Нечёткая логика

Малов Андрей Школьник Савелий

Магомедрасулов Курбан Богатиков Валерий

23 апреля 2020 г.

Оглавление

Глава 1

Введение

1.1 Определение нечёткой логики

Нечёткая логика — это форма многозначной логики, в которой истинностными значениями переменных могут быть любые вещественные числа от 0 до 1 включительно. Она используется для обработки концепции частичной истины, где значение истины может колебаться между полностью истинным и полностью ложным. В отличие от булевой логики, где значениями переменных могут быть только целочисленные значения 0 или 1.

Термин нечёткая логика был введён в 1965 году в теории нечётких множеств Лотфи Заде. Хотя нечёткая логика изучалась с 1920-х годов как бесконечнозначная логика, особенно Лукасевичем и Тарским.

Нечёткая логика основана на наблюдении, что люди принимают решения на основе неточной и нечисловой информации. Нечёткие модели или наборы являются математическим средством представления неопределённости и неточной информации (отсюда и термин нечёткий). Эти модели обладают способностью распознавать, представлять, манипулировать, интерпретировать и использовать данные и информацию, которые являются расплывчатыми и недостоверными.

Нечёткая логика применяется во многих областях, от теории управления до искусственного интеллекта.

1.2 Общее представление нечёткой логики

Классическая логика допускает только те выводы, которые либо истинны, либо ложны. Однако существуют также предложения с переменными ответами, например, если попросить группу людей определить цвет.

В таких случаях истина появляется в результате рассуждения на основе неточного или частичного знания, в котором отобранные ответы отображаются на спектре.

Обе степени истинности и вероятности находятся в диапазоне от 0 до 1 и поэтому могут показаться похожими на первый взгляд, но нечёткая логика использует степени истинности как математическую модель *неопределённости*, в то время как вероятность - это математическая модель *незнания*.

1.2.1 Применение истинных значений

Базовое приложение может характеризовать различные поддиапазоны непрерывной переменной. Например, измерение температуры для АБС тормозов может иметь несколько отдельных функций принадлежности, определяющих определённые температурные диапазоны, необходимые для правильного управления тормозами. Каждая функция отображает одно и то же значение температуры в диапазоне от 0 до 1. Эти значения истинности могут затем использоваться, чтобы определить, как следует управлять тормозами.

1.2.2 Лингвистические переменные

В то время как переменные в математике обычно принимают числовые значения, в нечёткой логике нечисловые значения часто используются для облегчения выражения правил и фактов.

Лингвистическая переменная, *возраст* может принимать такие значения, как *молодой*, так и его антоним *старый*. Поскольку естественные языки не всегда содержат достаточное количество ценностных терминов, чтобы выразить нечёткую шкалу значений, общепринятой практикой является изменение лингвистических значений с помощью прилагательных или наречий.

Операции фаззификации могут отображать математические входные значения в нечёткие функции принадлежности. А противоположные операции де-фаззификации могут быть использованы для отображения нечёткой выходной функции принадлежности в «чёткое» выходное значение, которое затем может быть использовано для целей принятия решений или управления.

1.3 Процесс

1. Привести все входные значения к нечётким функциям принадлежности.

2. Выполнить все применимые правила в базе правил для вычисления нечётких выходных функций.

3. Де-фаззифицировать нечёткие выходные функции, чтобы получить «чёткие» выходные значения.

1.3.1 Фаззификация

Фаззификация - это процесс присвоения числовых входных данных системы нечётким множествам с некоторой степенью принадлежности. Эта степень принадлежности может быть задана в пределах интервала [0,1]. Если она равна 0, то значение не принадлежит данному нечёткому множеству, а если равно 1, то значение полностью принадлежит нечёткому множеству. Любое значение в диапазоне от 0 до 1 представляет собой степень неопределённости того, что это значение принадлежит множеству. Эти нечёткие множества обычно описываются словами, и поэтому, назначая входные данные системы нечётким множествам, мы можем рассуждать о них лингвистически естественным образом.

Например, на рисунке ниже значения выражений *холодный*, *тёплый* и *горячий* представлены функциями, отображающими шкалу температуры. Точка на этой шкале имеет три «истинностных значения» — по одному для каждой из трёх функций. Вертикальная линия на изображении представляет собой определённую температуру, которую измеряют три стрелки (истинные значения). Поскольку красная стрелка указывает на ноль, эта температура может быть интерпретирована как «не горячая», т. е. эта температура имеет нулевое членство в нечётком множестве «горячая». Оранжевая стрелка (указывающая на 0.2) может описать его как «слегка тёплый», а синяя стрелка (указывающая на 0.8) — как «довольно холодный». Следовательно, эта температура имеет 0,2 членства в нечётком множестве «тёплый» и 0,8 членства в нечётком множестве «холодный». Степень принадлежности, присвоенная каждому нечёткому множеству, является результатом размытия.

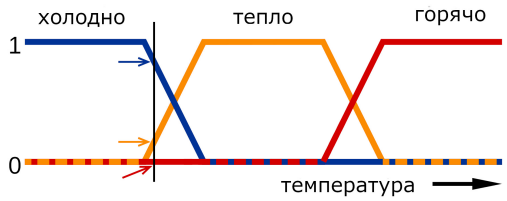

Рис. 1.1: Температура нечёткой логики

Нечёткие множества часто определяются как треугольные или трапециевидные кривые, наклон может быть, если значение увеличивается, пик, там где значение равно 1 (который может иметь длину 0 или больше) и наклон, где значение уменьшается. Они также могут быть определены с помощью сигмоиды. Одним из распространённых случаев является стандартная логистическая функция, определяемая как

$$S(x) = \frac{1}{1 + e^{-x}}$$

которая обладает следующим свойством симметрии

$$S(x) + S(-x) = 1$$

Из этого следует, что

$$(S(x) + S(-x)) * (S(y) + S(-y)) * (S(z) + S(-z)) = 1$$

1.3.2 Операторы нечёткой логики

Нечёткая логика работает со значениями принадлежности таким образом, что имитирует Булеву логику. С этой целью должны быть доступны замены для основных операторов И, ИЛИ, НЕТ. Для этого есть несколько способов. Распространённая замена называется *операторами Заде*:

Булева логика	**Нечёткая логика**
$AND(x, y)$	$MIN(x, y)$
$OR(x, y)$	$MAX(x, y)$
$NOT(x)$	$1 - x$

Для TRUE/1 и FALSE/0 нечёткие выражения дают тот же результат, что и логические выражения.

Однако таблица произвольного выбора не всегда определяет функцию нечёткой логики. Был сформулирован критерий для определения

того, определяет ли данная таблица выбора функцию нечёткой логики, и был предложен простой алгоритм синтеза нечёткой логической функции, основанный на введённых понятиях составляющих минимума и максимума. Функция нечёткой логики представляет собой дизъюнкцию составляющих минимума, где составляющая минимума представляет собой совокупность переменных текущей области, большей или равной значению функции в этой области (справа от значения функции в неравенстве, включая значение функции).

Другой набор операторов И/ИЛИ основан на умножении

x AND $y = x * y$
x OR $y = 1 - (1 - x) * (1 - y) = x + y - x * y$

$1 - (1 - x) * (1 - y)$ исходит из этого:

x OR $y = \text{NOT}(\text{AND}(\text{NOT}(x), \text{NOT}(y)))$
x OR $y = \text{NOT}(\text{AND}(1 - x, 1 - y))$
x OR $y = \text{NOT}((1 - x) * (1 - y))$
x OR $y = 1 - (1 - x) * (1 - y)$

1.3.3 Правило ЕСЛИ-ТО

Правила ЕСЛИ-ТО сопоставляют входные или вычисленные значения истинности с требуемыми выходными значениями истинности. Пример:

ЕСЛИ температура *очень холодная*, ТО скорость вращения вентилятора *останавливается*
ЕСЛИ температура *холодная*, ТО скорость вращения вентилятора *медленная*
ЕСЛИ температура *тёплая*, ТО скорость вращения вентилятора *умеренная*
ЕСЛИ температура *горячая*, ТО скорость вращения вентилятора *высокая*

При определённой температуре нечёткая переменная *горячий* имеет определённое значение истинности, которое копируется в переменную *высокая*.

1.3.4 Дефаззификация

Цель состоит в том, чтобы получить непрерывную переменную из нечётких истинностных значений.

Это было бы легко сделать, если бы выходные истинностные значения были точно такими же, как и те, которые получены в результате фаззификации данного числа. Однако, поскольку все выходные значения истинности вычисляются независимо, в большинстве случаев они не представляют такой набор чисел. Затем нужно выбрать число, которое лучше всего соответствует «намерению», закодированному в значении истинности. Например, для нескольких истинностных значений *скорости вращения вентилятора* необходимо найти фактическую скорость, которая наилучшим образом соответствует вычисленным истинностным значениям переменных «медленный», «умеренный» и т. д.

Для этой цели не существует единого алгоритма.

Общий алгоритм таков:

1. Для каждого значения истинности сократите функцию принадлежности на это значение
2. Объедините полученные кривые с помощью оператора OR
3. Найдите центр тяжести области под кривой
4. Позиция x этого центра является окончательным выводом.

1.4 Формирование консенсуса входных данных и нечётких правил

Поскольку вывод нечёткой системы является консенсусом всех входов и всех правил, системы нечёткой логики могут хорошо себя вести, когда входные значения недоступны или не заслуживают доверия. Весовые коэффициенты могут быть дополнительно добавлены к каждому правилу в базе правил, также весовые коэффициенты могут использоваться для регулирования степени влияния правила на выходные значения. Эти весовые коэффициенты могут основываться на приоритете, надёжности или согласованности каждого правила. Эти веса правил могут быть статическими или могут изменяться динамически, даже на основе выходных данных других правил.

1.5 Первое применение

Многие из ранних успешных применений нечёткой логики были реализованы в Японии. Первое известное применение это — метро в Сендай, в котором нечёткая логика была в состоянии улучшить экономичность, комфорт и точность поездки. Нечёткая логика также используется для распознавания рукописных символов в карманных компьютерах Sony, для помощи в полете для вертолётов, для управления системами метрополитена с целью повышения комфорта при вождении, точности остановки и экономии энергии, повышения расхода топлива для автомобилей, управление одной кнопкой для стиральных машин, автоматическое управление двигателем для пылесосов с распознаванием состояния поверхности и степени загрязнения, а также системы прогнозирования для раннего распознавания землетрясений через Институт сейсмологии Бюро метеорологии, в Японии.

Глава 2

Программная часть

2.1 Выбор инструмента

Задача состоит в том, чтобы составить программу для...

2.2 Программа на Python

```
1
2 import matplotlib.pyplot as plt
3
4 def triangle(upper, lower, peak, x):
5   u = 0
6   if ((x >= lower) and (x <= peak)):
7     u = 1 - ((peak - x) / (peak - lower))
8     return u
9   if ((x >= peak) and (x <= upper)):
10    u = 1 - ((x - peak) / (upper - peak))
11    return u
12  return u
13
14 def trapezoid(upper, lower, peak_lower, peak_upper, x):
15  u = 0
16  if ((x >= lower) and (x <= peak_lower)):
17    u = 1 - ((peak_lower - x) / (peak_lower - lower))
18    return u
19  if (x >= peak_lower) and (x <= peak_upper):
20    u = 1
21    return u
```

```
22    if ((x >= peak_upper) and (x <= upper)):
23      u = 1 - ((x - peak_upper) / (upper - peak_upper))
24      return u
25    return u
26
27
28  def main():
29
30    flag = int(input("What method do you want to solve? (
       triangular or trapezoidal) (1 or 2) --> "))
31
32
33    upper = float(input("Enter upper value  --> "))
34    lower = float(input("Enter lower value --> "))
35    if flag == 1:
36      peak = float(input("Enter yours peak value --> "))
37    else:
38      peak1 = float(input("Enter yours first peak value
       --> "))
39      peak2 = float(input("Enter yours second peak value
       --> "))
40    x = float(input("Enter crisp value --> "))
41
42    if flag == 1:
43      u = triangle(upper, lower, peak, x)
44
45      x_list = [lower, upper-x, x, upper]
46      y_list = [0, u, u, 0]
47
48
49      if (u == 1):
50        plt.plot([lower, peak, upper], [0, 1, 0])
51      else:
52        plt.plot(sorted(x_list), y_list)
53
54    else:
55      u = trapezoid(upper, lower, peak1, peak2, x)
56
57      x_list = [lower, upper-x, x, upper]
58      y_list = [0, u, u, 0]
59
```

```
60      if (u == 1):
61          plt.plot([lower, peak1, peak2, upper], [0, 1, 1,
        0])
62      else:
63          plt.plot(sorted(x_list), y_list)
64
65    print(u)
66    plt.show()
67
68 if __name__ == "__main__":
69    main()
```

Листинг 2.1: Полная программа на Python

Глава 3

Заключение

В заключение хочется отметить, что нечёткая логика всё активнее используется в разных областях нашей жизни. В середине 1970-х гг. были предложены первые реализации нечётких моделей в промышленности, а в начале 1980-х гг. нечеткая математика получила свое дальнейшее развитие в целом ряде программных средств поддержки принятия решений и в экспертных системах анализа данных.

В настоящее время приложения нечеткой математики можно найти в сотнях промышленных изделий — от систем управления электропоездами и боевыми вертолетами до бытовой техники. Нечеткая логика используется для принятия политических решений и моделирования возможных кризисных ситуаций в современных ситуационных центрах руководителей западных стран, в программных системах, обслуживающих большой бизнес. Например, компания Yamaichi Securities на основе нечёткой логики разработала интеллектуальную банковскую систему для средне и долгосрочных операций с корпоративными бумагами системы.

Нечеткой логике обязано своим рождением и новое поколение систем имитационного моделирования. Большинство программных комплексов, используемых в мире для экономического, политического и финансового моделирования, базируется на методах динамики систем (system dynamics). Системная динамика — это метод имитационного моделирования, основанный на представлении системы на высоком уровне абстракции как совокупности потоков, накопителей, вспомогательных переменных и субмоделей со своими элементами.

В последнее время нечеткое управление является одной из самых активных и результативных областей исследований применения теории нечетких множеств. Нечеткое управление оказывается особенно полезным, когда технологические процессы являются слишком сложными для

анализа с помощью общепринятых количественных методов, или когда доступные источники информации интерпретируются качественно, неточно или неопределенно. Экспериментально показано, что нечеткое управление дает лучшие результаты, по сравнению с получаемыми при общепринятых алгоритмах управления.

Нечеткие методы помогают управлять домной и прокатным станом, автомобилем и поездом, распознавать речь и изображения, проектировать роботов, обладающих осязанием и зрением. Нечеткая логика, на которой основано нечеткое управление, ближе по духу к человеческому мышлению и естественным языкам, чем традиционные логические системы. Нечеткая логика, в основном, обеспечивает эффективные средства отображения неопределенностей и неточностей реального мира. Наличие математических средств отражения нечеткости исходной информации позволяет построить модель, адекватную реальности.

Литература