LECTURE 10

21CSE451T

FUZZY LOGIC AND ITS APPLICATIONS

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

Virupakshan K

+91-9840847872 | VIRU@KART.LA

21CSE451T FUZZY LOGIC AND ITS APPLICATIONS
SYLLABUS

Unit	Hours
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
4.1O	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 10 Topics

4.1	Classification by Equivalence Relations
4.2	Cluster Analysis
4.3	Cluster Validity
4.4	c-Means Clustering
4.5	Fuzzy c-Means

Unit 4: Fuzzy Classification and Pattern Recognition

Learning Objectives

After studying this unit, one should be able to:

- Understand how fuzzy logic supports classification and pattern recognition.
- Explain equivalence relations, similarity relations, and fuzzy partitions.
- Apply cluster analysis techniques to group data.
- Understand crisp c-means and fuzzy c-means clustering.
- Evaluate clustering using cluster validity measures.
- Use classification metrics such as accuracy, precision, recall, and F1-score.
- Apply fuzzy logic to single-sample and multifeature pattern recognition.
- Understand a case study on handwritten character recognition using fuzzy logic.

4.1.1 Classification by Equivalence Relations

1. Introduction

Classification means assigning objects into groups or classes.

For example:

Object	Features	Class
Apple	Red, round, sweet	Fruit
Carrot	Orange, long, crunchy	Vegetable
Mango	Yellow, oval, sweet	Fruit

In **classical** classification, an object either belongs to a class or does not.

In **fuzzy** classification, an object can partially belong to many classes.

Example:

A tomato may belong to:

- Fruit with degree 0.6
- Vegetable with degree 0.8

This is useful when boundaries between classes are not clear.

4.1.2 Classification by Equivalence Relations

2. Relation

A **relation** describes how two objects are connected.

If we have a set: $X = \{x_1, x_2, x_3, ..., x_n\}$

A relation R on X tells whether two elements are related.

In crisp logic:

$$R(x_i, x_j) = \begin{cases} 1, & \text{if } x_i \text{ is related to } x_j \\ 0, & \text{otherwise} \end{cases}$$

In fuzzy logic:

$$R(x_i, x_j) \in [0, 1]$$

where:

- 0 means no relation
- 1 means complete relation
- values between 0 and 1 mean partial relation

4.1.3 Classification by Equivalence Relations

3. Equivalence Relation

A relation is called an equivalence relation if it satisfies three properties:

a. Reflexive

Every object is related to itself.

$$R(x_i, x_i) = 1$$

Example:

The similarity of a student with himself is 1.

b. Symmetric

If xi is related to xj, then xj is related to xi.

$$R(x_i, x_j) = R(x_j, x_i)$$

Example:

If A is similar to B by 0.7, then B is similar to A by 0.7.

c. Transitive

If xi is related to xj and xj is related to xk, then xi should be related to xk.

For fuzzy relations, max-min transitivity is often used:

$$R(x_i, x_k) \geq \max_j \min[R(x_i, x_j), R(x_j, x_k)]$$

4.1.4 Classification by Equivalence Relations

4. Equivalence Classes

An equivalence relation divides a set into groups called **equivalence classes**.

Example:

Students are classified based on grade:

Student	Mark	Class
S1	85	High
S2	82	High
S3	45	Low
S4	48	Low

Here:

- S1 and S2 are equivalent.
- S3 and S4 are equivalent.

4.1.5 Classification by Equivalence Relations

5. Fuzzy Equivalence Relation

A fuzzy equivalence relation is a fuzzy relation that is:

- Reflexive
- Symmetric
- Transitive

It allows classification with partial similarity.

4.1.6 Classification by Equivalence Relations

Worked Example

Consider three students with marks:

Student	Marks
A	80
B	75
C	40

Define **similarity** as:

$$R(x_i, x_j) = 1 - \frac{|m_i - m_j|}{100}$$

$$R(A, B) = 1 - \frac{|80 - 75|}{100} = 0.95$$

$$R(A, C) = 1 - \frac{|80 - 40|}{100} = 0.60$$

$$R(B, C) = 1 - \frac{|75 - 40|}{100} = 0.65$$

Similarity matrix:

	A	B	C
A	1.00	0.95	0.60
B	0.95	1.00	0.65
C	0.60	0.65	1.00

A and B are highly similar.

4.1.7 Classification by Equivalence Relations: Python Code

```
import numpy as np

marks = np.array([80, 75, 40])
n = len(marks)

R = np.zeros((n, n))

for i in range(n):
    for j in range(n):
        R[i, j] = 1 - abs(marks[i] - marks[j]) / 100

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

4.1.8 Classification by Equivalence Relations

Simple Diagram

```
Students based on similarity

A ----0.95---- B
 \             /
  \0.60   0.65/
   \         /
      C
```

4.1.9 Classification by Equivalence Relations

Quick Revision Points

- Classification assigns objects to classes.
- Equivalence relation groups similar objects.
- A relation must be reflexive, symmetric, and transitive.
- Fuzzy equivalence relation allows partial similarity.
- Similarity values lie between 0 and 1.

4.1.10 Classification by Equivalence Relations

Questions and Answers

Q1. What is an equivalence relation?

An equivalence relation is a relation that is reflexive, symmetric, and transitive.

Q2. What is the difference between crisp and fuzzy classification?

In crisp classification, an object belongs completely to one class. In fuzzy classification, an object can partially belong to multiple classes.

Q3. What does $R(xi,xj)$ =0.8 mean?

It means object xi and object xj are similar to degree 0.8.

4.2.1 Cluster Analysis

1. Introduction

Cluster analysis is the process of grouping similar data points together.

A cluster is a group of objects that are similar to each other and different from objects in other groups.

Example:

Suppose we have customer data:

Customer	Age	Income
C1	22	25000
C2	24	27000
C3	55	90000
C4	58	85000

Possible clusters:

- Cluster 1: Young, low-income customers
- Cluster 2: Older, high-income customers

4.2.2 Cluster Analysis

2. Why Cluster Analysis?

Cluster analysis is useful in:

- Market segmentation
- Image processing
- Document grouping
- Bioinformatics
- Medical diagnosis
- Pattern recognition

4.2.3 Cluster Analysis

3. Types of Clustering

a. Hard Clustering

Each data point belongs to exactly one cluster.

Example:

Data Point	Cluster
A	Cluster 1
B	Cluster 1
C	Cluster 2

b. Fuzzy Clustering

Each data point can belong to multiple clusters with different membership values.

Example:

Data Point	Cluster 1	Cluster 2
A	0.9	0.1
B	0.6	0.4
C	0.2	0.8

4.2.4 Cluster Analysis

4. Distance Measures

Clustering often uses distance to measure dissimilarity.

a. Euclidean Distance

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

For two-dimensional points:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

b. Manhattan Distance

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

c. Cosine Similarity

Used commonly in text mining.

$$\text{Cosine Similarity} = \frac{x \cdot y}{||x||||y||}$$

4.2.5 Cluster Analysis

Worked Example

Consider four points:

Point	X	Y
A	1	1
B	2	1
C	8	8
D	9	8

Using visual inspection:

- A and B are close.
- C and D are close.
- So we form two clusters.

Cluster 1: A, B
Cluster 2: C, D

Diagram

```
Y
9 |                     C    D
8 |                     *    *
7 |
6 |
5 |
4 |
3 |
2 |
1 |    A    B
0 |    *    *
     ------------------------
     1 2 3 4 5 6 7 8 9    X
```

4.2.6 Cluster Analysis

```python
import numpy as np
from sklearn.cluster import KMeans

X = np.array([
    [1, 1],
    [2, 1],
    [8, 8],
    [9, 8]
])

model = KMeans(n_clusters=2, random_state=0)
model.fit(X)

print("Cluster Labels:", model.labels_)
print("Cluster Centers:", model.cluster_centers_)
```


```python
import numpy as np
from sklearn.cluster import KMeans

X = np.array([
    [1, 1],
    [2, 1],
    [8, 8],
    [9, 8]
])

model = KMeans(n_clusters=2, random_state=0)
model.fit(X)

print("Cluster Labels:", model.labels_)
print("Cluster Centers:", model.cluster_centers_)
```

4.2.7 Cluster Analysis

Quick Revision Points

- Cluster analysis groups similar objects.
- Clustering is usually unsupervised learning.
- Hard clustering gives one cluster per point.
- Fuzzy clustering gives membership degrees.
- Distance measures are important in clustering.

4.2.8 Cluster Analysis

Questions and Answers

Q1. What is a cluster?

A cluster is a group of similar objects.

Q2. What is the difference between classification and clustering?

Classification uses predefined class labels. Clustering finds groups without predefined labels.

Q3. Give two applications of clustering.

Market segmentation and image segmentation.

4.3.1 Cluster Validity

1. Introduction

Cluster validity checks whether the clusters formed are meaningful and good.

A **clustering algorithm** always produces clusters, but those clusters may not always be useful.

Cluster validity answers questions like:

- Are the clusters compact?
- Are the clusters well separated?
- Is the chosen number of clusters correct?

4.3.2 Cluster Validity

2. Good Clustering Properties

A good clustering result should have:

- **a. High Intra-cluster Similarity**

  Objects within the same cluster should be close to each other.

- **b. Low Inter-cluster Similarity**

  Objects in different clusters should be far apart.

4.3.3 Cluster Validity

3. Types of Cluster Validity

a. Internal Validity

Uses only the data itself.

Examples:

- Silhouette score
- Davies-Bouldin index
- Dunn index

b. External Validity

Uses true class labels, if available.

Examples:

- Rand index
- Adjusted Rand index
- Normalized mutual information

c. Relative Validity

Compares different clustering results.

Example:

Trying c =2, c =3, and c =4 clusters and choosing the best.

4.3.4 Cluster Validity

4. Silhouette Score

The silhouette score measures how well a point fits into its assigned cluster.

For a point i:

- $a(i)$ = average distance from point i to other points in the same cluster
- $b(i)$ = smallest average distance from point i to points in another cluster

Formula:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Range:

$$-1 \leq s(i) \leq 1$$

Interpretation:

Silhouette Value	Meaning
Close to 1	Good clustering
Around 0	Overlapping clusters
Close to -1	Wrong clustering

4.3.5 Cluster Validity

5. Davies-Bouldin Index

The Davies-Bouldin index measures average similarity between clusters.

Lower value means better clustering.

6. Dunn Index

The Dunn index is:

$$D = \frac{\text{minimum inter-cluster distance}}{\text{maximum intra-cluster distance}}$$

Higher Dunn index **means better clustering**.

4.3.6 Cluster Validity

Worked Example

Suppose we test different numbers of clusters:

Number of Clusters	Silhouette Score
2	0.72
3	0.58
4	0.44

Best choice: 2 clusters, because it has the highest silhouette score.

4.3.7 Cluster Validity


```
import numpy as np
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

X = np.array([
    [1, 1],
    [2, 1],
    [8, 8],
    [9, 8],
    [1, 2],
    [8, 9]
])

for k in [2, 3, 4]:
    model = KMeans(n_clusters=k, random_state=0)
    labels = model.fit_predict(X)
    score = silhouette_score(X, labels)
    print("k =", k, "Silhouette Score =", score)
```

Diagram

```
Good clustering:

Cluster 1        Cluster 2
* * *            * * *
* *              * *
* * *            * * *

Bad clustering:

Cluster 1 overlaps Cluster 2
* * * * * * *
  * * * * *
* * * * * *
```

4.3.8 Cluster Validity

Quick Revision Points

- Cluster validity checks quality of clustering.
- Good clusters are compact and separated.
- Silhouette score ranges from -1 to 1.
- Higher silhouette score is better.
- Lower Davies-Bouldin index is better.
- Higher Dunn index is better.

4.3.9 Cluster Validity

Questions and Answers

Q1. What is cluster validity?

Cluster validity is the process of evaluating the quality of clustering results.

Q2. What does a silhouette score close to 1 mean?

It means the point is well matched to its own cluster and far from other clusters.

Q3. Which is better: Dunn index high or low?

A high Dunn index is better.

4.4.1 c-Means Clustering

1. Introduction

c-Means clustering is also called **k-means** clustering.

It is a hard clustering method.

Each data point belongs to exactly one cluster.

The goal is to divide data into c clusters.

4.4.2 c-Means Clustering

2. Objective Function

The **c-means algorithm** tries to minimize the total squared distance between data points and their assigned cluster centers.

$$J = \sum_{i=1}^{c} \sum_{x_k \in C_i} ||x_k - v_i||^2$$

where:

- c = number of clusters
- C_i = cluster i
- x_k = data point
- v_i = center of cluster i

4.4.3 c-Means Clustering

3. Algorithm Steps

1. Choose number of clusters c.
2. Randomly initialize cluster centers.
3. Assign each data point to the nearest center.
4. Recalculate cluster centers.
5. Repeat steps 3 and 4 until centers do not change much.

4.4.4 c-Means Clustering

4. Cluster Center Formula

For cluster Ci:

$$v_i = \frac{1}{|C_i|} \sum_{x_k \in C_i} x_k$$

4.4.5a c-Means Clustering

Worked Example

Data points:

Point	Value
A	2
B	4
C	10
D	12

Let c =2.

Initial centers:

- v_1 =2
- v_2 =10

Step 1: Assign Points

Point	Distance to $v1$	Distance to $v2$	Cluster
2	0	8	C1
4	2	6	C1
10	8	0	C2
12	10	2	C2

Step 2: Update Centers

$$v_1 = \frac{2+4}{2} = 3$$

$$v_2 = \frac{10+12}{2} = 11$$

Final clusters:

- Cluster 1: 2, 4
- Cluster 2: 10, 12

4.4.5b c-Means Clustering

Python Code

```
import numpy as np
from sklearn.cluster import KMeans

X = np.array([[2], [4], [10], [12]])

model = KMeans(n_clusters=2, random_state=0)
model.fit(X)

print("Labels:", model.labels_)
print("Centers:", model.cluster_centers_)
```

4.4.5c c-Means Clustering

```python
import numpy as np
from sklearn.cluster import KMeans

X = np.array([[2], [4], [10], [12]])

model = KMeans(n_clusters=2, random_state=0)
model.fit(X)

print("Labels:", model.labels_)
print("Centers:", model.cluster_centers_)
```

Diagram

```
Number line:

2     4           10     12
|-----|-----------|------|
 C1    C1          C2     C2

Centers:
v1 = 3, v2 = 11
```

4.4.6 c-Means Clustering

Quick Revision Points

- c-Means is hard clustering.
- Each data point belongs to only one cluster.
- It minimizes squared distance from centers.
- Cluster centers are calculated using mean.
- It is sensitive to initial centers.

4.4.7 c-Means Clustering

Questions and Answers

Q1. Why is c-means called hard clustering?

Because each data point is assigned to exactly one cluster.

Q2. What is the main objective of c-means?

To minimize the total squared distance between points and cluster centers.

Q3. What is a cluster center?

It is the mean position of all points in a cluster.

4.5.1 Fuzzy c-Means

1. Introduction

Fuzzy c-Means, or FCM, is a fuzzy clustering method.

Unlike c-means, each data point can belong to multiple clusters.

Each point has membership values for all clusters.

Example:

Point	**Cluster 1**	**Cluster 2**
A	0.8	0.2
B	0.5	0.5
C	0.1	0.9

For every point, memberships add up to 1:

$$\sum_{i=1}^{c} u_{ik} = 1$$

4.5.2 Fuzzy c-Means

2. Objective Function

FCM minimizes:

$$J_m = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^{m} ||x_k - v_i||^2$$

where:

- n = number of data points
- c = number of clusters
- u_{ik}= membership of point xk in cluster i
- m = fuzziness parameter
- v_i = center of cluster i

Usually:

m=2

4.5.3 Fuzzy c-Means

3. Cluster Center Update

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m}$$

4. Membership Update

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{||x_k - v_i||}{||x_k - v_j||} \right)^{\frac{2}{m-1}}}$$

4.5.4 Fuzzy c-Means

5. FCM Algorithm Steps

1. Choose number of clusters c.
2. Choose fuzziness parameter m.
3. Initialize membership matrix randomly.
4. Calculate cluster centers.
5. Update membership values.
6. Repeat until memberships or centers become stable.

4.5.5 Fuzzy c-Means

Worked Example

Suppose one-dimensional points:

Point	Value
A	2
B	4
C	10
D	12

Let cluster centers be:

v_1=3,v_2=11

Let m =2.

For point x =4:

Distance to cluster 1:

$$d_1 = |4 - 3| = 1$$

Distance to cluster 2:

$$d_2 = |4 - 11| = 7$$

Membership in cluster 1:

$$u_1 = \frac{1}{1 + \left(\frac{d_1}{d_2}\right)^2} = \frac{1}{1 + \left(\frac{1}{7}\right)^2} = \frac{1}{1 + \frac{1}{49}} = 0.98$$

Membership in cluster 2:

$$u_2 = 1 - 0.98 = 0.02$$

So point 4 strongly belongs to cluster 1.

4.5.6 Fuzzy c-Means

```
import numpy as np

X = np.array([2, 4, 10, 12], dtype=float)
centers = np.array([3, 11], dtype=float)
m = 2

membership = []

for x in X:
    distances = np.abs(x - centers)
    u = []
    for i in range(len(centers)):
        denominator = 0
        for j in range(len(centers)):
            denominator += (distances[i] / distances[j]) ** (2 / (m - 1))
        u.append(1 / denominator)
    membership.append(u)

membership = np.array(membership)

print("Membership Matrix:")
print(membership)
```

Diagram

```
Hard c-means:

Point 4 belongs only to Cluster 1.

Cluster 1: 2, 4
Cluster 2: 10, 12


Fuzzy c-means:

Point 4:
Cluster 1 membership = 0.98
Cluster 2 membership = 0.02
```

4.5.7 Fuzzy c-Means

Quick Revision Points

- FCM is soft clustering.
- Each point has membership values in all clusters.
- Membership values for a point sum to 1.
- Fuzziness parameter m controls softness.
- When m increases, clustering becomes fuzzier.

4.5.8 Fuzzy c-Means

Questions and Answers

Q1. What is fuzzy c-means?

Fuzzy c-means is a clustering method where each point can belong to multiple clusters with membership degrees.

Q2. What is the role of m in FCM?

m controls the fuzziness of clustering. Usually m =2 .

Q3. In FCM, should membership values of one point add to 1?

Yes. The membership values of a point across all clusters should add to 1.

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
4.1O	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 10 Topics

4.1	Classification by Equivalence Relations
4.2	Cluster Analysis
4.3	Cluster Validity
4.4	c-Means Clustering
4.5	Fuzzy c-Means

LECTURE 10

21CSE451T

FUZZY LOGIC AND ITS APPLICATIONS

Doubts?

THANK YOU!

Virupakshan K

+91-9840847872 | VIRU@KART.LA