

Интеллектуальные информационные системы (ИИС)

Лекция 7

Нечеткий логический вывод. Применение нечеткой логики в системах управления

Алгоритмы нечеткого вывода (методы FIS)

Алгоритмы нечеткого вывода различаются, главным образом, видом используемых правил, логических операций и разновидностью метода дефазификации. Разработаны модели нечеткого вывода Мамдани, Сугено, Ларсена, Цукамото.

FIS система

В общем случае механизм логического вывода включает четыре этапа: введение нечеткости (фаззификация), нечеткий вывод, композиция и приведение к четкости, или дефаззификация.

Фаззификация – преобразует четкие величине в нечеткие

Дефаззификация – преобразует нечеткие величине в четкие.



Булевой логической вывод

Обычный, булевый логический вывод базируется на следующих тавтологиях:

1) модус поненс: $(A \wedge (A \Rightarrow B)) \Rightarrow B$; если сегодня вторник, то Олег пойдет на работу. Сегодня вторник. Следовательно, Олег пойдёт на работу».

2) модус толленс: $(A \Rightarrow B) \wedge B \Rightarrow A$; Если сегодня идет дождь, то улица мокрая.

Допустим, у нас есть информация о том, что улица не мокрая (то есть B ложно). С использованием модуса толленс мы можем сделать следующий вывод:

- Если сегодня идет дождь, то улица мокрая.
- Улица не мокрая (B ложно).
- Следовательно, сегодня не идет дождь (A ложно).

3) силлогизм: $((A \Rightarrow B) \wedge (B \Rightarrow C)) \Rightarrow (A \Rightarrow C)$; Предпосылка 1: Все люди смертны. Предпосылка 2: Сократ - человек. Заключение: Следовательно, Сократ смертен.

4) контрапозиция: $(A \Rightarrow B) \Rightarrow (B \Rightarrow A)$; Утверждение: Если человек не умеет летать, то он не супергерой. Контрапозиция этого утверждения будет: Если человек не является супергероем, то он умеет летать.

Модус поненс выводит заключение "В есть истинно", если известно, что "А есть истинно" и существует правило "Если А, то В" (А и В – четкие логические утверждения). Однако, если прецедент отсутствует, то модус поненс не сможет вывести никакого, даже приближенного заключения. Даже в случае, когда известно, что близкое к А утверждение А' является истинным, модус поненс не может быть применен. Одним из возможных способов принятия решений при неопределенной информации является применение нечеткого логического вывода.

Нечетким логическим выводом называется получение заключения в виде нечеткого множества, соответствующего текущим значениям входов, с использованием нечеткой базы знаний и нечетких операций

Нечеткий логический вывод Мамдани

Эта система была предложена в 1975 году Эбхасимом Мамдани. По сути, предполагалось управлять комбинацией парового двигателя и котла путем синтеза набора нечетких правил, полученных от людей, работающих в системе.

Имеет следующий вид:

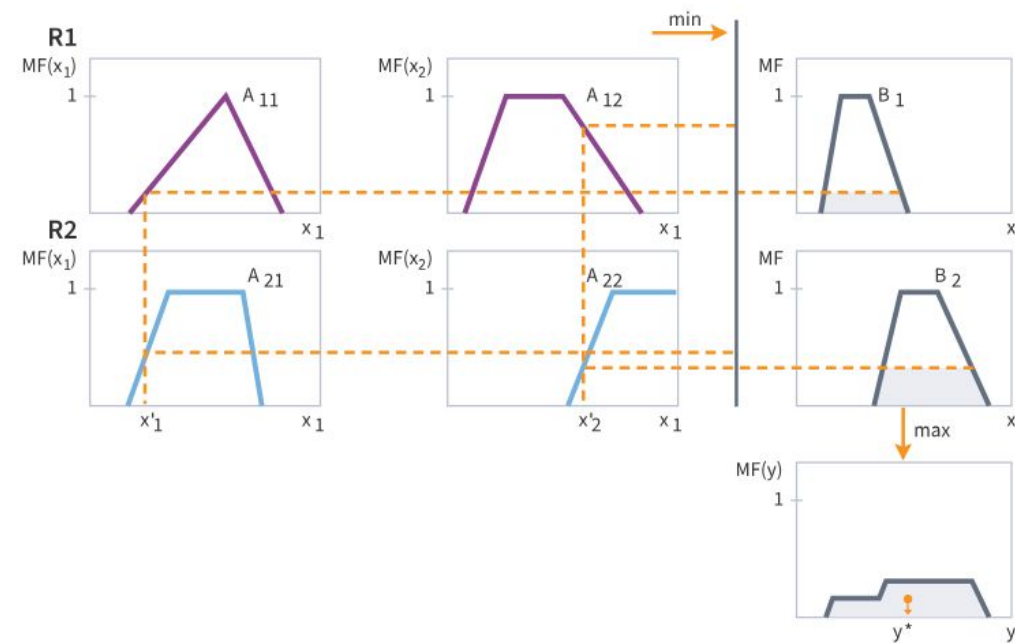
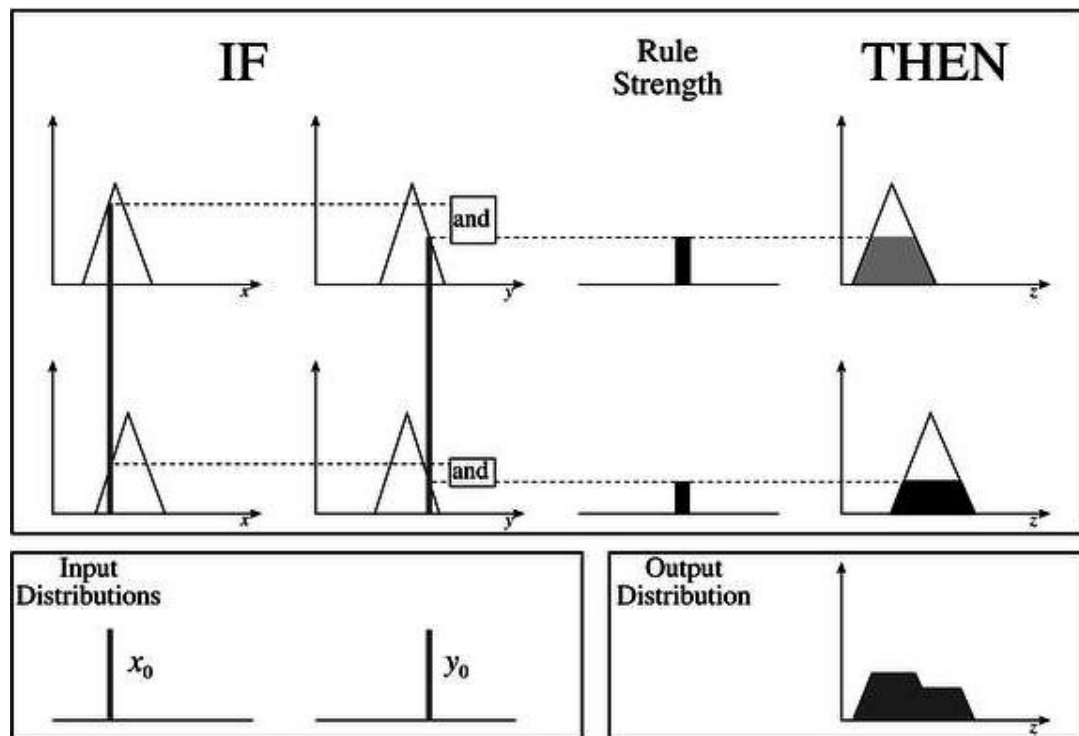
If x1 is A1 and x2 is A2 then y is B → Если x1 это A1 И x2 это A2 тогда y это B

Механизм работы Мамдани

- 1) Процедура фазификации: определяются степени истинности, т.е. значения функций принадлежности для левых частей каждого правила (предпосылок).
- 2) Нечеткий вывод. Сначала определяются уровни «отсечения» для левой части каждого из правил. Находятся «усеченные» функции принадлежности
- 3) Композиция, или объединение полученных усеченных функций, для чего используется максимальная композиция нечетких множеств.
- 4) Дефазификация, или приведение к четкости.

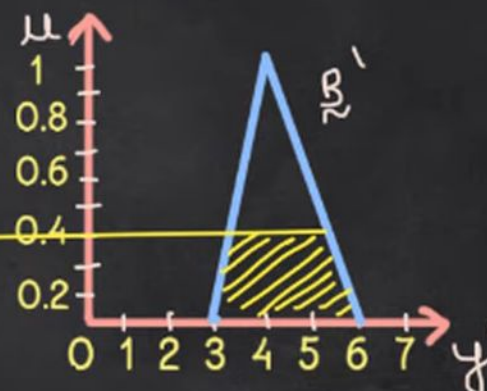
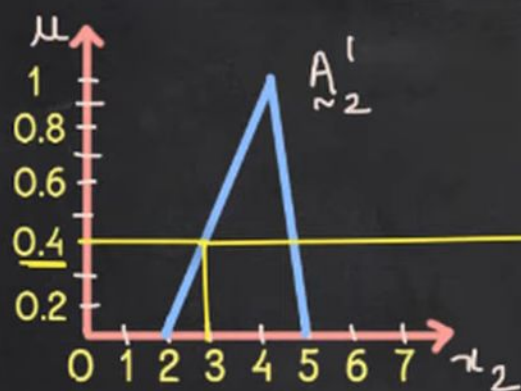
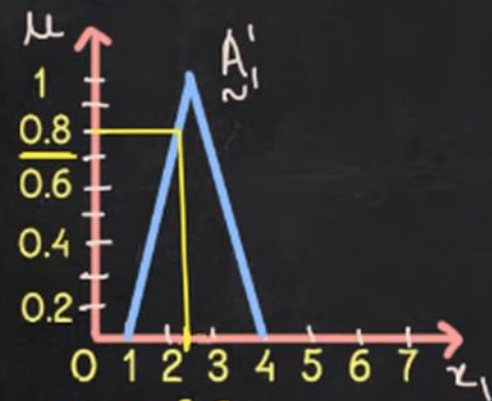
Если переменные в правиле объединены операцией `and`, то в качестве выхода принимается минимальное значение среди переменных, если операцией `or`, то максимальное значение.

Блок-схема нечеткой интерфейсной системы Мамдани

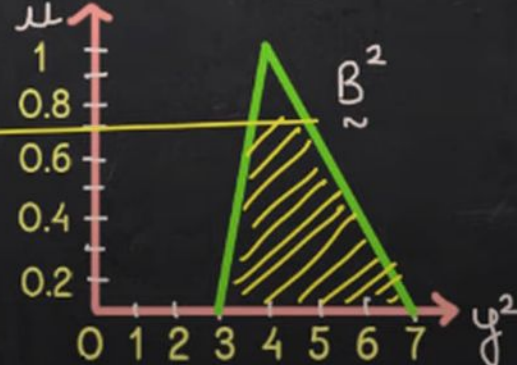
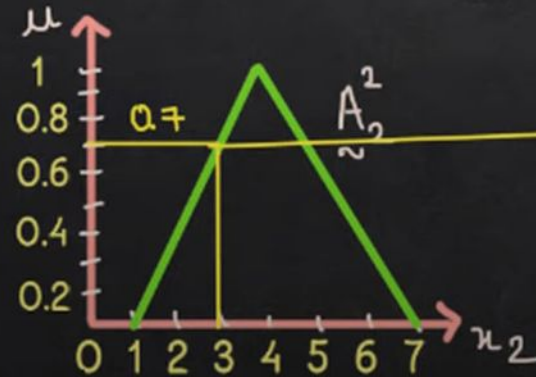
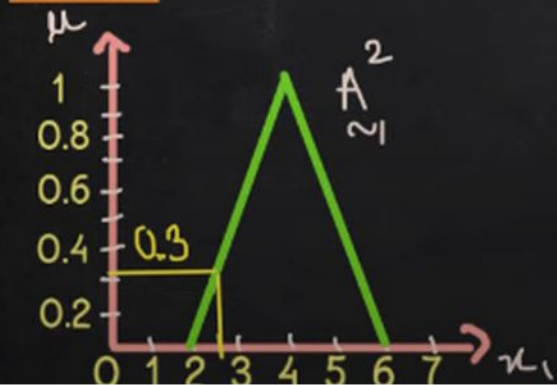


Пример

Rule 1:

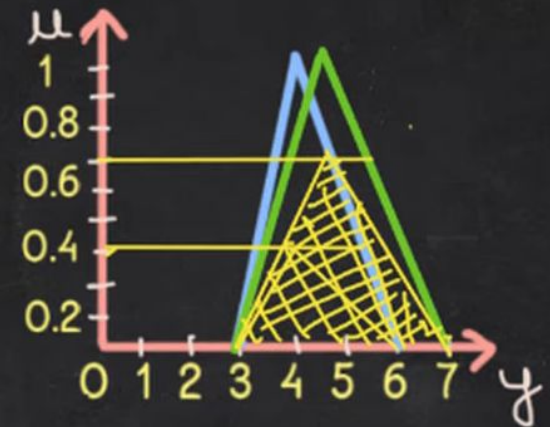
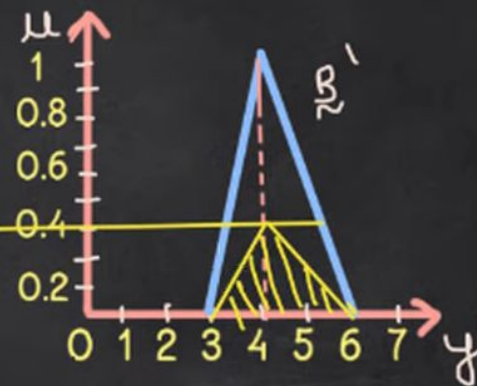
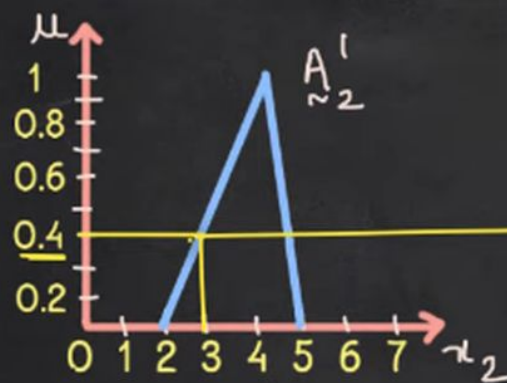


Rule 2:

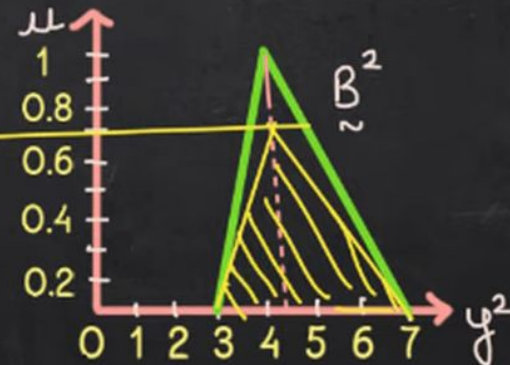
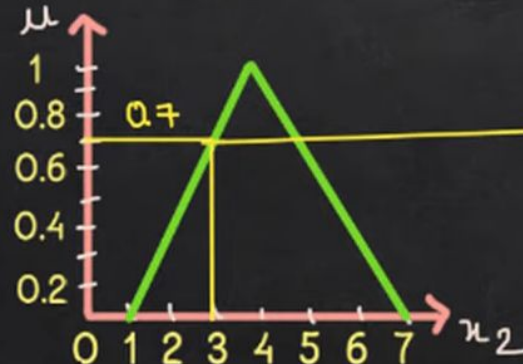
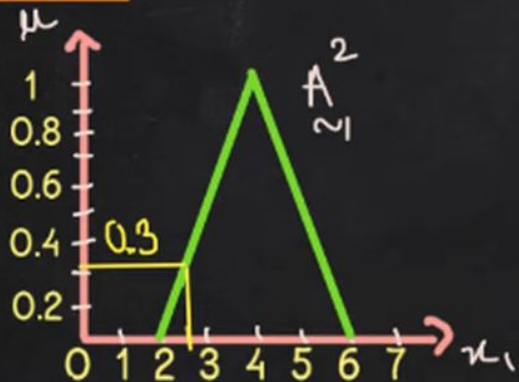


Пример с учетом функций принадлежности

Rule 1:



Rule 2:



Нечеткая модель Такаги-Сугено (метод TS)

Эта модель была предложена Такаги-Сугено и Кангом в 1985 году. Формат этого правила представлен как –

ЕСЛИ x – это A , а y – B , то $Z = f(x, y)$ (Сравните с Мамдани Если x_1 это A_1 И x_2 это A_2 тогда y это B)

Здесь AB – нечеткие множества в прошлых периодах, а $z = f(x, y)$ – четкая функция в последующем. Итоговой выход рассчитывается следующим образом:

$$y^* = \frac{\omega_{y_1} * y_1 + \omega_{y_2} * y_2 + \dots + \omega_{y_n} * y_n}{\omega_{y_1} + \omega_{y_2} + \dots + \omega_{y_n}}$$

y_1, y_2, y_n – результат применения правила (рассчитанное значение выхода по функции)

ω_{y_n} - минимум или максимум переменных из правила

Процесс нечеткого вывода Сугено

Процесс нечеткого вывода по нечеткой модели Такаги-Сугено (метод TS) работает следующим образом:

Шаг 1: Фаззификация входов – преобразование четких логических входов в нечеткие.

Шаг 2: Применение нечеткого оператора – на этом шаге должны быть применены нечеткие операторы для получения выходных данных.

Шаг 3: Дефаззификация выхода.

Графическая интерпретация метода Сугено

Пусть даны 4 правила:

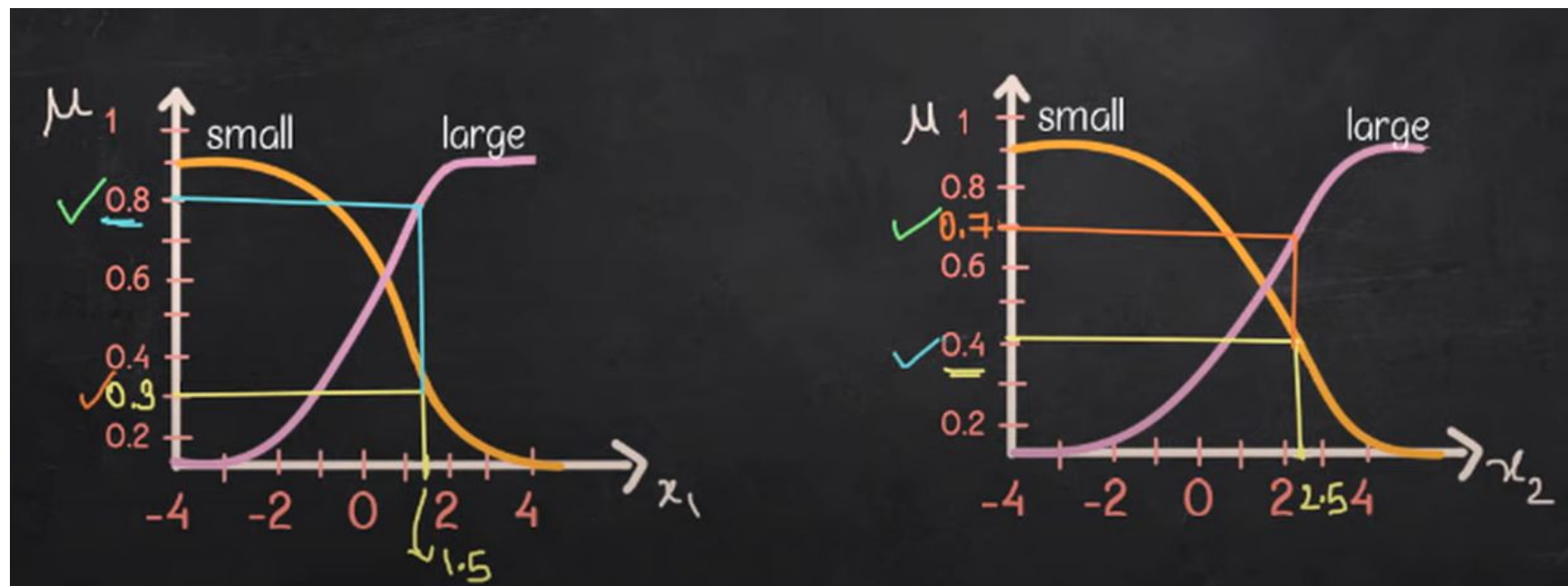
Rule 1: If x_1 is small and x_2 is small, They $y_1 = (-x_1) + x_2 + 1$

Rule 2: If x_1 is small and x_2 is large, They $y_2 = (-x_2) + 3$

Rule 3: If x_1 is large and x_2 is small, They $y_3 = (-x_1) + 3$

Rule 4: If x_1 is large and x_2 is large, They $y_4 = (-x_1) + x_2 + 2$

Входы $x_1 = 1.5$; $x_2 = 2.5$



Rule 1 : IF x_1 is small and x_2 is small, THEN $y_1 = (-x_1) + x_2 + 1$

Rule 2 : IF x_1 is small and x_2 is large, THEN $y_2 = (-x_2) + 3$

Rule 3 : IF x_1 is large and x_2 is small, THEN $y_3 = (-x_1) + 3$

Rule 4 : IF x_1 is large and x_2 is large, THEN $y_4 = (-x_1) + x_2 + 2$

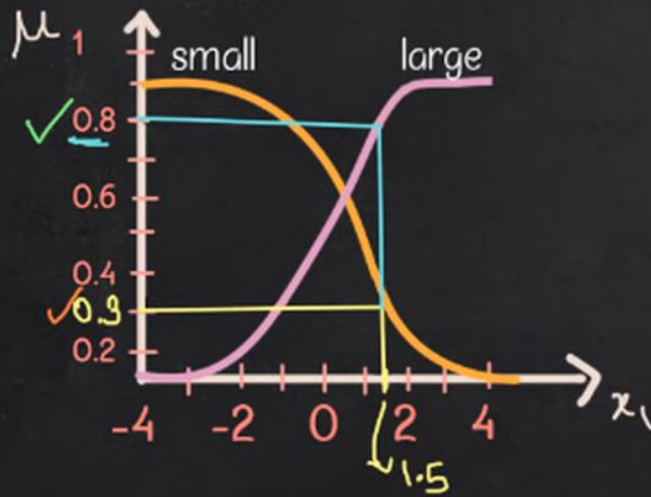
$$\omega_{y_1} = \min(\omega_{x_1}, \omega_{x_2}) = \min(0.3, 0.4)$$

$$\omega_{y_1} = \underline{\underline{0.3}}$$

$$\omega_{y_2} = \min(0.3, 0.7) = \underline{\underline{0.3}}$$

$$\omega_{y_3} = \min(0.8, 0.4) = \underline{\underline{0.4}}$$

$$\omega_{y_4} = \min(0.8, 0.7) = \underline{\underline{0.7}}$$



$$y_1 = (-x_1) + x_2 + 1 = (-1.5) + 2.5 + 1$$

$$y_1 = \underline{\underline{2}}$$

$$y_2 = \underline{\underline{0.5}}$$

$$y_3 = \underline{\underline{1.5}}$$

$$y_4 = \underline{\underline{6}}$$

$$y^* = \frac{\omega_{y_1} \cdot y_1 + \omega_{y_2} \cdot y_2 + \omega_{y_3} \cdot y_3 + \omega_{y_4} \cdot y_4}{\omega_{y_1} + \omega_{y_2} + \omega_{y_3} + \omega_{y_4}}$$

Сравнение методов Мамдани и Сугено

Метод Мамдани:

- Использование правил: В методе Мамдани используются нечеткие правила вида "Если А, то В," где А и В - нечеткие множества.
- Выходные значения: Выходное значение в методе Мамдани также является нечетким множеством, что позволяет учитывать неопределенность в выводе.
- Агрегация: Агрегация в методе Мамдани происходит с использованием операции взвешенной суммы правил с учетом функций принадлежности входных переменных.
- Дефаззификация: Дефаззификация преобразует нечеткий вывод в численное значение с использованием методов, таких как центр тяжести, максимум или среднего значения.

Метод Сугено:

- Использование правил: В методе Сугено используются нечеткие правила вида "Если A , то $Y = f(x)$," где A - нечеткое множество, а $Y = f(x)$ - функция.
- Выходные значения: В методе Сугено выходное значение является численным значением, а не нечетким множеством. Это делает результат более интерпретируемым.
- Агрегация: Вместо агрегации нечетких выводов метод Сугено использует функции применения (взвешивания) для каждого правила, чтобы получить численное значение выхода.
- Дефаззификация: Дефаззификация происходит после агрегации и представляет собой применение функции применения для определения численного выхода.

Сравнение:

- Метод Мамдани подходит для задач, где интерпретация результата в терминах нечетких множеств важна, а метод Сугено более подходит для задач, где результат должен быть численным и легко интерпретируемым.
- Метод Сугено менее чувствителен к нечетким правилам и входам, так как выходы являются численными, что может быть полезно в задачах с менее точными данными.
- Метод Мамдани может обрабатывать более сложные логические отношения между входами и выходами, в то время как метод Сугено ограничивается более простыми функциональными зависимостями.

Применение нечеткой логики в системах управления

- 1) Управление скоростью и движением автомобилей: Нечеткая логика используется для контроля скорости, торможения и управления движением автомобилей. Это позволяет автомобилям более безопасно и эффективно реагировать на различные дорожные условия.
- 2) Климат-контроль и отопление: Нечеткая логика используется для регулирования температуры в помещениях и оптимизации работы систем отопления и кондиционирования воздуха.
- 3) Контроль роботов: В робототехнике нечеткая логика используется для управления движением и действиями роботов в переменных окружающих условиях.
- 4) Системы безопасности: Нечеткая логика может применяться в системах безопасности для обнаружения и реагирования на различные ситуации, такие как вторжения, пожары и системы контроля доступа.
- 5) Управление транспортом: В общественном транспорте и системах управления трафиком нечеткая логика может помочь в оптимизации движения транспорта, управлении светофорами и контроле маршрутов.
- 6) Управление производственными процессами: В производственных системах нечеткая логика может быть использована для оптимизации производственных процессов, контроля качества и управления оборудованием.
- 7) Медицинская диагностика: В медицине нечеткая логика может помочь в диагностике и принятии решений на основе медицинских данных и симптомов.
- 8) Рекомендательные системы: Нечеткая логика может быть использована в рекомендательных системах для анализа пользовательских предпочтений и предоставления персонализированных рекомендаций.
- 9) Контроль процессов в зданиях: В умных зданиях нечеткая логика может контролировать освещение, вентиляцию и другие системы для повышения комфорта и эффективности.

Заключение

Фреймовые модели используются в искусственном интеллекте и информационных системах для структурирования и организации знаний.

Фреймы позволяют описывать не только взаимосвязи, но и их структуру.