Fuzzy Weather Implementation Project

Presented to Dr. Robin Dawes

CISC 467 - Fuzzy Logic

School of Computing, Faculty of Arts and Science

Queen’s University

Prepared By:

Cesur Kavaslar (10176178)

Matthew Dixon (10185771)

December 16, 2019

Introduction

Inspired by the daily struggle of deciding what to wear in the morning, FuzzyWeather is a proof of concept fuzzy logic control system that uses OpenWeatherAPI and the user's inputted location to help choose what type of clothing they should wear based on that day's current weather and forecast.

Implementation

FuzzyWeather was written in Python 3.7 and used the requests package to make API requests, the numpy package to simplify using arrays, and scikit's skfuzzy package to build and compute the fuzzy logic rules. Four python files were used in the implementation to segment code and ensure the project's code repository was clean, concise, and robust.

FuzzyControlSystem.py

Interacts with the user to get their location, makes the API call to receive the weather details, and applies the fuzzy logic control system to complete the implementation.

FuzzyRules.py

Contains the definitions for all of the fuzzy logic rules used.

FuzzyMembership.py

Contains the definitions for all of the fuzzy logic membership functions used.

API_Keys.txt

Contains the credentials for the OpenWeatherAPI forecast API.

requirements.txt

Contains the dependent packages and modules for the program.

Use Instructions

As described in the implementation, the program depends on several packages. To install the required packages run the following command.

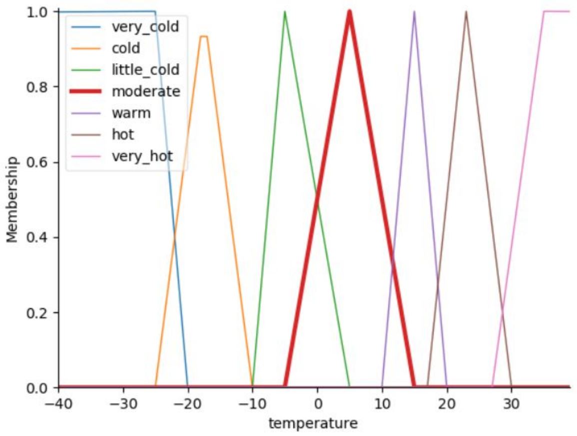
```
$ pip3 install requirements.txt
```

To run the program simply execute the FuzzyControlSystems.py program.

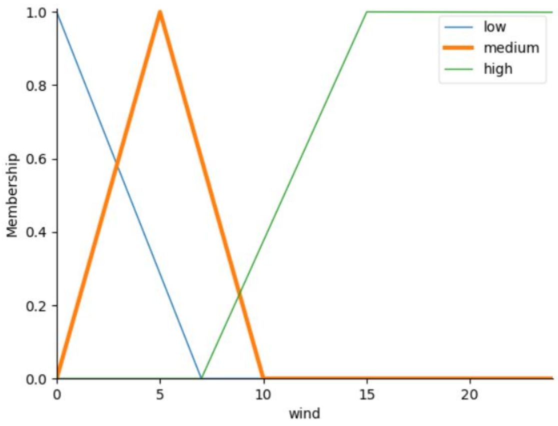
```
$ python3 FuzzyControlSystems.py
```

Membership Functions

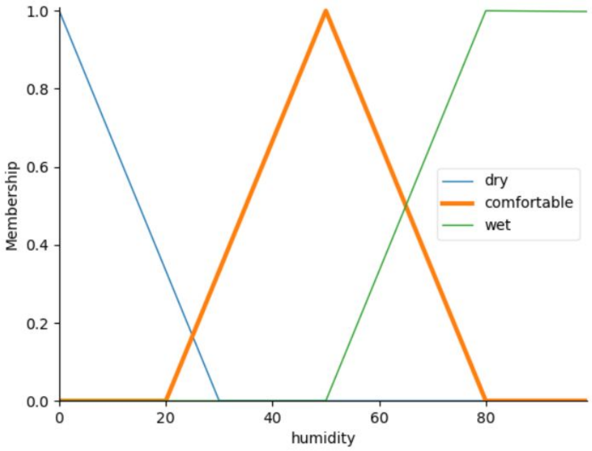
Temperature Membership Function (Celcius)



Wind Membership Function (km/hr)



Humidity Membership Function (%)



Fuzzy Rules

The control system implements two collections of rules. The first is focused on deciding to wear shorts or pants and includes the following rules:

1. If the temperature is warm, hot, or very hot, then wear shorts.
2. If the temperature is moderate, a little cold, cold, or very cold, then wear pants.
3. If the temperature is moderate and humidity is wet, then wear shorts.
4. If the temperature is warm and the wind speeds are high, then wear pants.

The second is aimed at deciding what type of top should be worn and includes the following rules:

1. If the temperature is hot or very hot, wear a shirt.
2. If the temperature is cold or very cold, wear a parka.

3. If the temperature is moderate, or the temperature is warm with high winds, wear a sweater.
4. If the temperature is a little cold, or the temperature is moderate and the humidity is wet, wear a light jacket.
5. If the winds are low and the temperature is warm, wear a shirt.

Future Work

There are several features we would like to implement in the future of the project. The first improvement we would like to make is to improve the location input handling. For example, during our testing we found that simply inputting “Kingston” defaulted to Kingston, Jamaica in the API. Similarly, we would like to eventually add location services so a user's location is automatically found without the need for any input.

We would also like to expand our rule set to include more weather conditions and provide more clothing suggestions beyond just tops and bottoms. As well, having more time to collect data on our existing rule set and membership functions would allow us to tweak our system to improve the accuracy of the results.

Finally, an idea we had was to integrate our program into an Alexa, Google Home, etc. routine. This would allow the user to be informed of what clothing choices to make through voice commands only, improving the ease of use of the system with the hopes that it would become part of their daily morning routine.