**БИБЛИОТЕКИ ДЛЯ РЕАЛИЗАЦИИ МАТРИЧНОГО НЕЧЁТКОГО ЛОГИЧЕСКОГО ВЫВОДА НА PYTHON**

**Описание функциональных возможностей**

На 10 листах

Епишина Наталья

epishina.nata@bk.ru

Пермь, 2023

Оглавление

[1 Общие сведения 3](#_Toc133363912)

[2 Функциональное назначение 4](#_Toc133363913)

[3 Описание логической структуры 5](#_Toc133363914)

[4 Используемые технические средства 10](#_Toc133363915)

# Общие сведения

Библиотека на python для работы с нечёткой логикой. Нечёткий вывод используется для решения задач, где необходимо работать с нечеткими, невыразительными или неопределенными данными. Например, в задачах управления процессами, где точные математические модели могут быть слишком сложными. Нечеткий вывод позволяет работать с нечеткими данными и принимать решения на основе лингвистических переменных, которые не имеют четкого числового значения.

# Функциональное назначение

Данная библиотека является инструментом для построения нечетких систем.

Библиотека позволяет:

* использовать матричные нечёткие векторы, предикаты;
* задавать входные и выходные параметры алгоритма;
* задавать графики функций истинности предикатов выбором из заранее настроенного набора;
* создавать правила базы знаний;
* производить нечёткий логический вывод с использованием созданной базы знаний в соответствии с матричным алгоритмом или алгоритмом Мамдани.

# Описание логической структуры

Реализация нечеткого логического вывода состоит из 7 классов (рис. 1).

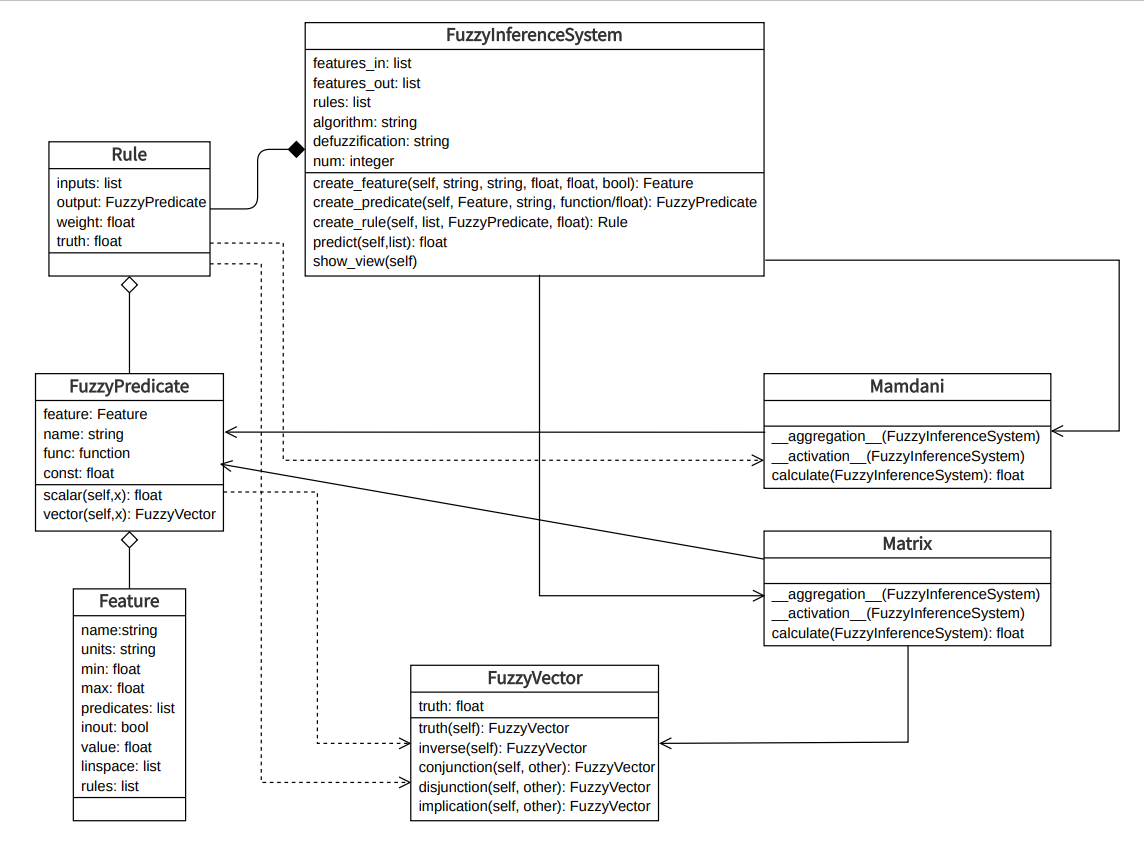


Рис. 1. Диаграмма классов системы нечеткого логического вывода

Так же в программе реализованы три функции принадлежности: треугольная, трапециевидная и гауссовская.

* Класс FuzzyVector

Класс описывает нечёткую векторную переменную, содержит поле truth – степень истинности.

Методы класса:

* truth(self) – значение поля truth, возвращает объект класса FuzzyVector;
* inverse(self) – операция инверсии нечёткого вектора, возвращает FuzzyVector;
* conjunction(self, other) – операция конъюнкции для двух нечётких векторов, возвращает объект класса FuzzyVector;
* disjunction(self, other) – операция дизъюнкции для двух нечётких векторов, возвращает объект класса FuzzyVector;
* implication(self, other) – операция обратной импликации для двух нечётких векторов, данная операция нужна для вычисления степени истинности заключения по известным значениям степени истинности условия и весе правила, возвращает объект класса FuzzyVector.
* Класс Feature

Класс описывает лингвистическую переменную, содержит поля name – название ЛП, units – система измерения, min – минимальное значение, max – максимальное значение, predicates – список термов, inout – значение типа bool, которое показывает входная это переменная (true) или выходная (false), value – значение переменной, если она является входной, linspace – список возможных значений от min до max, rules – список правил, которые говорят о данном рассчётном признаке.

* Класс FuzzyPredicate

Класс описывает терм, содержит поля feature – лингвистическая переменная, которой принадлежит данный терм, name – название терма, func – форма для описания нечёткого множества (треугольная, трапециевидная, гауссова), const – значение центра тяжести фигуры, которой принадлежит терм.

Методы класса:

* scalar(self, x) – переход от заданных чётких значений к степеням уверенности, возвращает float;
* vector(self, x) – представление степени уверенности в виде нечёткого вектора, возвращает объект класса FuzzyVector.
* Класс Rule

Класс описывает правила, содержит поля inputs – список из условий, output – заключение, weight – вес правила, степень уверенности в нём, thruth –истинность заключения.

* Класс Mamdani

Класс описывает алгоритм Мамдани.

Методы класса:

* \_\_aggregation\_\_ (FuzzyInferenceSystem) – агрегирование подусловий, вычисление истинности условий, в данном методе происходит запись в истинность правила значения минимума среди степеней уверенности входных параметров правила;
* \_\_activation\_\_(FuzzyInferenceSystem) – активизация подзаключений, вычисление степени истинности заключения путем умножения истинности условий на вес правила;
* calculate(FuzzyInferenceSystem) – вычисление значения выходного параметра, возвращает float. Реализовано два метода дефиззификации, с помощью расчета центра тяжести масс и упрощенный вариант.
* Класс Matrix

Класс описывает матричный нечёткий логический вывод.

Методы класса:

* \_\_aggregation\_\_ (FuzzyInferenceSystem) – агрегирование подусловий, вычисление истинности условий, в данном методе происходит запись в истинность правила результат операции конъюнкции степеней уверенности входных параметров правила;
* \_\_activation\_\_ (FuzzyInferenceSystem) – активизация подзаключений, вычисление степени истинности заключения с помощью обратной импликации;
* calculate (FuzzyInferenceSystem) – вычисление значения выходного параметра, возвращает float. Реализовано два метода дефиззификации, с помощью расчета центра тяжести масс и упрощенный вариант. При выборе рассчёта с помощью центра тяжести масс сначала обрезаем функцию принадлежности выходного параметра используя операцию конъюнкции, а затем объединяем все правила с помощью операции дизъюнкции, далее используем формулу, описанную выше.
* Класс FuzzyInferenxeSystem

Класс описывает нечёткую систему, с которой мы будем работать, содержит поля features\_in – список лингвистической переменной, которые являются входными параметрами, features\_out – список лингвистической переменной, которые являются выходными параметрами, rules – список правил, algorithm – выбор алгоритма (Mamdani или Matrix), defuzzification – выбор метода получения чёткого результата (Centroid или Simple), num – количество значений для интервала допустимых значений лингвистической переменной.

Методы класса:

* create\_feature (self, name, units, min, max, inout) – запись лингвистической переменной, возвращает объект класса Feature;
* create\_predicate (self, feature: Feature, name, func=None, const=None) – запись терма, возвращает объект класса FuzzyPredicate;
* create\_rule (self, input\_predicates, output\_predicate, weight) – запись правила, возвращает объект класса Rule;
* predict (self,\*x) – вычисление значений выходных параметров по входным данным, возвращает float;
* show\_view (self) – вывод всех графиков принадлежности лингвистических переменных и соответствующих им термов.

Пользователю предлагается использовать следующие элементы библиотеки:

* **FuzzyInferenceSystem** – основной класс, отвечающий за взаимодействие внешней программы с библиотекой.
* **algorithm** – выбор алгоритма (Mamdani или Matrix);
* **defuzzification** - выбор метода получения чёткого результата (Centroid или Simple);
* **create\_feature** - метод для создания признака или лингвистической переменной (ЛП);
* **create\_predicate** – метод для создания предиката или терма ЛП;
* **create\_rule** – метод для создания правила;
* **predict** – метод для вычисления значений выходных параметров по заданным значениям входных.

# Используемые технические средства

Для работы с библиотекой достаточно иметь среду разработки, которая поддерживает язык программирования python, загрузить библиотеку можно через терминал, с помощью команды pip install fuzzy\_matrix\_mamdani.