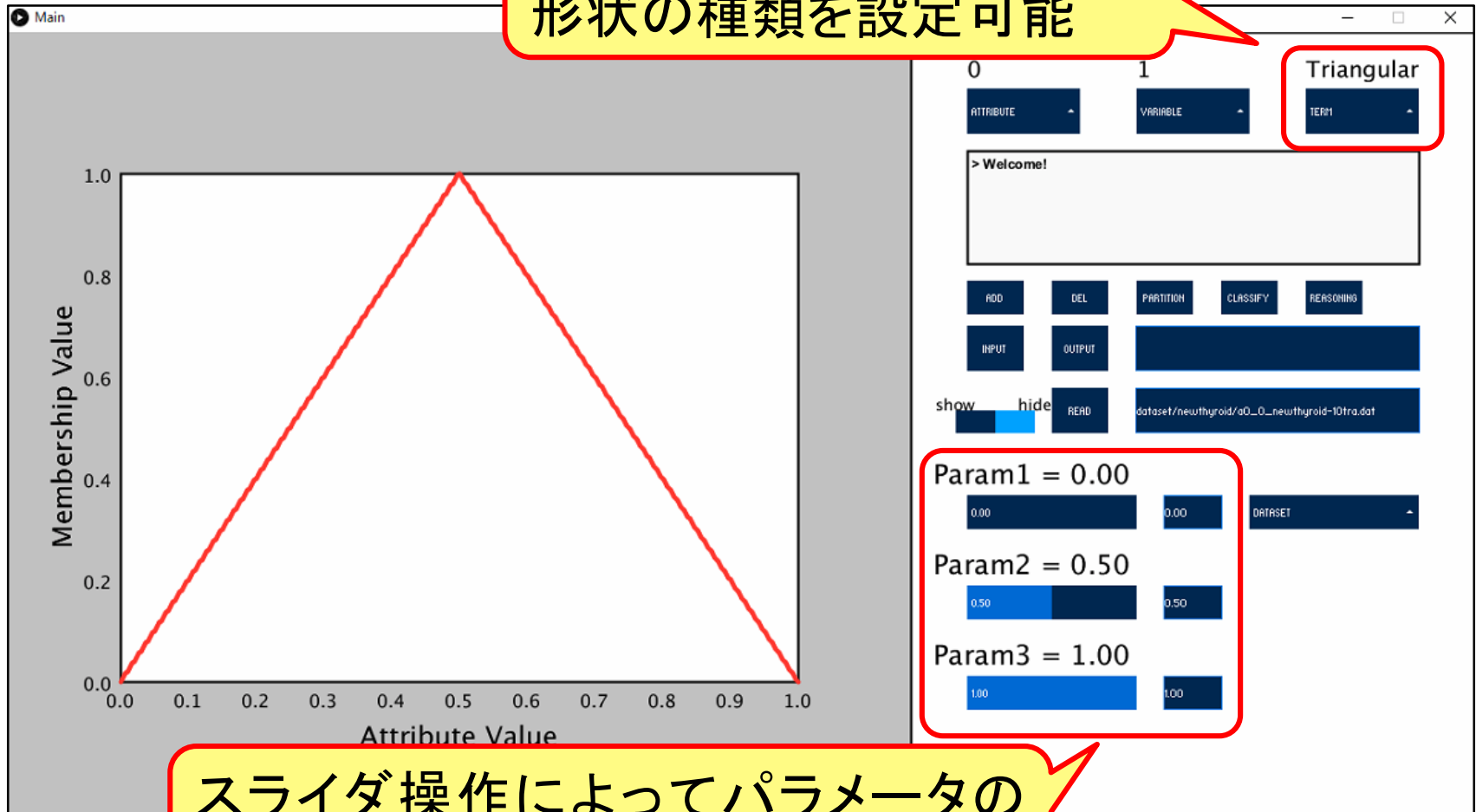
形状の種類名・パラメータの影響の知識が必要なため、初学者にとって導入が難しい

FMLの導入を支援するGUIツール

FML初学者のための支援

14

リストから選択することで
形状の種類を設定可能

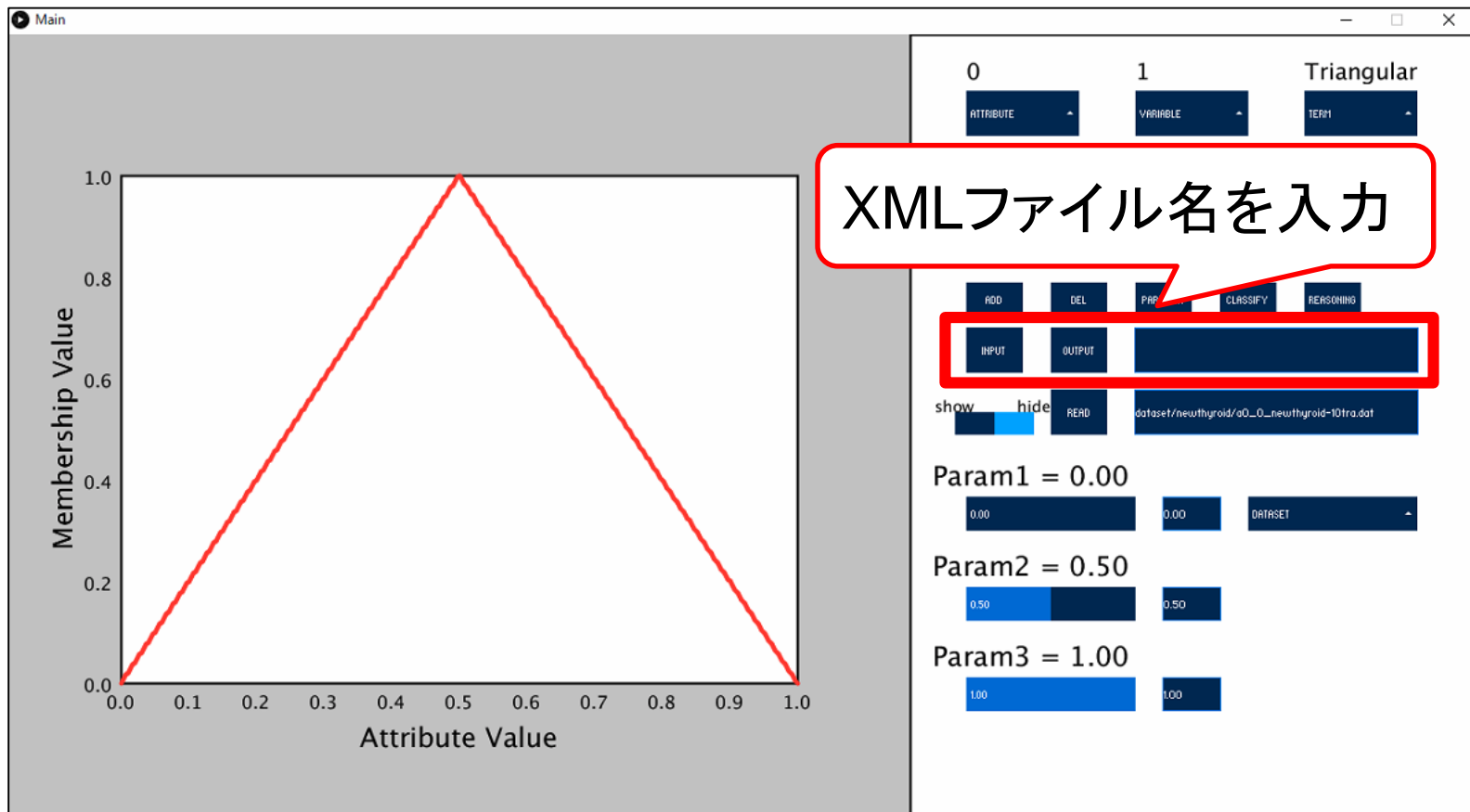


スライダ操作によってパラメータの
影響をリアルタイムに確認可能

FMLの導入を支援するGUIツール ユーザビリティ

15

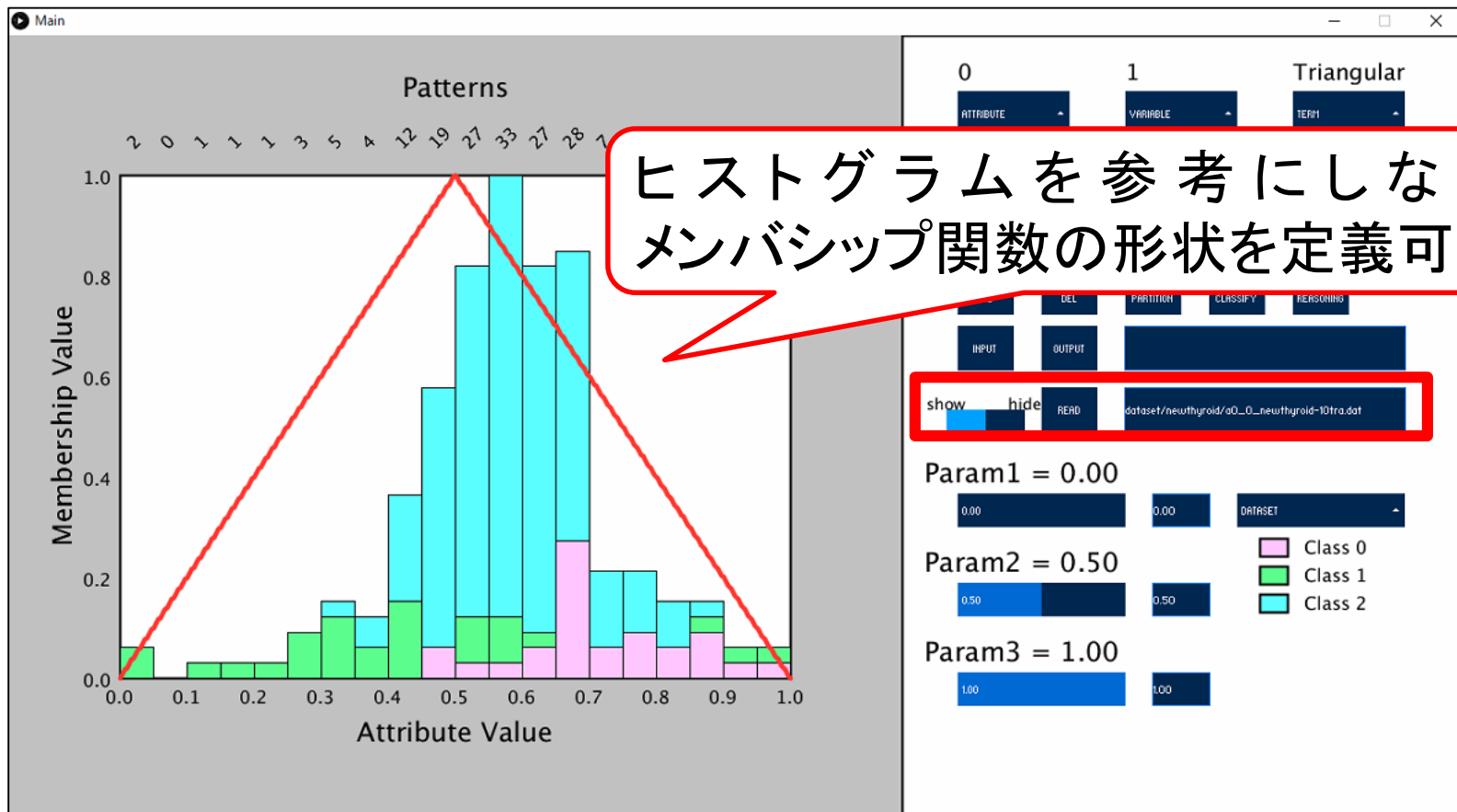
- XMLファイルの入出力機能
- データセットのヒストグラムの表示機能



FMLの導入を支援するGUIツール ユーザビリティ

16

- XMLファイルの入出力機能
- データセットのヒストグラムの表示機能



FMLの導入を支援するGUIツール

詳しい操作方法・その他機能について

17

- XMLファイルの入出力機能
- データセットのヒストグラムの表示機能
- メンバシップ関数の自動生成機能
- 条件部の全組合せファジィ識別器の設計機能

SOFT-CRからダウンロード可能.

以下のダウンロード先のファイルを参照していただくか、FSS2019のポスターセッション・デモにて説明します.

SOFT-CR:

<http://soft-cr.org/jfml>を用いたファジィシステム開発を支援するguiツール/

研究室GitHub:

https://github.com/CI-labo-OPU/GUI_FMLtool.git

ファジィ集合の形状がファジィ識別器に与える影響を調査する数値実験を行った.

等分割ファジィ集合や, GUIツールを用いて主観的に定義したファジィ集合を用意し, これらを用いた条件部の全組合せで構成されるファジィ識別器の識別性能を比較する.

使用したデータセットの詳細

Dataset	Number of Patterns	Number of Attributes	Number of Classes
Newthyroid	193	5	3
Phoneme	4,863	5	2

ファジィ集合の形状がファジィ識別器に与える影響を調査する数値実験を行った.

等分割ファジィ集合や, GUIツールを用いて主観的に定義したファジィ集合を用意し, これらを用いた条件部の全組合せで構成されるファジィ識別器の識別性能を比較する.

If x_1 is small and x_2 is small
If x_1 is small and x_2 is medium
If x_1 is small and x_2 is large
If x_1 is medium and x_2 is small
If x_1 is medium and x_2 is medium
If x_1 is medium and x_2 is large
If x_1 is large and x_2 is small
⋮

数値実験: 等分割ファジィ集合

20

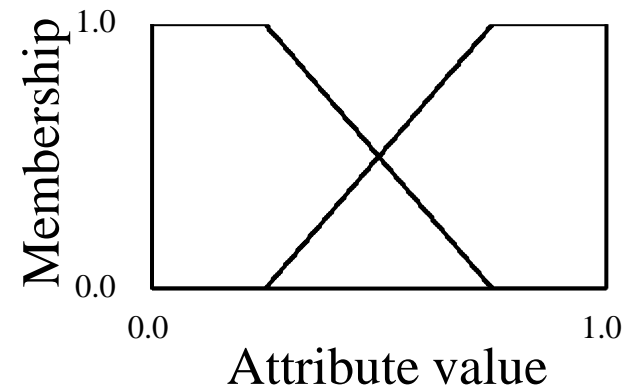
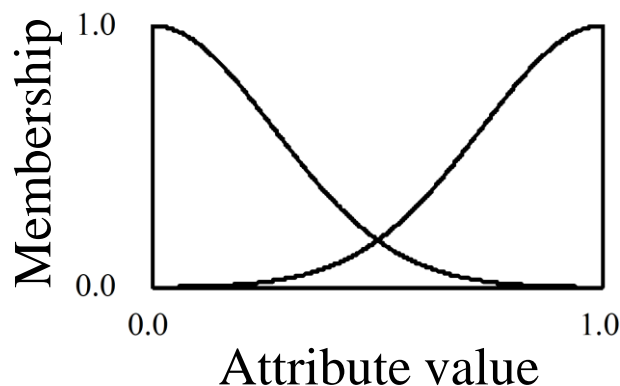
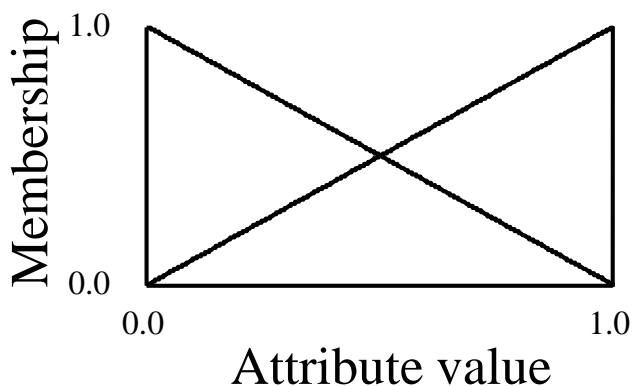
2分割 / 3分割

Triangular

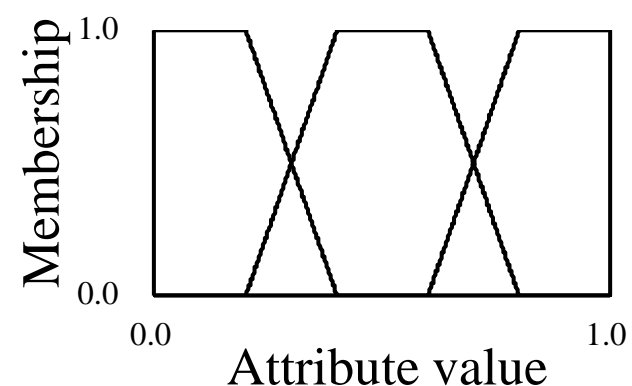
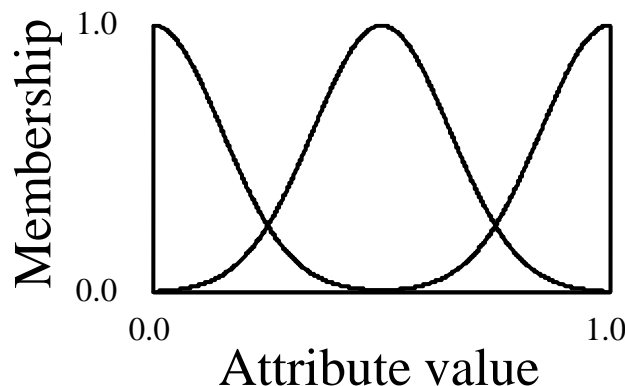
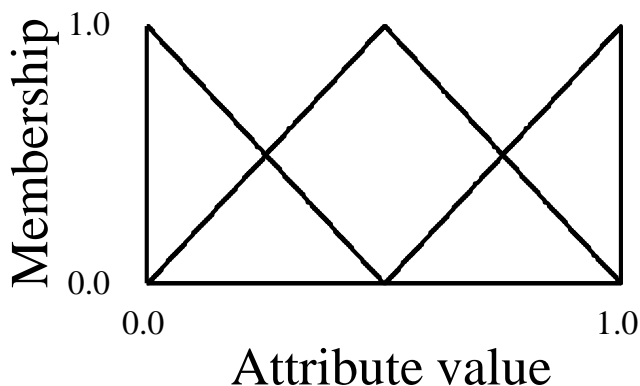
Gaussian

Trapezoid

$K = 2$



$K = 3$



数値実験: 等分割ファジィ集合

21

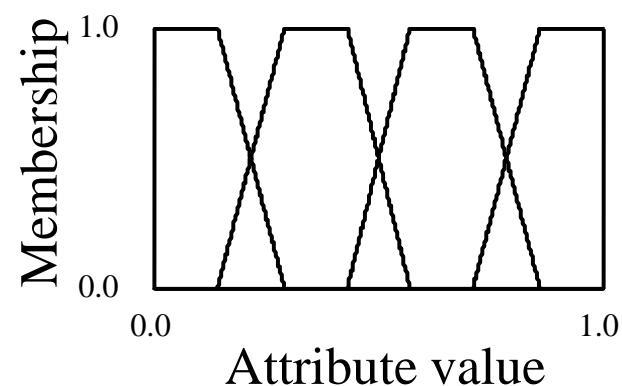
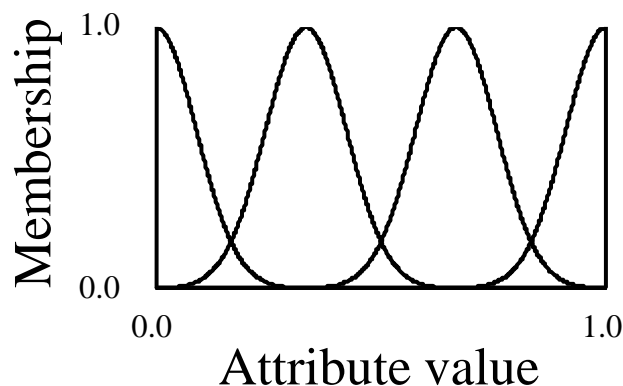
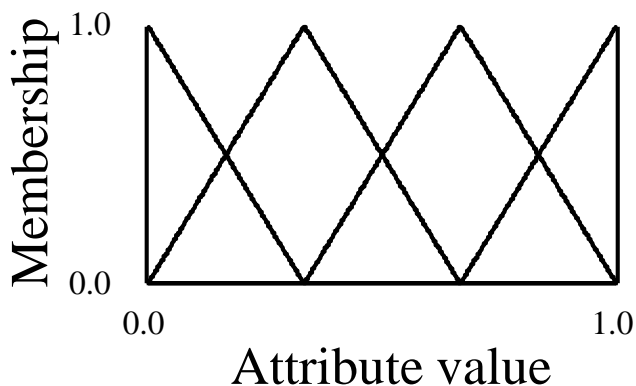
4分割 / 5分割

Triangular

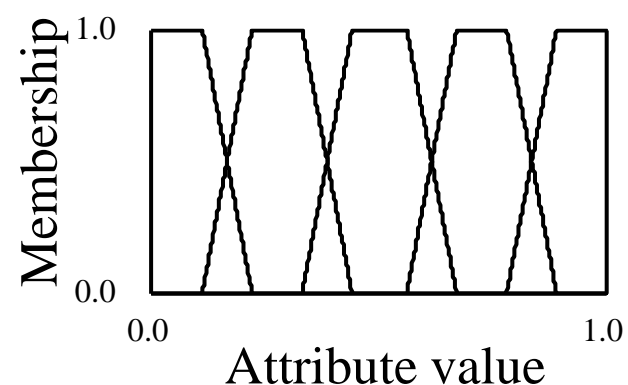
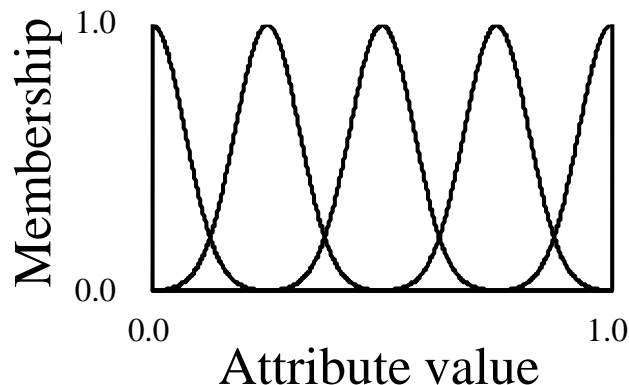
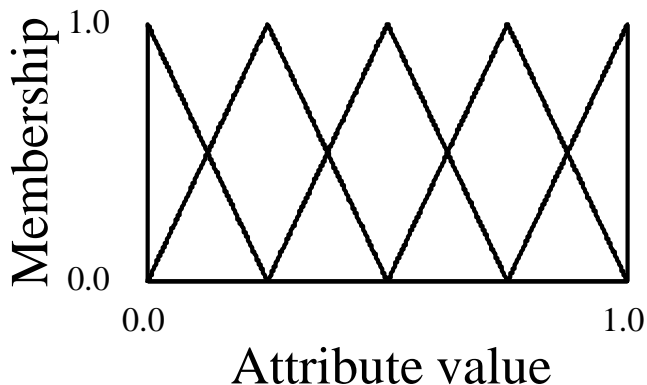
Gaussian

Trapezoid

$K = 4$



$K = 5$



数値実験: 主観的なファジィ集合

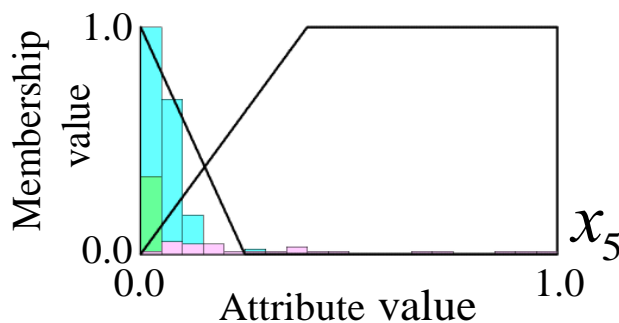
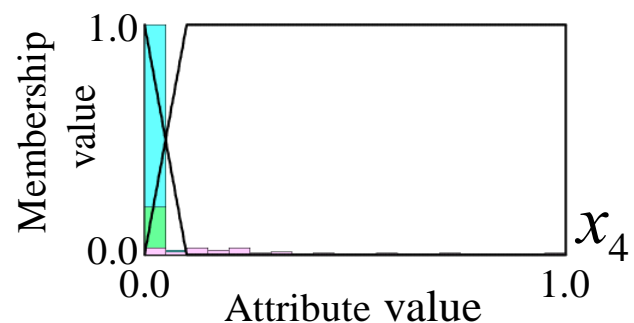
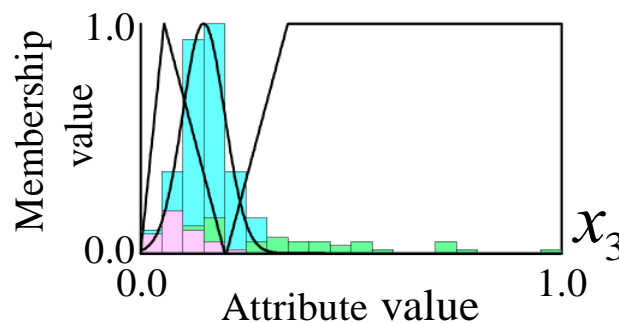
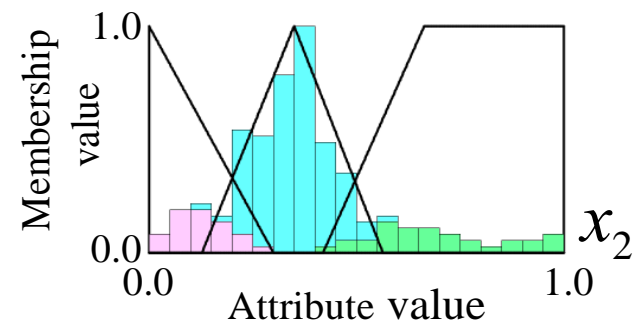
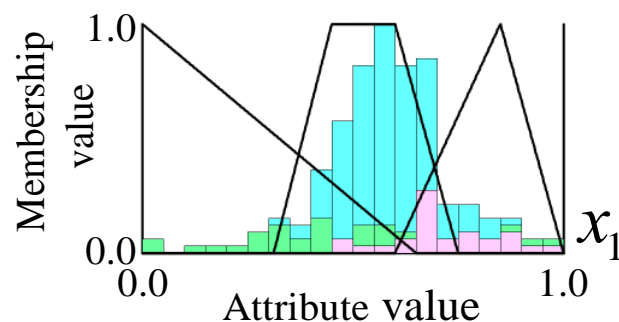
22

Newthyroid (193パターン, 5属性, 3クラス)

Newthyroid

データセットの
分布を参考にし、
筆者が主観的に
定義した.

複数の形状の
ファジィ集合を
使用した.



Class 1
Class 2
Class 3

数値実験: 主観的なファジィ集合

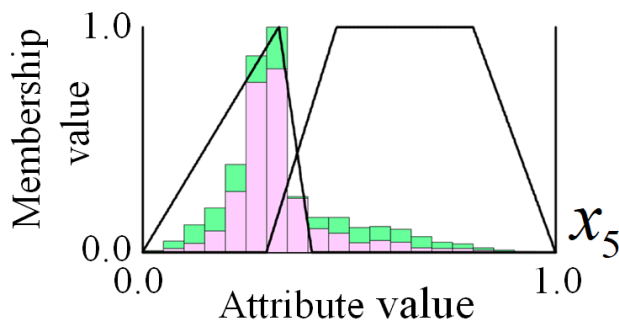
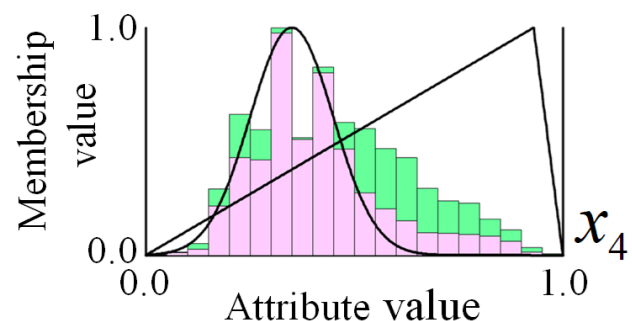
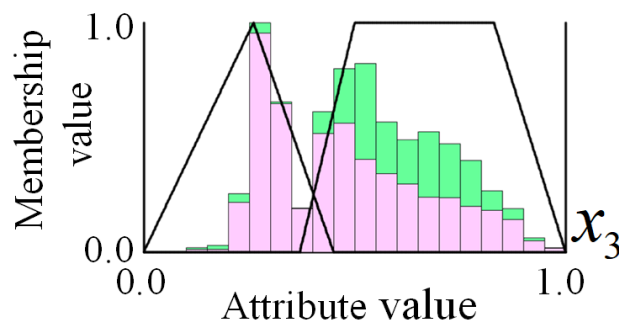
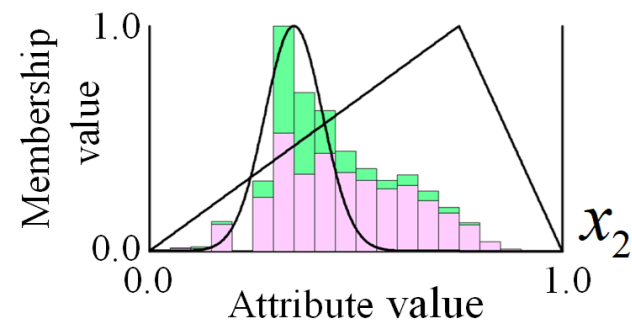
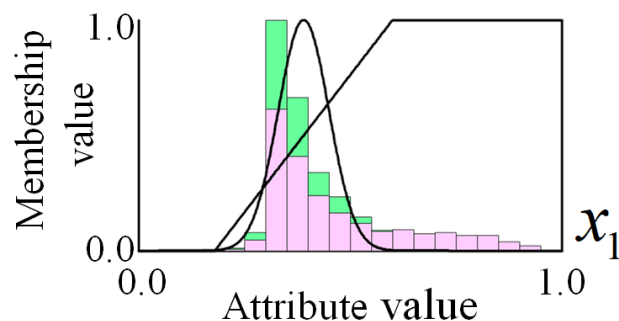
23

Phoneme (4,863パターン, 5属性, 2クラス)

Phoneme

データセットの
分布を参考にし、
筆者が主観的に
定義した。

複数の形状の
ファジィ集合を
使用した。



Class 1

Class 2

数值実験: 結果

誤識別率(%)・ルール数

30回試行平均

赤字: Best

青字: 等分割 < 主観的分割

24

Dataset		Newthyroid			Phoneme		
Shape	K	Train (%)	Test (%)	# of Rules	Train (%)	Test (%)	# of Rules
等分割 Triangular	5	6.41	7.42	3,125	20.48	21.14	3,125
	4	8.60	8.82	1,024	21.74	22.11	1,024
	3	14.09	14.07	243	28.09	28.11	243
	2	22.51	22.75	32	29.35	29.34	32
等分割 Gaussian	5	4.91	6.04	3,125	17.57	18.89	3,125
	4	4.63	6.81	1,024	19.96	20.36	1,024
	3	9.78	10.68	243	28.31	28.61	243
	2	16.06	16.42	32	24.69	24.80	32
等分割 Trapezoid	5	3.82	8.69	3,125	16.60	17.75	3,125
	4	5.06	7.75	1,024	18.91	19.82	1,024
	3	9.08	9.76	243	25.53	25.97	243
	2	18.59	18.74	32	25.74	25.74	32
主観的分割		5.24	6.04	108	24.43	24.38	32

まとめ

- FMLを用いることで、開発環境に依存しないファジィシステムの開発を行うことができる.
- FMLの初学者に対して、FMLの導入を支援するGUIツールの開発を行った.

今後の課題

- 条件部ファジィ集合の全組合せで構成されるファジィ識別器以外のファジィ識別器への対応.
- 定義域が $[0, 1]$ でないデータセットの読込みを可能にする.

メンバシップ関数自動生成機能を用いた実験

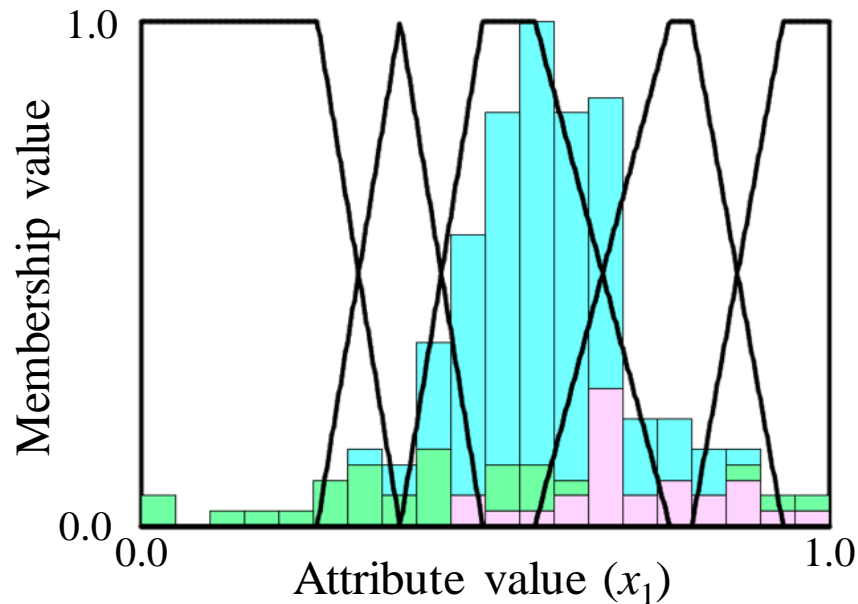
メンバーシップ関数自動生成機能

自動生成したメンバーシップ関数の例

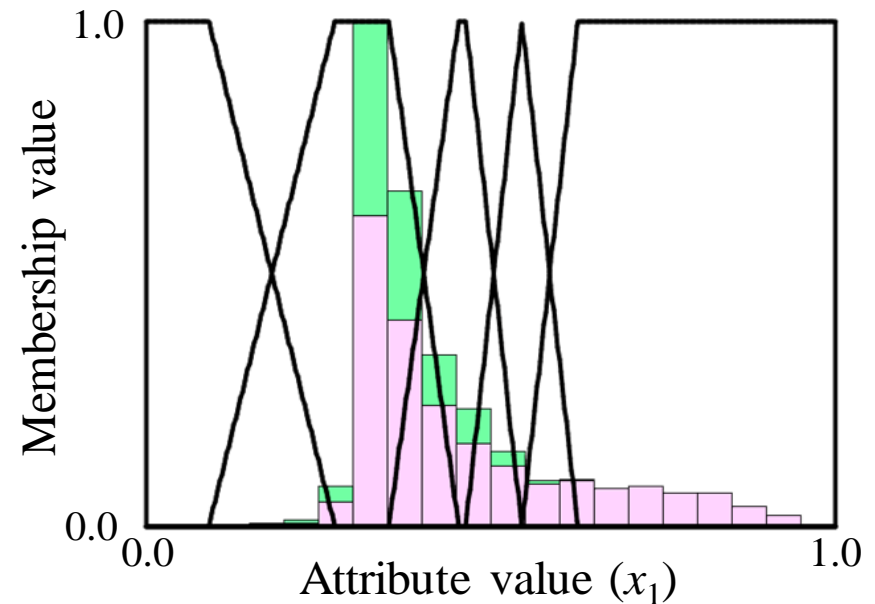
28

ヒストグラムから自動生成したメンバーシップ関数例
($K = 5$ の例)

Newthyroid



Phoneme



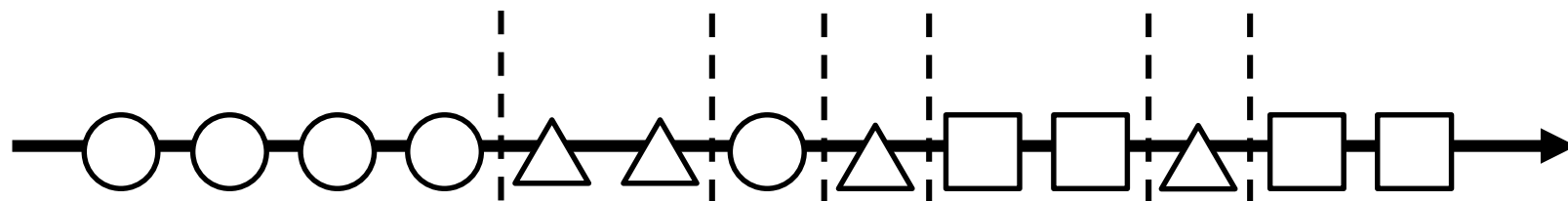
Dataset		Newthyroid		Phoneme	
Shape	K	Train (%)	Test (%)	Train (%)	Test (%)
等分割 Triangular	5	6.41	7.42	20.48	21.14
	4	8.60	8.82	21.74	22.11
	3	14.09	14.07	28.09	28.11
	2	22.51	22.75	29.35	29.34
等分割 Gaussian	5	4.91	<u>6.04</u>	17.57	18.89
	4	4.63	6.81	19.96	20.36
	3	9.78	10.68	28.31	28.61
	2	16.06	16.42	24.69	24.80
等分割 Trapezoid	5	<u>3.82</u>	8.69	<u>16.60</u>	<u>17.75</u>
	4	5.06	7.75	18.91	19.82
	3	9.08	9.76	25.53	25.97
	2	18.59	18.74	25.74	25.74
自動生成	5	0.09	5.57	14.84	16.86
	4	1.74	3.86	16.41	17.86
	3	3.03	5.11	19.00	19.75
	2	7.08	9.14	25.62	25.70

メンバシップ関数自動生成機能

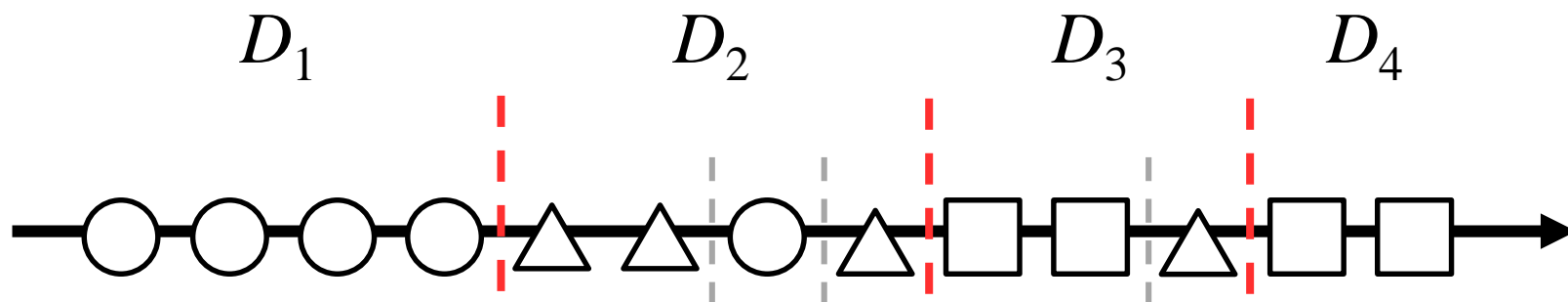
Optimal Splitting Method (分割点生成)

30

1. 分割点の候補となる点を列挙する.

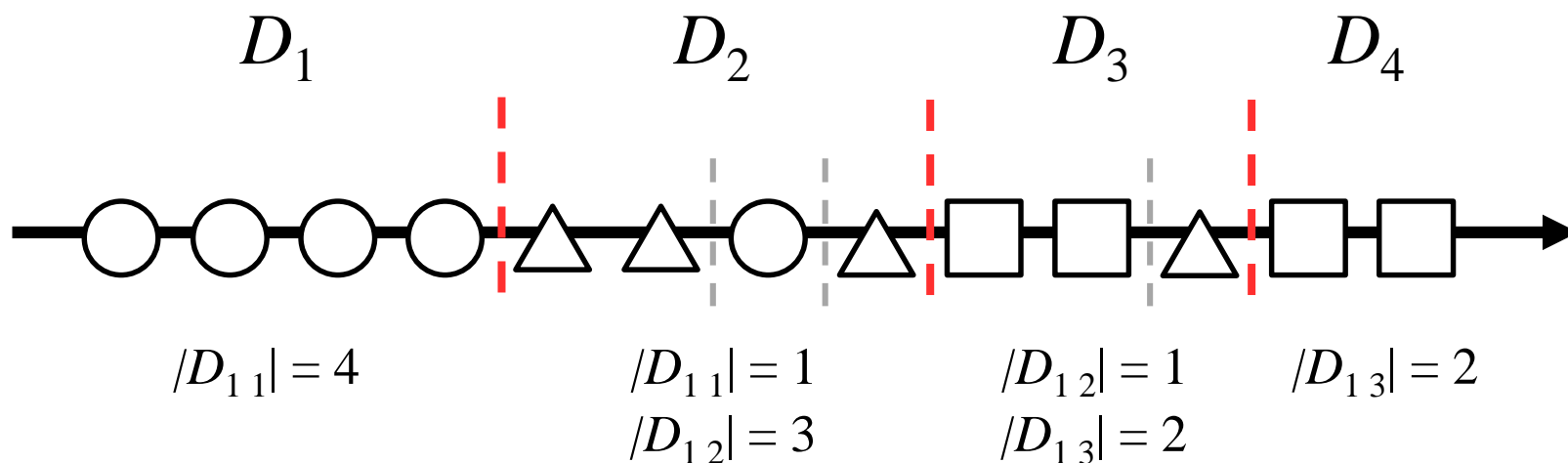


2. $(K-1)$ 個の分割点におけるクラスエントロピーが最小となる組合せを求める.



Optimal Splitting Method (分割点生成)

3. $(K-1)$ 個の分割点におけるクラスエントロピーが最小となる組合せを求める.



クラスエントロピー

$$H(A_1, \dots, A_K) = - \sum_{j=1}^K \frac{|D_j|}{|D|} \sum_{h=1}^M \left(\frac{|D_{jh}|}{|D_j|} \cdot \log_2 \frac{|D_{jh}|}{|D_j|} \right)$$

A_i : 獲得された区間
 M : クラス数

単一勝利ルールについて

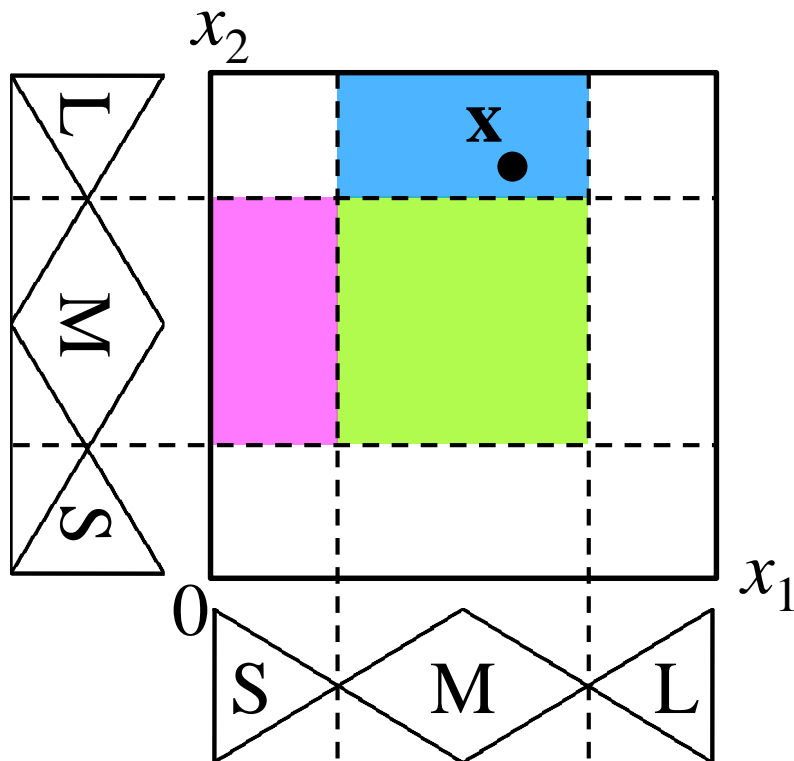
ファジィ識別器

単一勝利ルール戦略

34

入力パターンへの適合度とルール重みの積が最大となるルールを勝利ルールとする.

例: $\mathbf{x} = (0.6, 0.8)$ のクラス推論



ルール集合 (識別器)

R_1 : If x_1 is S and x_2 is M
then Class 1 with 0.5

R_2 : If x_1 is M and x_2 is L
then Class 2 with 0.4

R_3 : If x_1 is M and x_2 is M
then Class 1 with 0.7

ファジィ識別器

単一勝利ルール戦略

35

入力パターンへの適合度とルール重みの積が最大となるルールを勝利ルールとする.

例: $\mathbf{x} = (0.6, 0.8)$ のクラス推論

$$R_1: \mu_S(0.6) \times \mu_M(0.8) \times 0.5 = 0$$

$$R_2: \mu_M(0.6) \times \mu_L(0.8) \times 0.4 = 0.192$$

$$R_3: \underbrace{\mu_M(0.6) \times \mu_M(0.8)}_{\text{適合度}} \times \underbrace{0.7}_{\text{ルール重み}} = 0.224$$

適合度 × ルール重み

R_3 が勝利ルール



識別結果

R_3 の結論部クラス
Class 1

結論部の学習

ファジィ識別器

結論部の学習方法

37

適合度

$$\mu_{\mathbf{A}_q}(\mathbf{x}) = \mu_{A_{q1}}(x_1) \times \mu_{A_{q2}}(x_2) \times \cdots \times \mu_{A_{qn}}(x_n)$$

q 番目のルールのクラス h に対する信頼度

$$c(\mathbf{A}_q \Rightarrow h) = \frac{\sum_{p \in h} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}{\sum_{p=1}^m \mu_{\mathbf{A}_q}(\mathbf{x}_p)}$$

結論部クラス C_q とルール重み CF_q

$$\text{Class } C_q = \max_h \left\{ c(\mathbf{A}_q \Rightarrow \text{Class } h) \right\}$$

$$CF_q = c(\mathbf{A}_q \Rightarrow C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^M \left\{ c(\mathbf{A}_q \Rightarrow \text{Class } h) \right\}$$

R_q : q 番目のルール
 \mathbf{x} : 入力パターン
 \mathbf{A}_q : 条件部ファジィ集合
 C_q : 結論部クラス
 CF_q : ルール重み
 m : パターン数
 M : クラス数

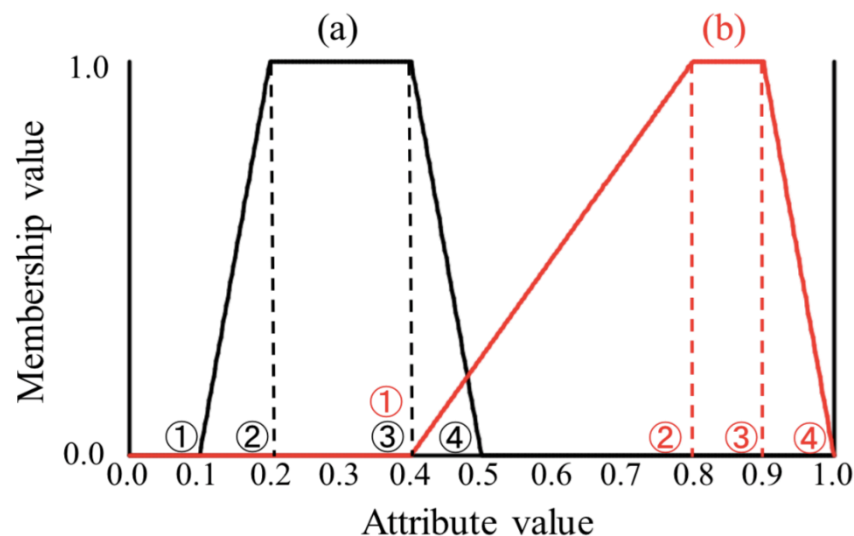
JFMLで実装されているメンバシップ関数の形状

Termにおけるメンバシップ関数の定義

- 様々な形状がJFMLで実装されている
- 複数のパラメータの設定で定義される

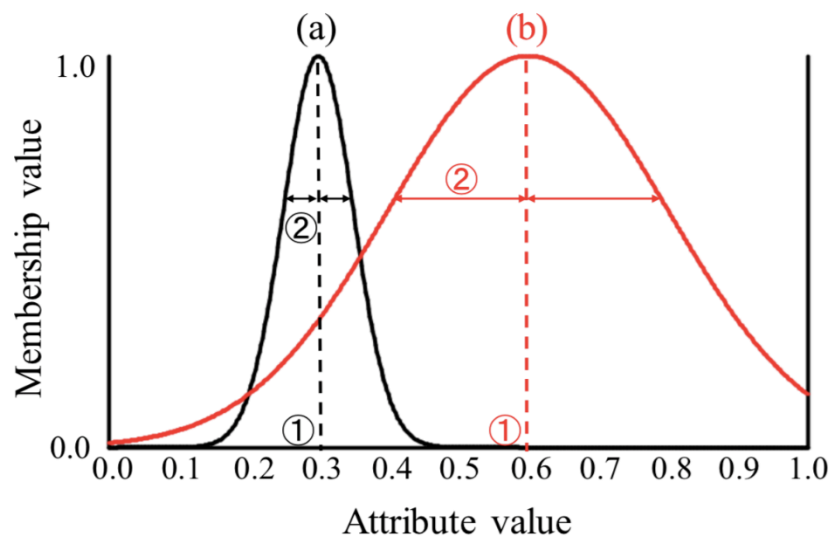
Trapezoid

4点を定義



Gaussian

正規分布の平均と分散



Termにおけるメンバシップ関数の定義

- 様々な形状がJFMLで実装されている
- 複数のパラメータの設定で定義される

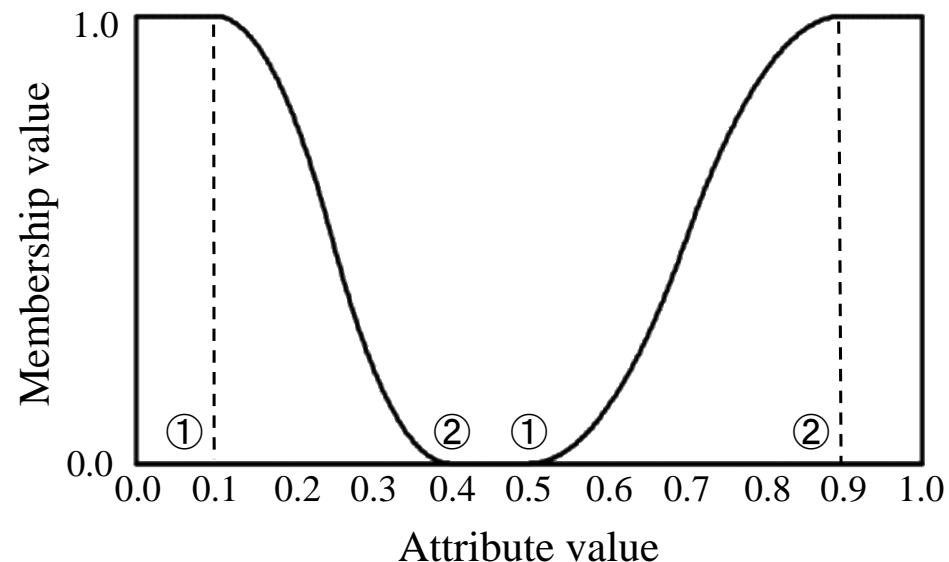
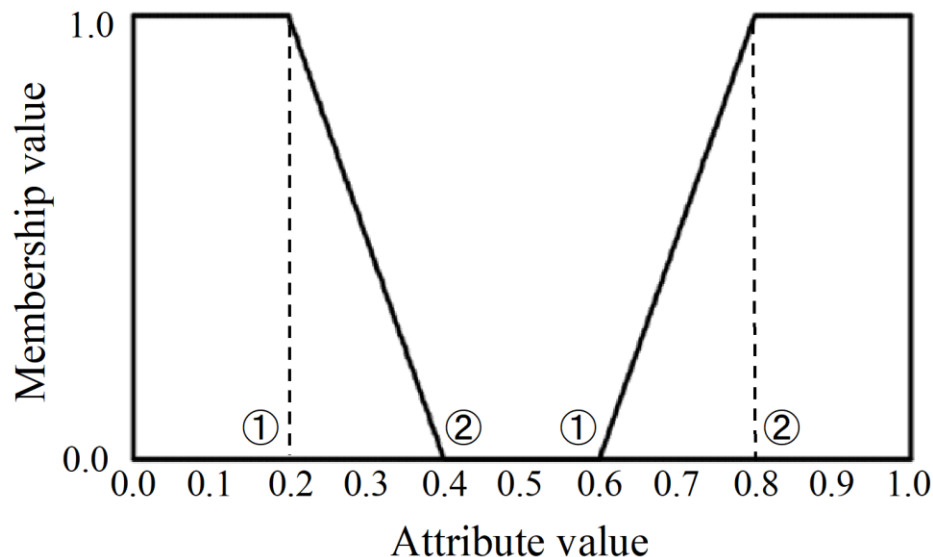
Left / Right Linear

直線の端2点を定義

z-Shape

s-Shape

2点間のスプライン曲線



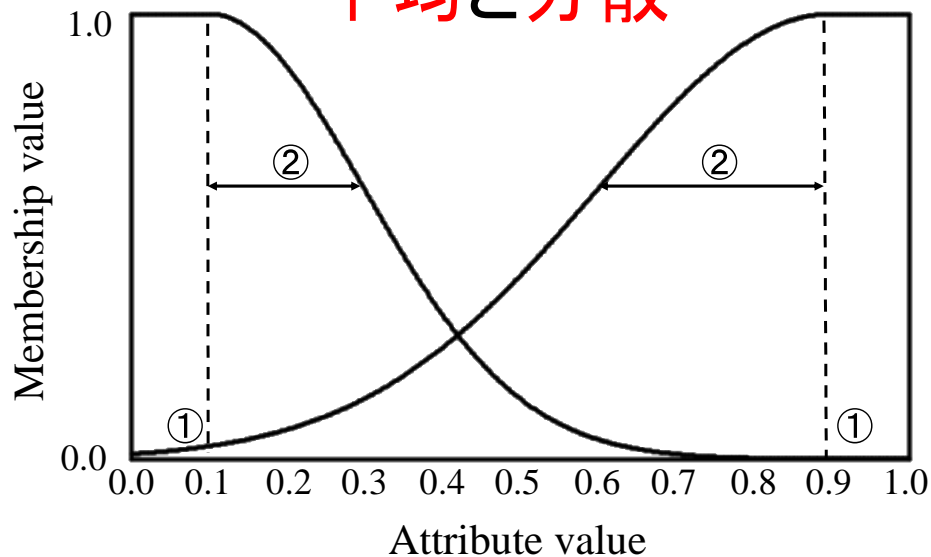
Termにおけるメンバシップ関数の定義

- 様々な形状がJFMLで実装されている
- 複数のパラメータの設定で定義される

Left / Right Gaussian

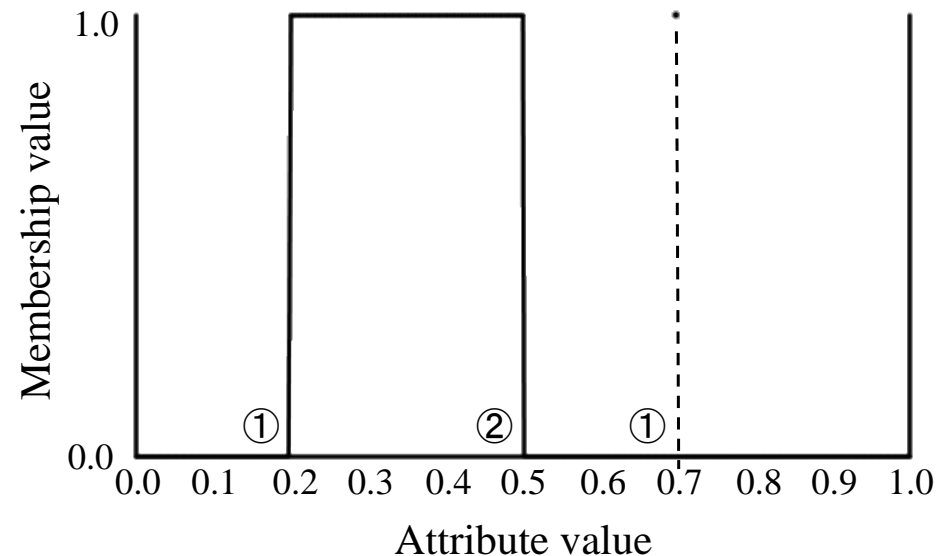
正規分布型の左/右半分

平均と分散



Rectangular

2点間の区間



Singleton

1点

FMLのネスト構造/XML構造について

全体の構造

FML

Knowledge Base

Variable

変数を定義

x_1 is small

Term

ファジィ集合を定義

x_1 is small

Shape 形状を定義

⋮

⋮

Rule Base

Rule

Antecedent

条件部を定義

x_1 is small and ...

Consequent

結論部を定義

Class C

⋮

Fuzzy Markup Language

全体のXML構造 (Knowledge Base)

44

FML

Knowledge Base

Variable

変数を定義

x_1 is small

Term

ファジィ集合を定義

x_1 is small

⋮

Rule Base

```
<fuzzySystem>
  <knowledgeBase>
    <fuzzyVariable name=" $x_1$ ">

      <fuzzyTerm name="small">
        <triangularShape param1=...
      </fuzzyTerm>

      <fuzzyTerm name="medium">
        <triangularShape param1=...
      </fuzzyTerm>

    </fuzzyVariable>
  </knowledgebase>

  <ruleBase>
  </ruleBase>
</fuzzySystem>
```

Fuzzy Markup Language

全体のXML構造 (Knowledge Base)

45

FML

Knowledge Base

Variable

変数を定義

x_1 is small

Term

ファジィ集合を定義

x_1 is small

⋮
⋮

Rule Base

```
<fuzzySystem>
```

```
<knowledgeBase>
```

```
<fuzzyVariable name="x1">
```

```
<fuzzyTerm name="small">  
  <triangularShape param1=...  
</fuzzyTerm>
```

```
<fuzzyTerm name="medium">  
  <triangularShape param1=...  
</fuzzyTerm>
```

```
</fuzzyVariable>
```

```
</knowledgebase>
```

```
<ruleBase>
```

```
</ruleBase>
```

```
</fuzzySystem>
```

Fuzzy Markup Language

46

全体のXML構造 (Rule Base)

FML

Knowledge Base

Rule Base

Rule

Antecedent

条件部を定義

x_1 is small and ...

Consequent

結論部を定義

Class C

⋮

```
<fuzzySystem>
  <knowledgeBase>
  </knowledgebase>

  <ruleBase>
    <rule weight="1.0">
      <antecedent>
        <clause>
          <variable>x1</variable>
          <term>small</term>
        </clause>
        <clause> ... </clause>
      </antecedent>
      <consequent>
      </consequent>
    </rule>
  </ruleBase>
</fuzzySystem>
```

Fuzzy Markup Language

47

全体のXML構造 (Rule Base)

FML

Knowledge Base

Rule Base

Rule

Antecedent

条件部を定義

x_1 is small and ...

Consequent

結論部を定義

Class C

⋮

```
<fuzzySystem>
```

```
<knowledgeBase>
```

```
</knowledgebase>
```

```
<ruleBase>
```

```
<rule weight="1.0">
```

```
<antecedent>
```

```
<clause>
```

```
<variable>x1</variable>
```

```
<term>small</term>
```

```
</clause>
```

```
<clause> ... </clause>
```

```
</antecedent>
```

```
<consequent>
```

```
</consequent>
```

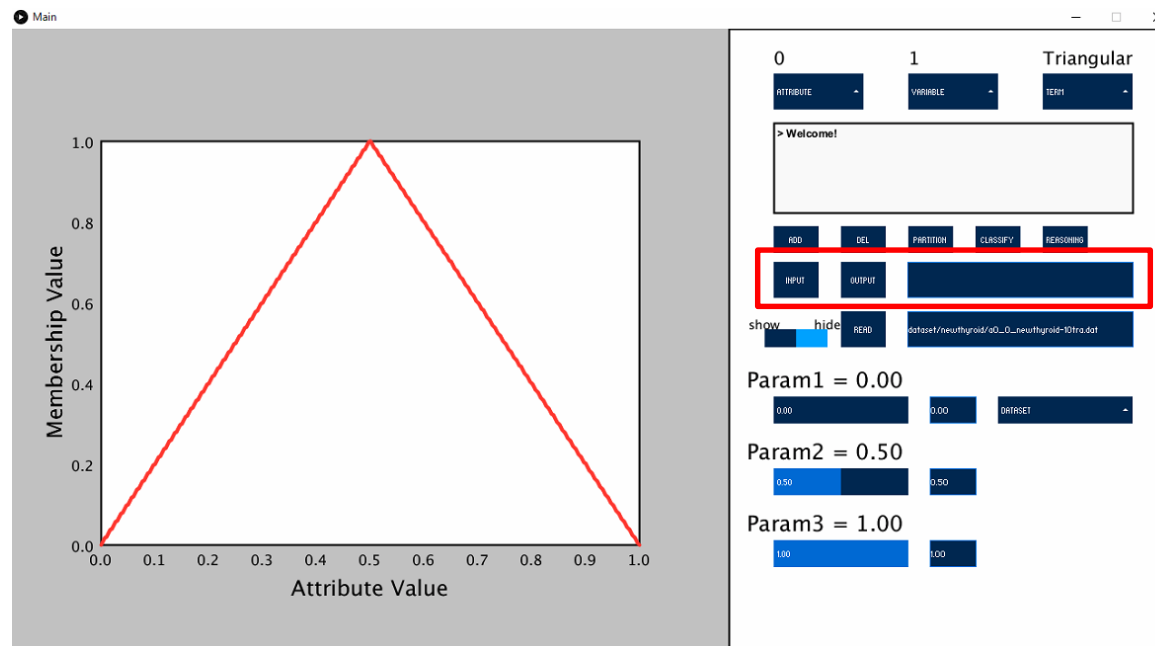
```
</rule>
```

```
</ruleBase>
```

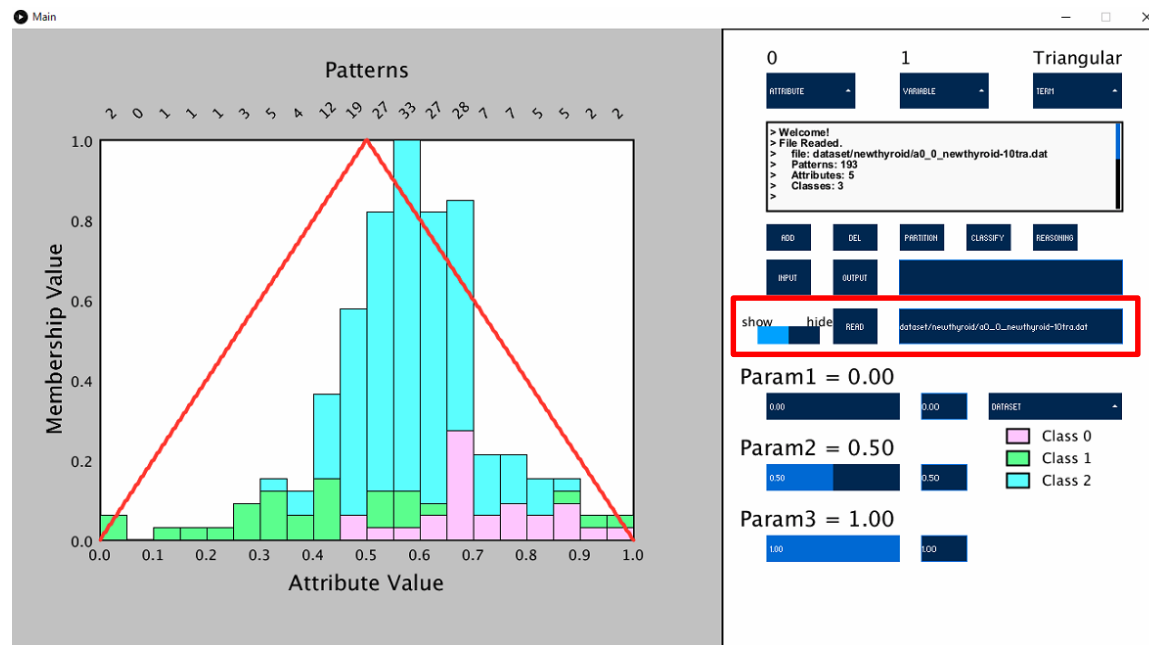
```
</fuzzySystem>
```

GUIツールのユーザビリティ

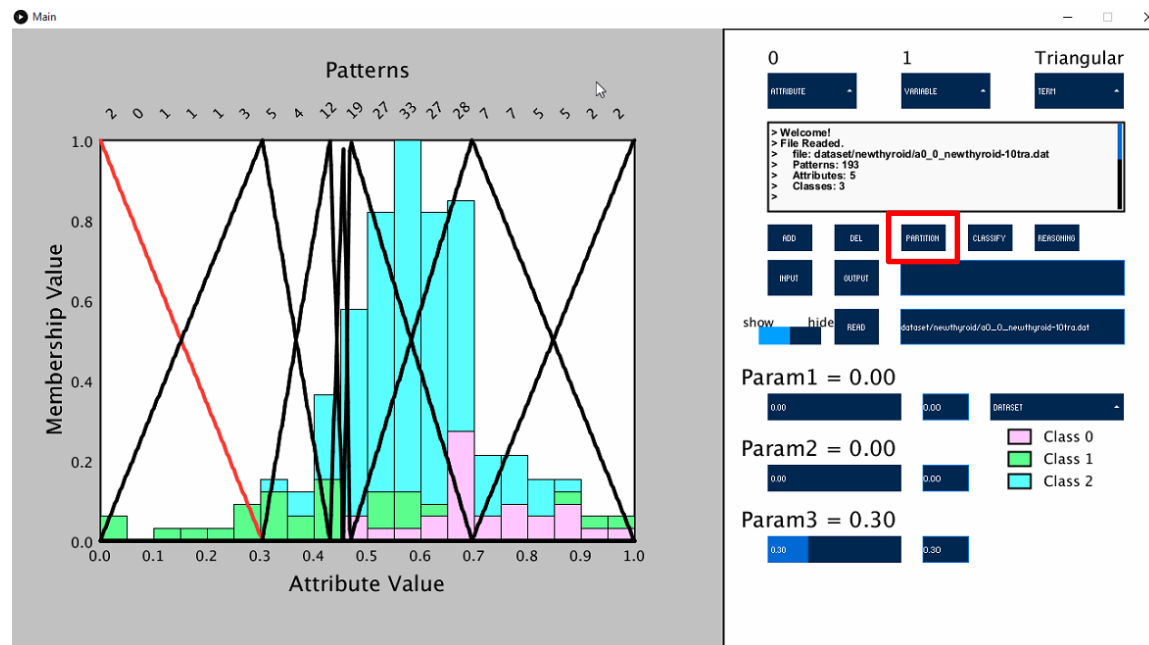
- XMLファイルの入出力機能
- データセットのヒストグラムの表示機能
- メンバシップ関数の自動生成機能
- 条件部の全組合せファジィ識別器の設計機能



- XMLファイルの入出力機能
- データセットのヒストグラムの表示機能
- メンバーシップ関数の自動生成機能
- 条件部の全組合せファジィ識別器の設計機能



- XMLファイルの入出力機能
- データセットのヒストグラムの表示機能
- **メンバーシップ関数の自動生成機能**
- 条件部の全組合せファジィ識別器の設計機能



- XMLファイルの入出力機能
- データセットのヒストグラムの表示機能
- メンバシップ関数の自動生成機能
- 条件部の全組合せファジィ識別器の設計機能

If x_1 is small and x_2 is small

If x_1 is small and x_2 is medium

If x_1 is small and x_2 is large

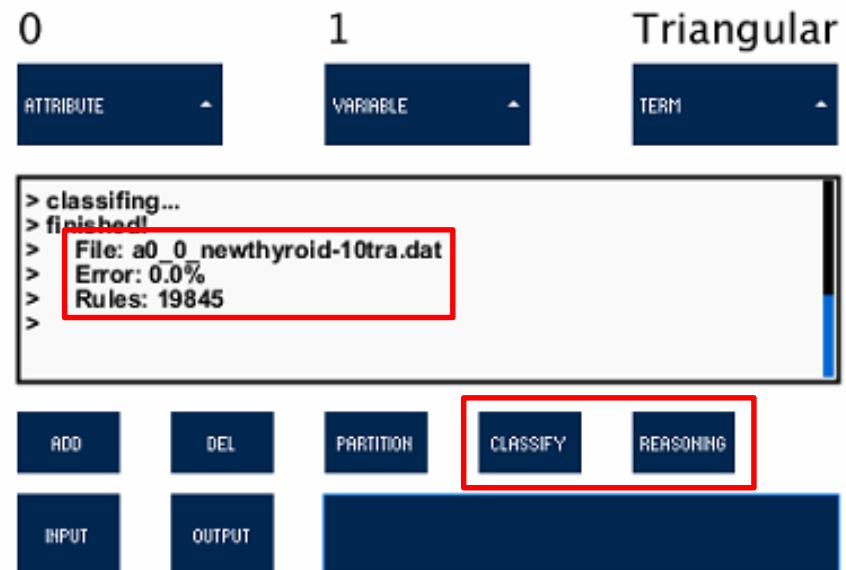
If x_1 is medium and x_2 is small

If x_1 is medium and x_2 is medium

If x_1 is medium and x_2 is large

If x_1 is large and x_2 is small

⋮



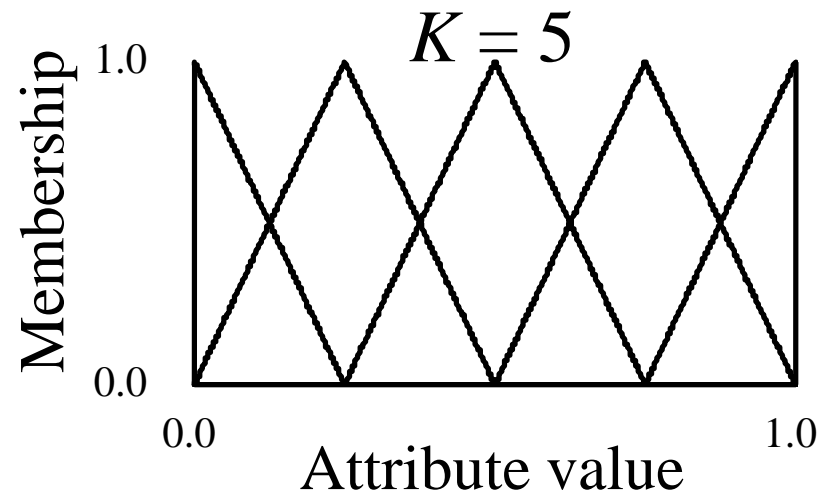
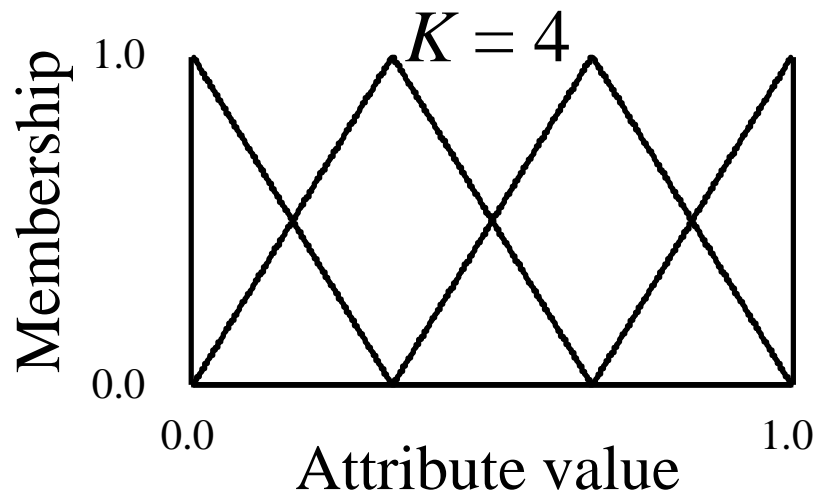
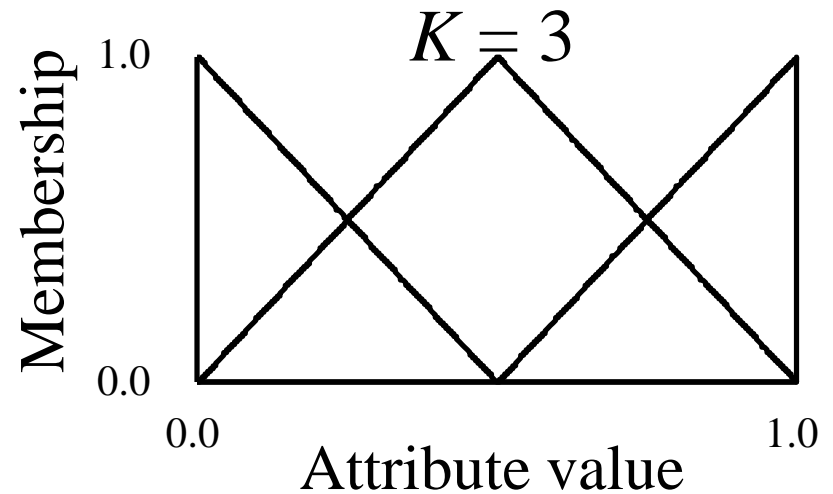
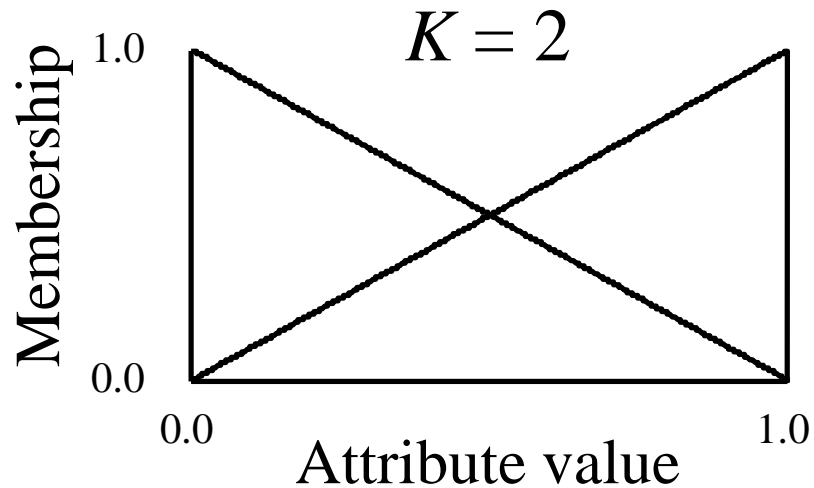
形状別に見た等分割ファジィ集合

数値実験: 等分割ファジィ集合

Triangle

54

2/3/4/5分割の等分割三角型ファジィ集合

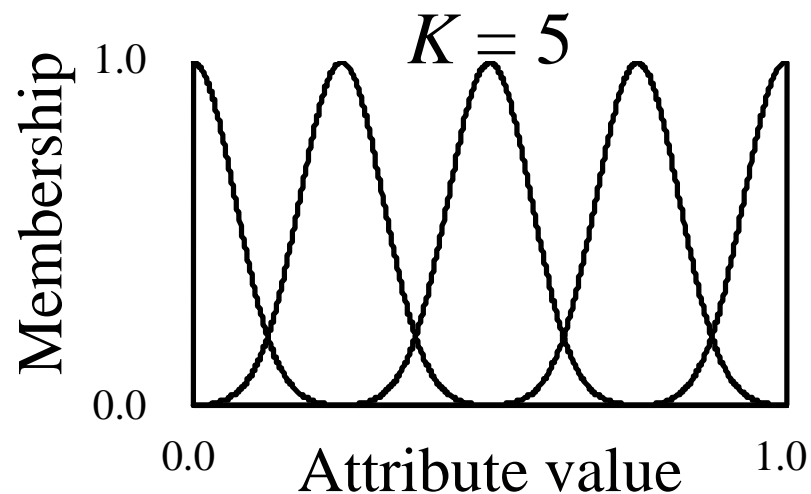
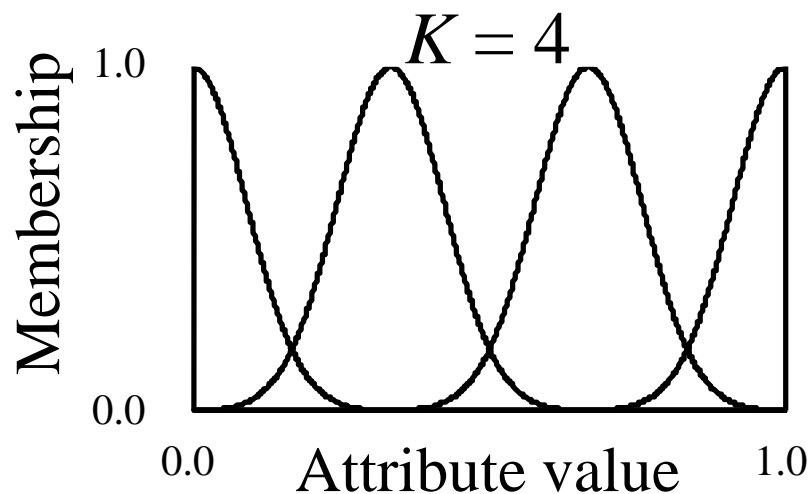
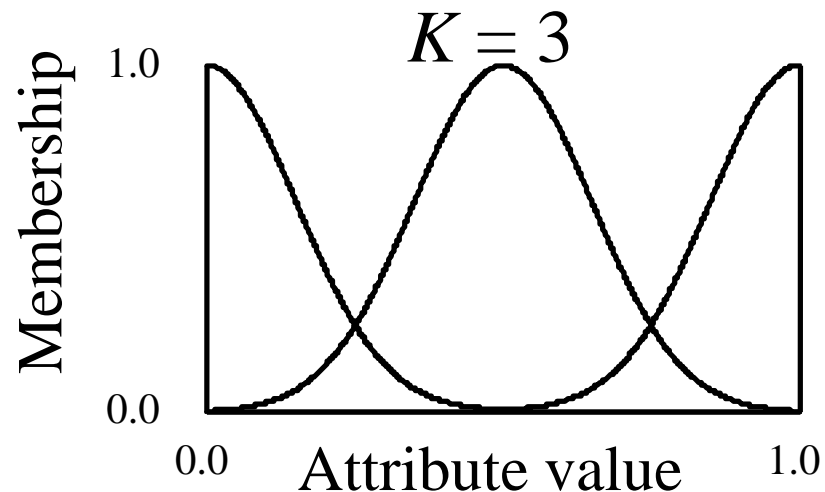
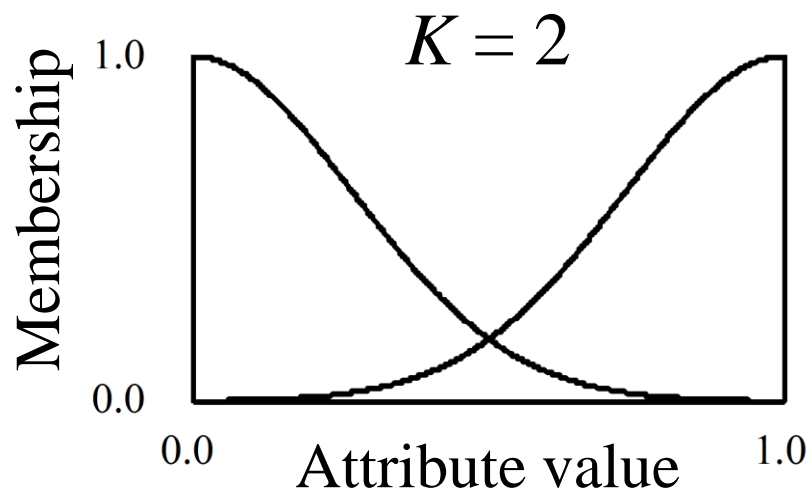


数値実験: 等分割ファジィ集合

55

Gaussian

2/3/4/5分割のガウシアン分布型ファジィ集合

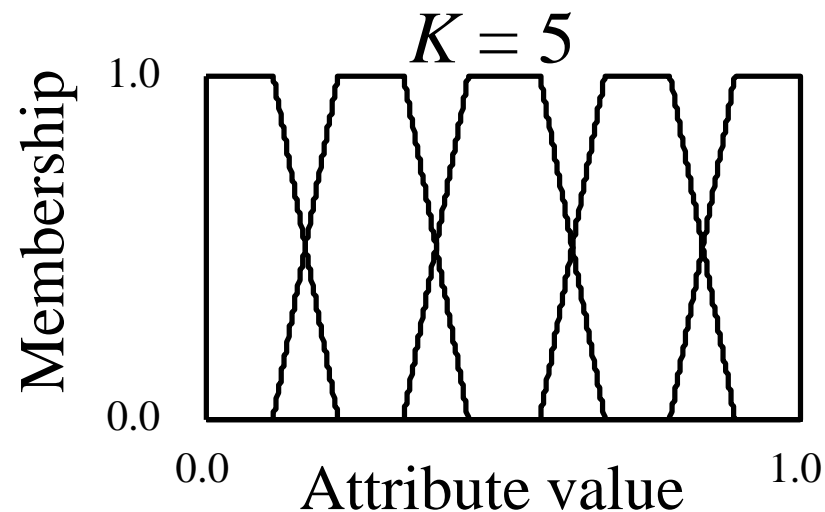
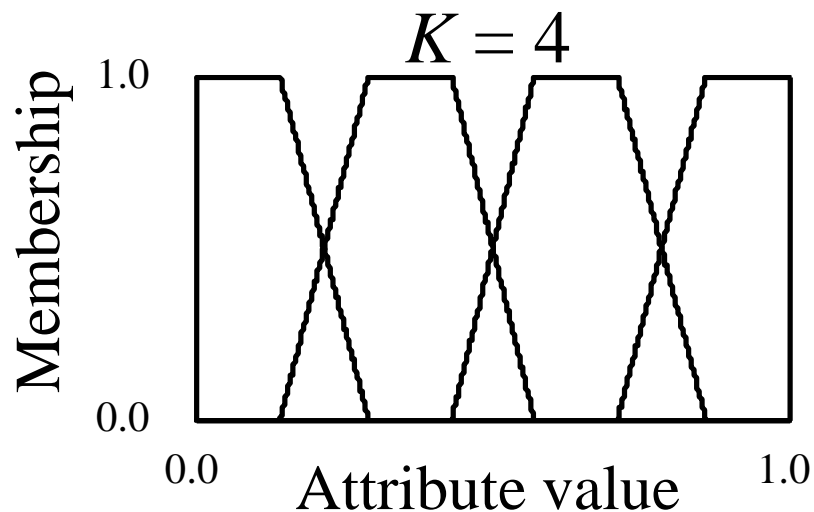
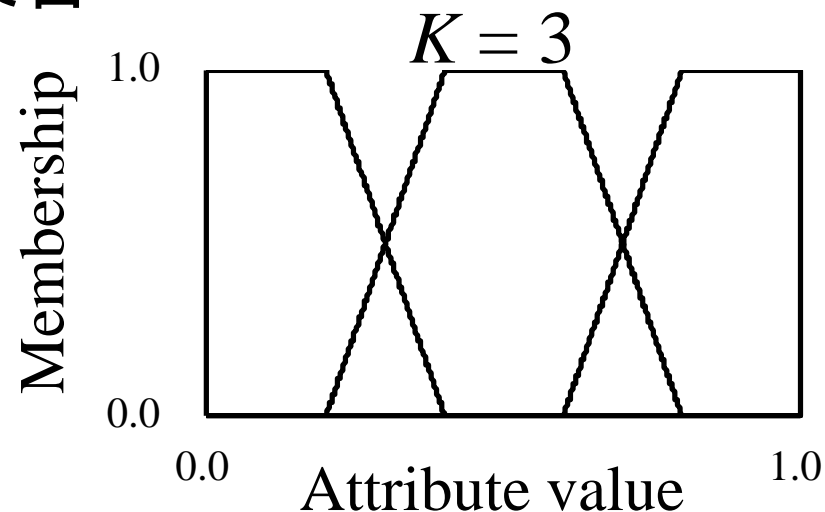
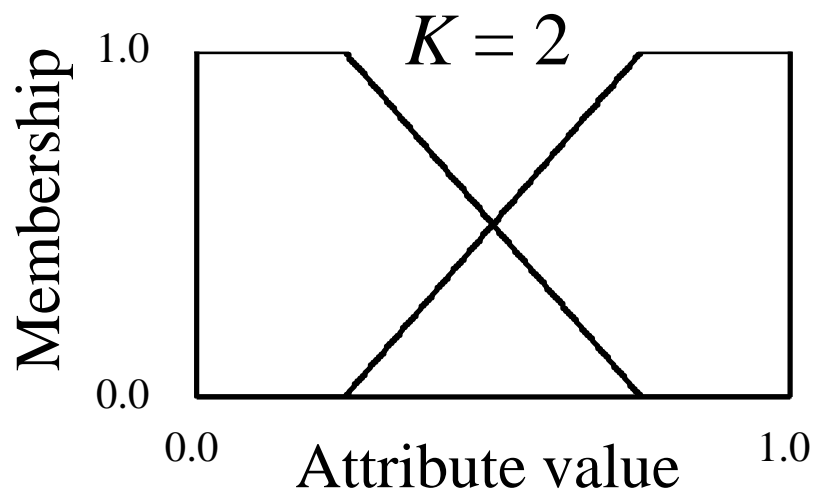


数値実験: 等分割ファジィ集合

56

Trapezoid

2/3/4/5分割の台形型ファジィ集合

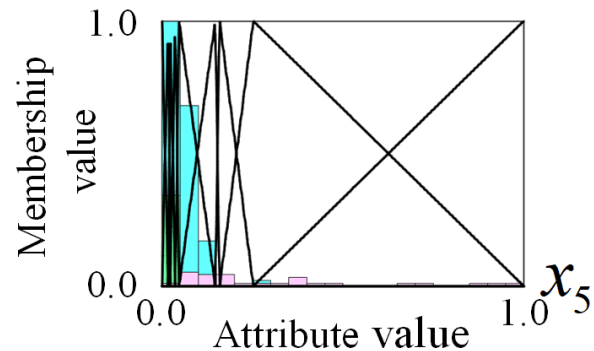
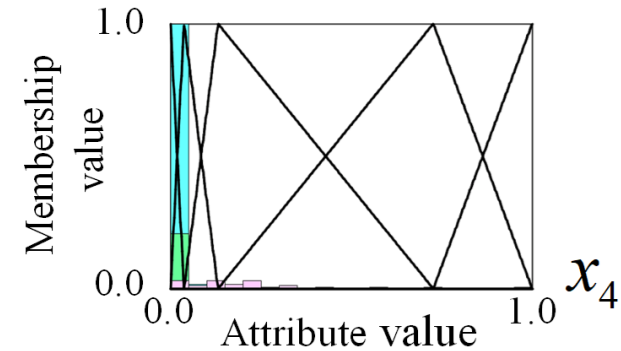
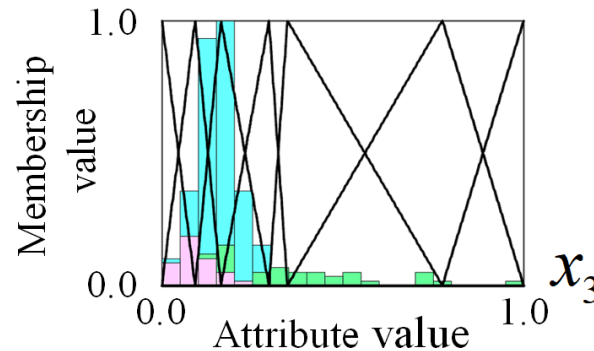
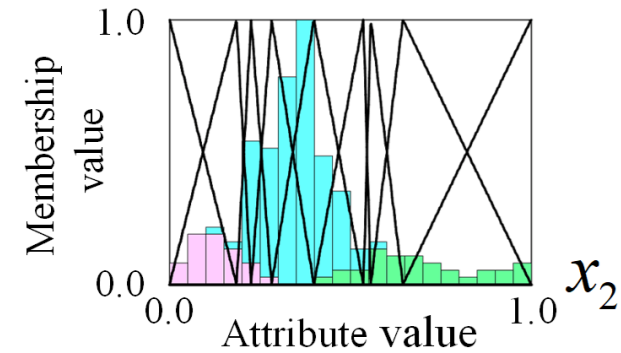
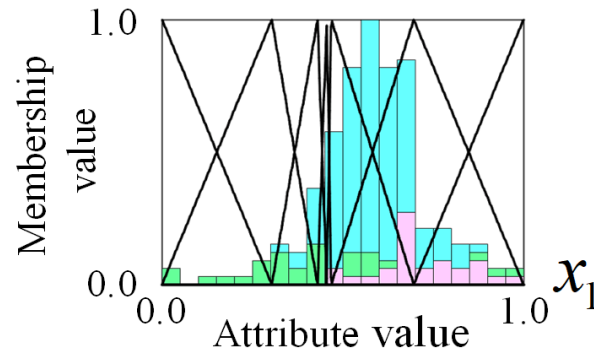


メンバシップ関数自動生成_Algorithm1

Newthyroid (193パターン, 5属性, 3クラス)

Newthyroid

データセットの
分布から自動的に
分割点を獲得.



Class 1
Class 2
Class 3

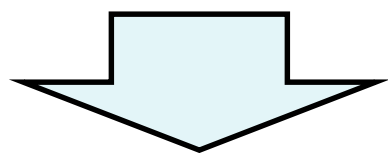
数値実験:自動生成+主観的削減

59

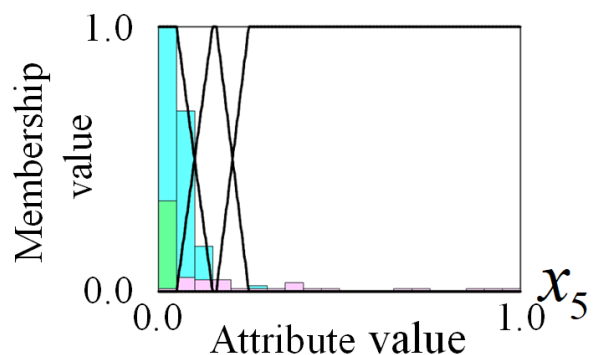
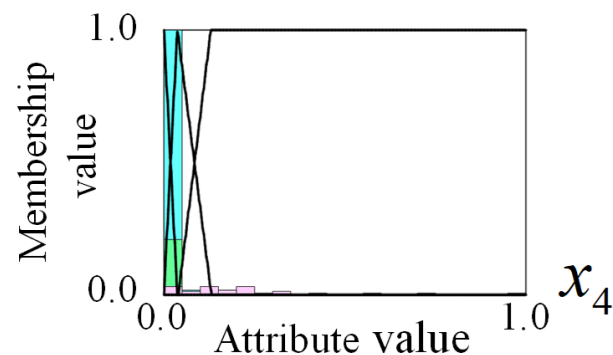
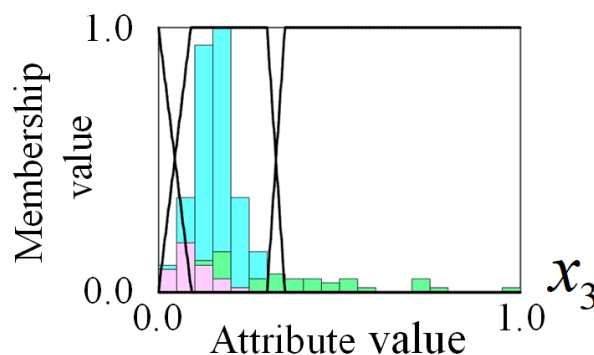
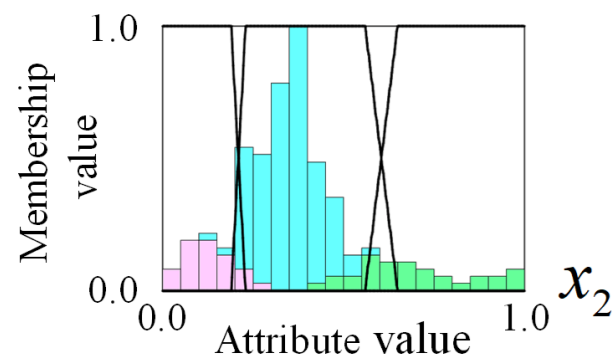
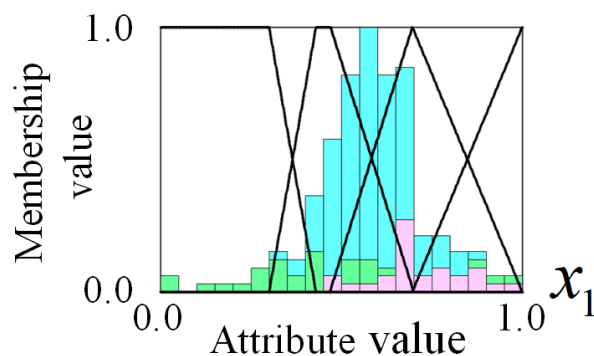
Newthyroid (193パターン, 5属性, 3クラス)

Newthyroid

データセットの
分布から自動的に
分割点を獲得.



複数のファジィ
集合を統合して,
ファジィ集合数を
削減する.



Class 1
Class 2
Class 3

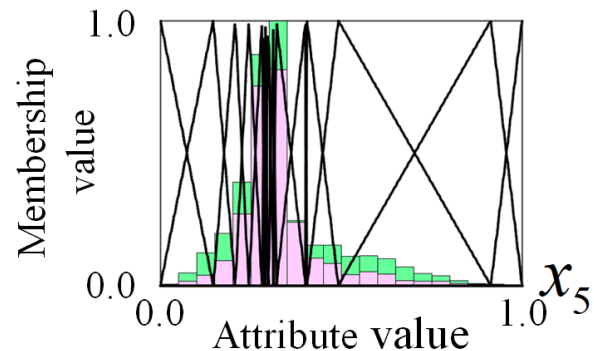
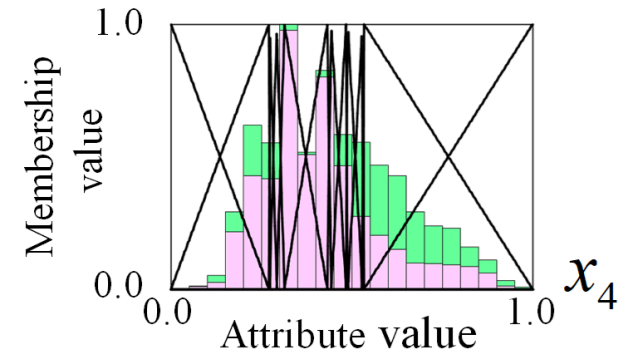
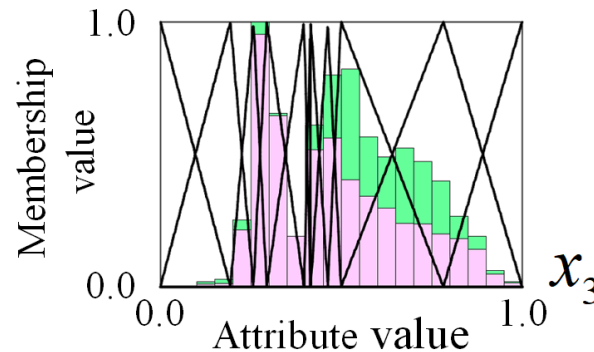
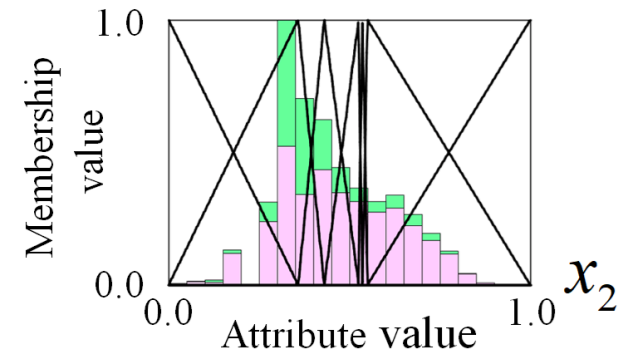
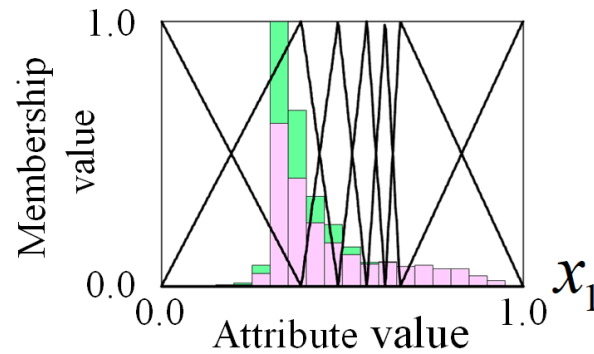
数値実験:自動生成

60

Phoneme (4,863パターン, 5属性, 2クラス)

Phoneme

データセットの
分布から自動的に
分割点を獲得.



Class 1
Class 2

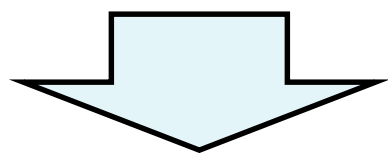
数値実験:自動生成+主観的削減

61

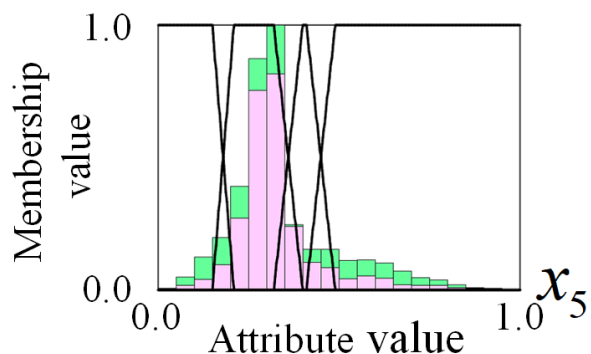
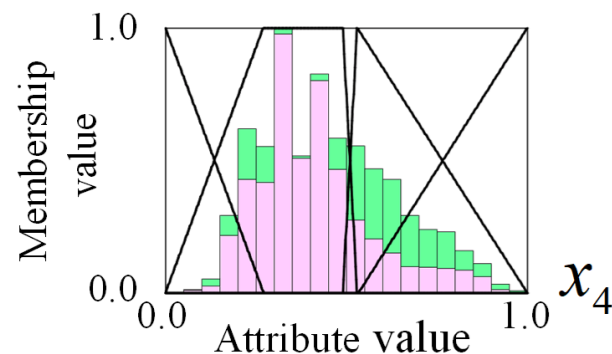
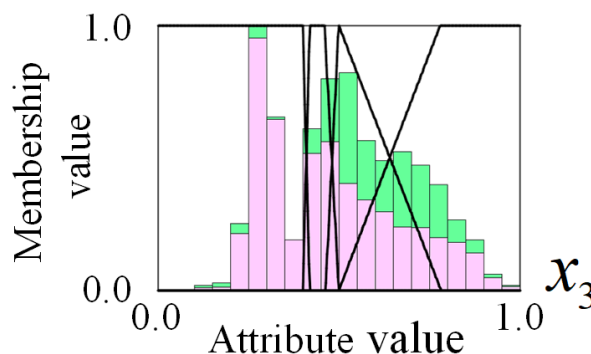
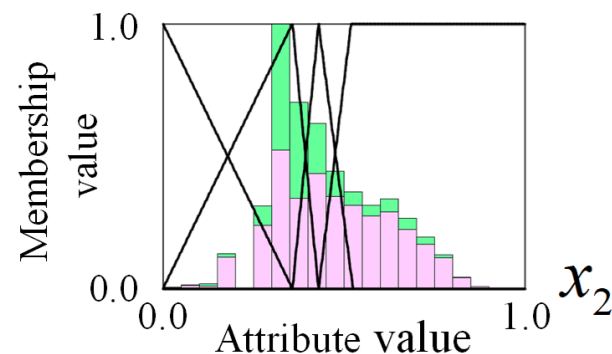
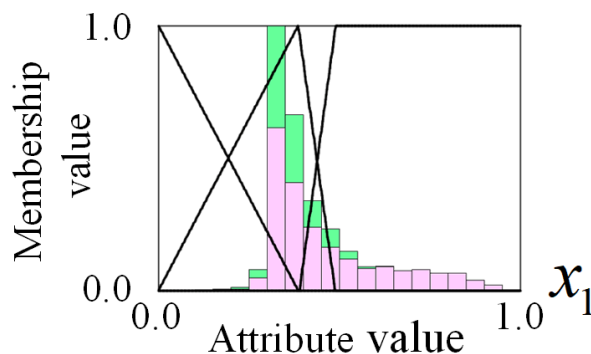
Phoneme (4,863パターン, 5属性, 2クラス)

Phoneme

データセットの
分布から自動的に
分割点を獲得.



複数のファジィ
集合を統合して,
ファジィ集合数を
削減する.



Class 1

Class 2

Dataset		Newthyroid			Phoneme		
Shape	K	Train (%)	Test (%)	# of Rules	Train (%)	Test (%)	# of Rules
等分割 Triangular	5	6.41	7.42	3,125	20.48	21.14	3,125
	4	8.60	8.82	1,024	21.74	22.11	1,024
	3	14.09	14.07	243	28.09	28.11	243
	2	22.51	22.75	32	29.35	29.34	32
等分割 Gaussian	5	4.91	6.04	3,125	17.57	18.89	3,125
	4	4.63	6.81	1,024	19.96	20.36	1,024
	3	9.78	10.68	243	28.31	28.61	243
	2	16.06	16.42	32	24.69	24.80	32
等分割 Trapezoid	5	3.82	8.69	3,125	16.60	17.75	3,125
	4	5.06	7.75	1,024	18.91	19.82	1,024
	3	9.08	9.76	243	25.53	25.97	243
	2	18.59	18.74	32	25.74	25.74	32
主觀的分割		5.24	6.04	108	24.43	24.38	32
自動生成		0.00	5.27	19,845	12.41	16.18	119,119
自動生成 +主觀的削減		4.10	6.03	324	16.74	17.62	768

