БИБЛИОТЕКА ДЛЯ СОЗДАНИЯ ЭКСПЕРТНЫХ СИСТЕМ НА БАЗЕ АЛГОРИТМОВ НЕЧЕТКОГО ВЫВОДА НА ЯЗЫКЕ TYPESCRIPT

Ляпко А. М., Курочкин А. В.
Кафедра интеллектуальных систем, Факультет радиофизики и компьютерных технологий,
Белорусский государственный университет
Минск, Республика Беларусь
E-mail: alex.v.kurochkin@gmail.com, andrey.lyapko@gmail.com

Данная работа рассматривает разработку библиотеки для создания экспертных систем на базе алгоритмов нечеткого вывода на языке TypeScript. Исследуются основные особенности работы с нечеткой логикой, рассматривается статическая типизация лингвистических переменных и термов в TypeScript, а также приводятся особенности программного интерфейса разработанной библиотеки.

Введение

Построение экспертных систем на базе нечеткой логики позволяет описать процесс принятия решения формально, в виде правил, на понятном человеку или эксперту языке. В отличие от популярного на сегодняшний день подхода с использованием машинного обучения с учителем, для построения таких экспертных систем не требуется обширная обучающая выборка, а процесс принятия решений является прозрачным [1, 2].

Для поддержки построения экспертных систем на базе нечеткого вывода на различных языках и платформах существуют встроенные или сторонние библиотеки, предоставляющие соответствующую функциональность. Тем не менее, существующие реализации для языка JavaScript являются крайне ограниченными по функциональности, несмотря на очень высокую популярность языка на сегодняшний день.

Целью этой работы является рассмотрение основных особенностей разработки такой библиотеки, а также рассмотрение основных особенностей разработанного программного интерфейса для задания лингвистических переменных, термов и базы правил системы нечеткого вывода, и реализации непосредственного нечеткого вывода по алгоритмам Мамдани.

I. Системы нечеткого вывода

Экспертные системы на базе нечеткого вывода – математические модели, которые для моделирования процесса принятия решений используют базу правил, описанную в терминах классической логики высказываний, а также информацию об элементарных переменных и их возможных состояниях в терминах нечетких множеств. На основании базы правил отдельные элементарные высказывания могут быть определены на основании входных данных нечеткой системы и нечетких множеств, соответствующих термам этих высказываний. После этого на основании классического логического вывода строится математическая модель, которая связывает полученные нечеткие множества с элементарными высказываниями о неизвестных переменных.

Лингвистическая переменная описывается названием, универсальным множеством его возможных значений, а также набором нечетких множеств над универсальным множеством, называемых термами. Поскольку каждый терм лингвистической переменной является нечетким множеством, для элементарного высказывания «переменная = терм» по конкретному значению из универсального множества может быть определена принадлежность (истинность) как значение от 0 до 1 [1].

Разработка и применение систем нечеткого вывода включает в себя ряд этапов, реализация которых выполняется на основе рассмотренных ранее положений нечеткой логики. К таким этапам относятся:

1. Формирование базы правил
2. Фаззификация входных данных
3. Агрегирование подусловий
4. Активизация подзаключений
5. Аккумулирование заключений
6. Деффазификация выходных переменных

Рассмотренные этапы нечеткого вывода могут быть реализованы неоднозначным образом: агрегирование может проводиться не только в базисе нечеткой логики, активизация может проводиться различными методами нечеткой композиции, на этапе аккумуляции объединение можно провести отличным от max-объединения способом, дефаззификация также может проводиться различными методами. Таким образом, выбор конкретных способов реализации отдельных этапов нечеткого вывода определяет тот или иной алгоритм нечеткого вывода. В настоящее время остается открытым вопрос критериев и методов выбора алгоритма нечеткого вывода в зависимости от конкретной технической задачи [2].

II. Реализация библиотеки для поддержки нечеткого вывода

Разработанная библиотека предоставляет программный интерфейс для задания входных и выходных лингвистических переменных и их

термов, формирования базы правил и нечеткого вывода.

Лингвистические переменные задаются в виде объекта с двумя полями для описания входных переменных (`input`) и выходных переменных (`outputs`). Сами же лингвистические переменные представлены в виде объекта, у которого в качестве ключа используется название переменной, в качестве значения – объект с его термами. Каждый объект с термами в качестве ключа использует название терма, а в качестве значения – его функцию принадлежности. Для представления объекта с лингвистическими переменными был использован универсальный объект (`ObjectMap<T>`). Таким образом, поля `input` и `output` имеют один и тот же тип `ObjectMap<ObjectMap<Function>>`. Для перечисления переменных и термов в соответствующем объекте используется функция `Object.keys` [3].

Функция активации отдельного терма задаётся не аналитически, а в виде полноценной функции одной переменной. Для удобства использования в рамках библиотеки предоставляются реализации типичных функций активации – треугольной, трапецевидной, ступенчатой, сигмоидальной и т.п. с различными параметрами. Кроме того, библиотека предоставляет набор функций высшего порядка для комбинирования базовых функций принадлежности с операциями инверсии, объединения и пересечения на основе max-нормы. Кроме того, внешний интерфейс описания терма позволяет использовать в качестве функции активации произвольную реализацию, которая должна удовлетворять всем свойствам функции принадлежности.

База правил задаётся в виде массива элементов `Rule`, которые являются кортежами из условий `Condition` и заключений `Conclusion`. Условия и заключения, в свою очередь, являются нечеткими высказываниями `Statement`. Нечеткое высказывание содержит название нечеткой переменной и значение одного из её термов. С помощью строгой типизации на этапе компиляции подтверждается, что терм для указанной нечеткой переменной является одним из ключей в объявлении переменной. Также в заключении используется дополнительное поле с весовым коэфицентом правила. Данная структура позволяет удобно работать с данными на разных стадиях алгоритма; например, на стадии фаззификации и аггрегации удобно пользоваться массивом условий из базы правил, которая использует модель `Rule`.

В рамках разрабатываемой библиотеки реализован алгоритм нечеткого вывода Мамдани. К основным особенностям можно отнести следующие:

- Фаззификация входных переменных осуществляется на основании термов входных переменных и соответствующих им нечетких множеств;
- Агрегирование подусловий правил нечеткой продукции осуществляется при помощи нечеткой логической операции «И» на основании min-нормы $A, B : T(A \cap B) = min(T(A); T(B))$;
- Активизация подзаключений правил нечеткой продукции осуществляется методом min-активизации $\mu(y) = min(c; \mu(x))$, где $\mu(x)$ и c – соответственно функции принадлежности термов лингвистических переменных и степени истинности нечетких высказываний;
- Аккумуляция подзаключений правил нечеткой продукции проводится при помощи max-объединения функций принадлежности $\forall x \in X : \mu_{AB}(x) = max(\mu_A(x); \mu_B(x))$;
- Дефаззификация проводится методом центра тяжести или центра площади.

Для реализации нечеткого вывода по алгоритму Мамдани полученная нечеткая модель реализует метод `predict`, которому на вход передаются строгие значения входных переменных в виде объекта. В качестве возвращаемого значения выступает объект, полями которого являются названия выходных переменных, а значения – полученными дефаззифицированными выходными переменными.

Заключение

Разработка экспертных систем на базе нечеткого вывода позволяет формализовать экспертный процесс принятия решений и использовать полученную систему для решения различных задач. В работе рассмотрена реализация библиотеки для поддержки разработки экспертной системы для языка TypeScript и описаны основные особенности полученного программного интерфейса пользователя. Дальнейшие направления исследований включают возможность более тонкой настройки используемого метода нечеткого вывода, а также поддержка других алгоритмов нечеткого вывода и нейронечетких систем.

Список литературы

1. Куценко, Д. А. Представление и использование знаний в интеллектуальных САУ при помощи методов теории нечетких множеств / Д. А. Куценко // Вестник БГТУ им. В. Г. Шухова [Электронный ресурс]. – 2015. – № 4.– Режим доступа: http://nrsu.bstu.ru/chap27.html – Дата доступа: 17.05.2019.
2. Хаптахаева, Н. Б. Введение в теорию нечетких множеств: учебное пособие / Н. Б. Хаптахаева, С. В. Дамбаева, Н. Н. Аюшеева. – Улан-Удэ: Изд-во ВСГТУ, 2004. – Ч.1. – 68 С.
3. Fenton, S. Pro TypeScript: Application-Scale JavaScript Development / S. Fenton, –Apress, 2014. –1st ed. –248 P.