LECTURE 11

21CSE451T

FUZZY LOGIC AND ITS APPLICATIONS

Unit 4: Fuzzy Classification and Pattern Recognition

4.6	Classification Metric
4.7	Similarity Relations from Clustering
4.8	Feature Analysis
4.9	Partitions of the Feature Space

Virupakshan K

+91-9840847872 | VIRU@KART.LA

21CSE451T FUZZY LOGIC AND ITS APPLICATIONS
SYLLABUS

Unit-1 - Introduction	***9 Hour***
The Case for Imprecision, The Utility of Fuzzy Systems, Limitations of Fuzzy Systems, Uncertainty and Information, Fuzzy Sets and Membership, Chance versus Fuzziness - Fuzzy Sets: Fuzzy Set Operations, Properties of Fuzzy Sets, Noninteractive Fuzzy Sets, Alternative Fuzzy Set Operations - Fuzzy Relations: Crisp Relations, Fuzzy Relations, Fuzzy Tolerance and Equivalence Relations, Value Assignments, Problems on fuzzy relation - Membership function – various forms –fuzzification – defuzzification to crisp sets.	
Unit-2 - Logic and Fuzzy Systems	***9 Hour***
classical logic, fuzzy logic, fuzzy systems – Development of Membership functions: membership value assignments, intuition, Inference, rank ordering – Automated Methods for Fuzzy Systems: Definitions, Batch Least Squares Algorithm, Recursive Least Squares Algorithm, Gradient Method, Learning From Example, Modified Learning From Example, Problems on logic and fuzzy systems	
Unit-3 - Rule-Base Reduction Methods	***9 Hour***
: Fuzzy Systems Theory and Rule Reduction, Singular Value Decomposition, Combs Method, SVD and Combs Method Examples, problems on SVD and Combs method for rapid inference - Decision Making with Fuzzy Information: Fuzzy Synthetic Evaluation, Fuzzy Ordering, Nontransitive Ranking, Preference and Consensus, Multiobjective Decision Making, Decision Making under Fuzzy States and Fuzzy Actions, problems on decision making with fuzzy information.	
Unit-4 - Fuzzy Classification and Pattern Recognition	***9 Hour***
Classification by Equivalence Relations, Cluster Analysis, Cluster Validity, c-Means Clustering, Fuzzy c-Means, Classification Metric, Similarity Relations from Clustering - Pattern Recognition: Feature Analysis, Partitions of the Feature Space, Single-Sample Identification, Multifeature Pattern Recognition, problems on fuzzy classification and pattern recognition, Case Study: Hand written character recognition using fuzzy logic.	
Unit-5 - Fuzzy Control Systems	***9 Hour***
Control System Design Problem, Control (Decision) Surface, Assumptions in a Fuzzy Control System Design, Simple Fuzzy Logic Controllers, Examples of Fuzzy Control System Design, Aircraft Landing Control Problem - Fuzzy Optimization - Fuzzy Linear Regression – problems on fuzzy optimization and regression, Case study: Robot Navigation using fuzzy logic.	

Unit 4: Fuzzy Classification and Pattern Recognition

4.1	Classification by Equivalence Relations
4.2	Cluster Analysis
4.3	Cluster Validity
4.4	c-Means Clustering
4.5	Fuzzy c-Means
4.6	Classification Metric
4.7	Similarity Relations from Clustering
4.8	Feature Analysis
4.9	Partitions of the Feature Space
4.10	Single-Sample Identification
4.11	Multifeature Pattern Recognition
4.12	Problems on fuzzy classification and pattern recognition
4.13	Case Study: Hand written character recognition using fuzzy logic

Lecture 11 Topics

4.6	Classification Metric
4.7	Similarity Relations from Clustering
4.8	Feature Analysis
4.9	Partitions of the Feature Space

4.6.1 Classification Metric

1. Introduction

Classification metrics are used to evaluate the performance of a classification model.

A classifier predicts class labels.

Example:

- Email: spam or not spam
- Patient: disease or no disease
- Image: cat or dog

We need **metrics** to check whether predictions are correct.

4.6.2 Classification Metric

2. Confusion Matrix

For binary classification:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive, TP	False Negative, FN
Actual Negative	False Positive, FP	True Negative, TN

3. Accuracy

Accuracy measures the total correct predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

4.6.3 Classification Metric

4. Precision

Precision tells how many predicted positives are actually positive.

$$Precision = \frac{TP}{TP + FP}$$

5. Recall

Recall tells how many actual positives are correctly identified.

$$Recall = \frac{TP}{TP + FN}$$

4.6.4 Classification Metric

6. F1-Score

F1-score is the harmonic mean of precision and recall.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

7. Specificity

Specificity tells how many actual negatives are correctly identified.

$$Specificity = \frac{TN}{TN + FP}$$

4.6.5 Classification Metric

Worked Example

Suppose a disease classifier gives:

	Predicted Disease	Predicted No Disease
Actual Disease	40	10
Actual No Disease	5	45

So:

- TP = 40
- FN = 10
- FP = 5
- TN = 45

$$Accuracy = \frac{40+45}{40+45+5+10} = \frac{85}{100} = 0.85$$

$$Precision = \frac{40}{40+5} = \frac{40}{45} = 0.889$$

$$Recall = \frac{40}{40+10} = \frac{40}{50} = 0.80$$

$$F1 = \frac{2(0.889)(0.80)}{0.889+0.80} = 0.842$$

4.6.6 Classification Metric

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
                                                                    confusion_matrix
y_true = [1]*50 + [0]*50
y_pred = [1]*40 + [0]*10 + [1]*5 + [0]*45

print("Confusion Matrix:")
print(confusion_matrix(y_true, y_pred))

print("Accuracy:", accuracy_score(y_true, y_pred))
print("Precision:", precision_score(y_true, y_pred))
print("Recall:", recall_score(y_true, y_pred))
print("F1 Score:", f1_score(y_true, y_pred))
```

4.6.7 Classification Metric

Quick Revision Points

- Classification metrics measure model performance.
- Accuracy is useful when classes are balanced.
- Precision is important when false positives are costly.
- Recall is important when false negatives are costly.
- F1-score balances precision and recall.

4.6.8 Classification Metric

Questions and Answers

Q1. What is accuracy?

Accuracy is the ratio of correct predictions to total predictions.

Q2. When is recall important?

Recall is important when missing a positive case is costly, such as disease detection.

Q3. What is F1-score?

F1-score is the harmonic mean of precision and recall.

4.7.1 Similarity Relations from Clustering

1. Introduction

After clustering, we can define similarity between objects.

If two objects <u>belong strongly</u> to the same cluster, they are <u>highly similar</u>.

In fuzzy clustering, similarity can be calculated using membership values.

2. Similarity from Crisp Clustering

In crisp clustering:

$$R(x_i, x_j) = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ are in same cluster} \\ 0, & \text{otherwise} \end{cases}$$

Example:

Object	Cluster
A	C1
B	C1
C	C2

Then:

- $R(A,B) = 1$
- $R(A,C) = 0$

4.7.2 Similarity Relations from Clustering

3. Similarity from Fuzzy Clustering

If objects have fuzzy memberships, **similarity** can be calculated by:

$$R(x_p, x_q) = \sum_{i=1}^{c} \min(u_{ip}, u_{iq})$$

where:

- u_{ip} = membership of object x_p in cluster i
- u_{iq} = membership of object x_q in cluster i

4.7.3 Similarity Relations from Clustering

Worked Example
Membership table:

Object	C1	C2
A	0.8	0.2
B	0.7	0.3
C	0.1	0.9

Similarity between A and B:

$$R(A,B) = \min(0.8, 0.7) + \min(0.2, 0.3)$$

$$R(A,B) = 0.7 + 0.2 = 0.9$$

Similarity between A and C:

$$R(A,C) = \min(0.8, 0.1) + \min(0.2, 0.9)$$

$$R(A,C) = 0.1 + 0.2 = 0.3$$

So **A and B** are <u>more similar </u>than **A and C**.

4.7.4 Similarity Relations from Clustering

```python
import numpy as np

U = np.array([
    [0.8, 0.2],  # A
    [0.7, 0.3],  # B
    [0.1, 0.9]   # C
])

n = U.shape[0]
R = np.zeros((n, n))

for i in range(n):
    for j in range(n):
        R[i, j] = np.sum(np.minimum(U[i], U[j]))

print("Similarity Matrix:")
print(R)
```

Diagram

```
Membership space:

A: C1=0.8, C2=0.2
B: C1=0.7, C2=0.3
C: C1=0.1, C2=0.9

A and B overlap strongly.
A and C overlap weakly.
```

4.7.5 Similarity Relations from Clustering

Quick Revision Points

- Clustering can generate similarity relations.
- Crisp clustering gives similarity 0 or 1.
- Fuzzy clustering gives similarity between 0 and 1.
- Similarity can be computed from membership overlap.
- Higher overlap means higher similarity.

4.7.6 Similarity Relations from Clustering

Questions and Answers

Q1. How is similarity calculated in crisp clustering?
Similarity is 1 if two objects are in the same cluster, otherwise 0.

Q2. How is fuzzy similarity calculated using memberships?
By summing the minimum membership values across all clusters.

Q3. What does high similarity mean?
It means two objects behave similarly or belong strongly to the same clusters.

4.8.1 Feature Analysis

1. Introduction

A feature is a measurable property of an object.

Example for a fruit:

- Color
- Weight
- Shape
- Sweetness
- Size

Feature analysis means studying which features are useful for classification or pattern recognition.

4.8.2 Feature Analysis

2. Types of Features

a. Numerical Features

Values are numbers.

Examples:

- Height
- Weight
- Temperature
- Age

b. Categorical Features

Values are categories.

Examples:

- Color: red, green, yellow
- Gender: male, female
- Shape: round, oval, square

c. Binary Features

Only two values.

Examples:

- Yes/no
- Present/absent
- True/false

4.8.3 Feature Analysis

3. Feature Vector

An **<u>object</u>** can be represented as a feature vector.

Example:

A fruit may be represented as:

$$x\text{=[weight,sweetness,redness]}$$

Suppose: $x\text{=[150,0.8,0.9]}$

This may represent an apple.

4.8.4 Feature Analysis

4. Feature Selection

Feature selection means choosing only important features.

Benefits:

- Reduces computation
- Improves accuracy
- Removes noise
- Makes model simpler

4.8.5 Feature Analysis

5. Feature Extraction

Feature extraction means creating new features from **existing data**.

Example in image recognition:

Original image pixels are converted into:

- Edges
- Curves
- Corners
- Stroke width

4.8.6 Feature Analysis

6. Fuzzy Features

In **fuzzy logic**, features can have degrees.

Example:

A person's height:

Height	Short	Medium	Tall
160 cm	0.7	0.4	0.0
175 cm	0.1	0.8	0.5
190 cm	0.0	0.2	1.0

Worked Example

Classify a fruit using features:

Fruit	Weight	Sweetness	Redness
Apple	150	0.7	0.8
Orange	180	0.6	0.1

A test fruit: x=[160, 0.75, 0.75]

It is **closer to apple** because redness and sweetness are high.

Diagram

```
Object --------------> Feature Extraction --------------> Feature Vector

Fruit image            color, shape, sweetness            [160, 0.75, 0.75]
```

4.8.7 Feature Analysis

```python
import numpy as np

apple = np.array([150, 0.7, 0.8])
orange = np.array([180, 0.6, 0.1])
test = np.array([160, 0.75, 0.75])

d_apple = np.linalg.norm(test - apple)
d_orange = np.linalg.norm(test - orange)

print("Distance to Apple:", d_apple)
print("Distance to Orange:", d_orange)

if d_apple < d_orange:
    print("Class: Apple")
else:
    print("Class: Orange")
```


```python
import numpy as np

apple = np.array([150, 0.7, 0.8])
orange = np.array([180, 0.6, 0.1])
test = np.array([160, 0.75, 0.75])

d_apple = np.linalg.norm(test - apple)
d_orange = np.linalg.norm(test - orange)

print("Distance to Apple:", d_apple)
print("Distance to Orange:", d_orange)

if d_apple < d_orange:
    print("Class: Apple")
else:
    print("Class: Orange")
```

4.8.8 Feature Analysis

Quick Revision Points

- Features describe objects.
- Feature analysis improves classification.
- Feature selection chooses useful features.
- Feature extraction creates new features.
- Fuzzy features allow partial descriptions.

4.8.9 Feature Analysis

Questions and Answers:

Q1. What is a feature?

A feature is a measurable property of an object.

Q2. What is feature selection?

Feature selection is the process of choosing important features.

Q3. Give examples of features in image recognition.

Edges, corners, shapes, color, texture, and stroke width.

4.9.1 Partitions of the Feature Space

1. Introduction

Feature space is the space formed by all features.

Example:

If we use two features:

- Height
- Weight

Then each person is a point in a two-dimensional feature space.

A partition divides this space into regions.

4.9.2 Partitions of the Feature Space

2. Crisp Partition

In a crisp partition, each point belongs to exactly one region.

Example: A student mark classification:

Mark Range	Grade
0–49	Fail
50–69	Average
70–100	Good

A mark of 69 belongs only to Average.

A mark of 70 belongs only to Good.

This sudden boundary may be unrealistic.

4.9.3 Partitions of the Feature Space

3. Fuzzy Partition

In a fuzzy partition, points can partially belong to multiple regions.

Example:

A mark of 69 may belong to:

- Average with degree 0.6
- Good with degree 0.4

This is more flexible.

4. Membership Functions

A membership function maps a feature value to a degree between 0 and 1.

$$\mu_A(x) : X \to [0, 1]$$

4.9.4 Partitions of the Feature Space

5. Triangular Membership Function

$$\mu_A(x) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a < x \le b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \ge c \end{cases}$$

where:

- a = left point
- b = peak
- c = right point

Worked Example

Define fuzzy grade "Good" using triangular function:

a=60, b=80, c=100

For mark x =70:

$$\mu_{Good}(70) = \frac{70-60}{80-60} = \frac{10}{20} = 0.5$$

So 70 belongs to "Good" with degree 0.5.

For mark x =80:

$$\mu_{Good}(80) = 1$$

4.9.5 Partitions of the Feature Space

```python
def triangular_membership(x, a, b, c):
    if x <= a:
        return 0
    elif a < x <= b:
        return (x - a) / (b - a)
    elif b < x < c:
        return (c - x) / (c - b)
    else:
        return 0

marks = [50, 60, 70, 80, 90, 100]

for mark in marks:
    print(mark, triangular_membership(mark, 60, 80, 100))
```


```python
def triangular_membership(x, a, b, c):
    if x <= a:
        return 0
    elif a < x <= b:
        return (x - a) / (b - a)
    elif b < x < c:
        return (c - x) / (c - b)
    else:
        return 0

marks = [50, 60, 70, 80, 90, 100]

for mark in marks:
    print(mark, triangular_membership(mark, 60, 80, 100))
```


```
Membership Degree
1.0 |          /\
    |         /  \
0.5 |        /    \
    |       /      \
0.0 |------/--------\--------
         60    80     100      Marks

       Good fuzzy set
```

4.9.6 Partitions of the Feature Space

Quick Revision Points

- Feature space contains all feature values.
- Partition divides feature space into regions.
- Crisp partition gives sharp boundaries.
- Fuzzy partition gives soft boundaries.
- Membership functions define fuzzy partitions.

4.9.7 Partitions of the Feature Space

Questions and Answers

Q1. What is feature space?

Feature space is the mathematical space formed by the features of data.

Q2. What is a fuzzy partition?

A fuzzy partition allows a point to partially belong to multiple regions.

Q3. Why are fuzzy partitions useful?

They handle uncertainty and gradual boundaries better than crisp partitions.

Unit 4: Fuzzy Classification and Pattern Recognition

4.1	Classification by Equivalence Relations
4.2	Cluster Analysis
4.3	Cluster Validity
4.4	c-Means Clustering
4.5	Fuzzy c-Means
4.6	Classification Metric
4.7	Similarity Relations from Clustering
4.8	Feature Analysis
4.9	Partitions of the Feature Space
4.10	Single-Sample Identification
4.11	Multifeature Pattern Recognition
4.12	Problems on fuzzy classification and pattern recognition
4.13	Case Study: Hand written character recognition using fuzzy logic

Lecture 11 Topics

4.6	Classification Metric
4.7	Similarity Relations from Clustering
4.8	Feature Analysis
4.9	Partitions of the Feature Space

LECTURE 11

21CSE451T

FUZZY LOGIC AND ITS APPLICATIONS

Doubts?

THANK YOU!

Virupakshan K

+91-9840847872 | VIRU@KART.LA