LECTURE 05

21CSE451T

FUZZY LOGIC AND ITS APPLICATIONS

Unit 2: Logic and Fuzzy Systems

2.6	Inference
2.7	Rank ordering
2.8	Definitions
2.9	Batch Least Squares Algorithm
2.10	Recursive Least Squares Algorithm

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 2: Logic and Fuzzy Systems

#	Topic
2.1	Classical logic
2.2	Fuzzy logic
2.3	Fuzzy systems
2.4	Membership value assignments
2.5	Intuition
2.6	Inference
2.7	Rank ordering
2.8	Definitions
2.9	Batch Least Squares Algorithm
2.10	Recursive Least Squares Algorithm
2.11	Gradient Method
2.12	Learning From Example
2.13	Modified Learning From Example
2.14	Problems on logic and fuzzy systems

Lecture 05 Topics

2.6	Inference
2.7	Rank ordering
2.8	Definitions
2.9	Batch Least Squares Algorithm
2.10	Recursive Least Squares Algorithm

2.6 Inference

Fuzzy inference is the process of deriving output fuzzy values from input fuzzy values using fuzzy rules.

A fuzzy rule is usually written as:

```
IF condition THEN conclusion
```

Example:

```
IF temperature is hot THEN fan speed is high.
```

2.6 Inference: Common Fuzzy Operators

1. Fuzzy AND

Usually represented by minimum:

$$\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

Alternative product form:

$$\mu_{A\cap B}(x) = \mu_A(x) \times \mu_B(x)$$

2. Fuzzy OR

Usually represented by maximum:

$$\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

3. Fuzzy NOT

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

2.6 Inference: Mamdani Inference Method

The **Mamdani method** is one of the most common fuzzy inference methods.

Steps:

1. Fuzzify inputs
2. Apply fuzzy rules
3. Combine rule outputs
4. Defuzzify output

Example

Rules:

Rule 1: IF temperature is hot THEN fan speed is high.

Rule 2: IF temperature is warm THEN fan speed is medium.

Suppose: $\mu_{hot}(x) = 0.7$ and $\mu_{warm}(x) = 0.4$

Then:

- Rule 1 fires with strength 0.7
- Rule 2 fires with strength 0.4

2.6 Inference: Defuzzification

Defuzzification converts fuzzy output into a crisp value.

Centroid Method

The centroid method is the most commonly used method.

$$z^* = \frac{\int z\mu(z)dz}{\int \mu(z)dz}$$

For discrete values:

$$z^* = \frac{\sum z_i\mu(z_i)}{\sum \mu(z_i)}$$

Worked Example

Given output values:

Fan Speed z_i	Membership $\mu(z_i)$
30	0.2
60	0.5
90	0.8

Find crisp output.

$$z^* = \frac{30(0.2) + 60(0.5) + 90(0.8)}{0.2 + 0.5 + 0.8}$$

$$z^* = \frac{6 + 30 + 72}{1.5}$$

$$z^* = \frac{108}{1.5} = 72$$

Therefore: `Fan speed = 72 rpm`

2.7 Rank ordering

Rank ordering is the process of arranging fuzzy alternatives according to their importance, preference, or score.

It is useful in decision-making problems.

Example Applications

- Selecting best supplier
- Ranking job candidates
- Choosing best project
- Medical diagnosis
- Risk analysis

Simple Ranking Method

If alternatives have fuzzy scores, we can rank them using average or weighted score.

Suppose:

Alternative	Membership Score
A	0.75
B	0.60
C	0.90

Ranking:

C > A > B

because: 0.90 > 0.75 > 0.60

2.7 Rank ordering: Weighted Rank Ordering

Weighted Rank Ordering

If criteria have different importance, use:

$$S_i = \sum_{j=1}^{n} w_j r_{ij}$$

where:

- S_i = score of alternative i
- w_j = weight of criterion j
- r_{ij} = rating of alternative i under criterion j

Worked Example

Three students are evaluated using two criteria:

Student	Knowledge	Communication
A	0.8	0.6
B	0.7	0.9
C	0.9	0.5

Weights:

Criterion	Weight
Knowledge	0.6
Communication	0.4

Scores:

For A: S_A =0.6(0.8)+0.4(0.6) =0.48+0.24 =0.72

For B: S_{B-} =0.6(0.7)+0.4(0.9) =0.42+0.36 =0.78

For C: S_C = 0.6(0.9)+0.4(0.5) =0.54+0.20 =0.74

Ranking: B > C > A

2.8 Definitions

1. Universe of Discourse

The complete range of possible values for a variable.

Example: For temperature: X=[0,50]

2. Fuzzy Set

A set in which each element has a degree of membership between 0 and 1.

3. Membership Function

A function that assigns membership value to each element.

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

4. Linguistic Variable

A variable whose values are words instead of numbers.

Example: `Temperature = low, medium, high`

5. Fuzzy Relation

A fuzzy relation represents the degree of relationship between elements of two sets. $R=\{((x,y),\mu_R(x,y))\}$

6. Fuzzy Rule

A rule written using linguistic terms.

Example: `IF speed is high THEN brake pressure is strong.`

7. Fuzzification

Conversion of crisp values into fuzzy values.

8. Defuzzification

Conversion of fuzzy output into crisp output.

9. Normal Fuzzy Set

A fuzzy set is normal if at least one element has membership value 1. $\max_x \mu_A(x) = 1$

10. Convex Fuzzy Set

A fuzzy set is convex if all membership values between two points are not less than the minimum of those two membership values. (where $0 \le \lambda \le 1$)

$$\mu_A(\lambda x_1 + (1-\lambda)x_2) \ge \min[\mu_A(x_1), \mu_A(x_2)]$$

2.9 Batch Least Squares Algorithm

Purpose

The **Batch Least Squares Algorithm** is used to estimate unknown parameters of a model using all available data at once.

It is commonly used in fuzzy systems to tune parameters.

Model Form

Assume a linear model:

$$y = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

In matrix form:

$$Y = X\theta + e$$

where:

Y = output vector

X = input data matrix

θ = parameter vector

e = error vector

Least Squares Objective

Minimize the sum of squared errors:

$$J(\theta) = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

Solution $\theta = (X^T X)^{-1} X^T Y$

Algorithm Steps

1. Collect all input-output data.
2. Form matrix X and vector Y.
3. Compute $X^T X$.
4. Compute inverse of $X^T X$.
5. Calculate parameter vector θ.
6. Use model for prediction.

2.9 Batch Least Squares Algorithm: Worked Example

Given data:

x	y
1	2
2	4
3	6

Assume model: $y = \theta x$

Then:

$$X = \begin{bmatrix}1\\2\\3\end{bmatrix}, \quad Y = \begin{bmatrix}2\\4\\6\end{bmatrix}$$

Formula:

$$\theta = (X^T X)^{-1} X^T Y$$

Calculate:

$$X^T X = 1^2 + 2^2 + 3^2 = 14$$

$$X^T Y = 1(2) + 2(4) + 3(6) = 28$$

$$\theta = \frac{28}{14} = 2$$

So: $y = 2x$

2.9 Batch Least Squares Algorithm: Python Code


```python
import numpy as np

X = np.array([[1], [2], [3]])
Y = np.array([[2], [4], [6]])

theta = np.linalg.inv(X.T @ X) @ X.T @ Y

print("Estimated parameter:", theta)
```

2.10 Recursive Least Squares Algorithm

Purpose

The **Recursive Least Squares**, or RLS, algorithm updates model parameters whenever new data arrives.

Unlike batch least squares, it does not need all data at once.

Why RLS is Useful

RLS is useful when:

- Data arrives continuously
- Online learning is required
- The system changes with time
- Memory is limited

2.10 Recursive Least Squares Algorithm

Model

$$y(k) = \phi^T(k)\theta(k) + e(k)$$

where:

- $y(k)$ = output at time k
- $\phi(k)$ = input vector
- $\theta(k)$ = parameter vector
- $e(k)$ = error

2.10 Recursive Least Squares Algorithm : RLS Equations

Prediction Error

$$e(k) = y(k) - \phi^T(k)\theta(k-1)$$

Gain Vector

$$K(k) = \frac{P(k-1)\phi(k)}{\lambda + \phi^T(k)P(k-1)\phi(k)}$$

Parameter Update

$$\theta(k) = \theta(k-1) + K(k)e(k)$$

Covariance Update

$$P(k) = \frac{1}{\lambda}[P(k-1) - K(k)\phi^T(k)P(k-1)]$$

where:

- $P(k)$ = covariance matrix
- $K(k)$ = gain vector
- λ = forgetting factor
- Usually, $0.95 \leq \lambda \leq 1$

2.10 Recursive Least Squares Algorithm: Algorithm Steps

1. Initialize $\theta(0)$.
2. Initialize covariance matrix $P(0)$ with large values.
3. For each new data sample:
   - Compute prediction error.
   - Compute gain vector.
   - Update parameters.
   - Update covariance matrix.
4. Repeat.

2.10 Recursive Least Squares Algorithm: Python Code

```python
import numpy as np

# Data
x_data = np.array([1, 2, 3, 4, 5])
y_data = np.array([2, 4, 6, 8, 10])

# Initial values
theta = np.array([[0.0]])
P = np.array([[1000.0]])
lam = 1.0

for x, y in zip(x_data, y_data):
    phi = np.array([[x]])
    y_pred = phi.T @ theta
    error = y - y_pred[0, 0]

    K = (P @ phi) / (lam + phi.T @ P @ phi)
    theta = theta + K * error
    P = (1 / lam) * (P - K @ phi.T @ P)

print("Estimated theta:", theta)
```

Lecture 6

2.11 Gradient Method
2.12 Learning From Example
2.13 Modified Learning From Example
2.14 Problems on logic and fuzzy systems

Unit 2: Logic and Fuzzy Systems

#	Topic
2.1	Classical logic
2.2	Fuzzy logic
2.3	Fuzzy systems
2.4	Membership value assignments
2.5	Intuition
2.6	Inference
2.7	Rank ordering
2.8	Definitions
2.9	Batch Least Squares Algorithm
2.10	Recursive Least Squares Algorithm
2.11	Gradient Method
2.12	Learning From Example
2.13	Modified Learning From Example
2.14	Problems on logic and fuzzy systems

Lecture 05 Topics

2.6	Inference
2.7	Rank ordering
2.8	Definitions
2.9	Batch Least Squares Algorithm
2.10	Recursive Least Squares Algorithm

LECTURE 05

21CSE451T

FUZZY LOGIC AND ITS APPLICATIONS

Doubts?

THANK YOU!

Virupakshan K

+91-9840847872 | VIRU@KART.LA