Carleton UNIVERSITY
Canada's Capital University

Nu-Sense

SYSC 5709 Course Project
User Manual

Course Supervisor:
Dr. Christina Ruiz Martin

Department of Systems and Computer Engineering,
Carleton University, Ottawa, ON, Canada

Revision 1.0

Student Name: Akaljot Singh
Student Name: Karanbir Singh Sandhu
Student Name: Ripudaman Singh

Submitted on Dec 17 2019

Table of Contents

Contents

1. Introduction

This manual incorporates manual to use Nu-Sense, that uses Sensor Fusion Algorithm to compute results. This sensor fusion algorithm emphasizes on generating single output from multiple outputs. This algorithm is based on "A Simple Multi-Sensor Data Fusion Algorithm Based on Principal Component Analysis". It takes input from multiple sensors. Initially, it computes support degree matrix which is further used to calculate eigen values and eigen vectors. Eigen values are sorted in descending order. Further, this helps in calculating principal component. At fourth step, contribution rate of kth component is calculated leading to accumulated contribution rate. At step 6, values from step 3 are used to calculate integrated support degree score. Further, equation is used to check if specific sensor is supported by specified percentage of sensors. Sensors which do not satisfy are removed from further calculations. These values are used to calculate omega for every sensor. This leads to final fusion for every sensor. At last, these fused values are combined to generate final output. Nu-sense takes input.csv file as input. This contains data for different time frames. It asks user to enter sensors and matches it to sensors available in csv file. If it matches programs proceeds, otherwise finishes by displaying error. After entering number of time instances, sensor fusion algorithm works on given inputs and write output to output.csv. It also checks for stuck sensors whose values do not change after specific time along with list of sensors whose values are out of range.

Software Development Environment:

1. Programming Language: C
2. IDE Used: XCode
3. Version Control System: GitHub
4. External Library used: GSL, GNU Scientific Library
5. Testing:
   - MiUnit Testing
6. Manual Documentation
7. External Reviewers

2. Compatible Platforms

This software is tested and compatible on below platforms

1. Ubuntu 18.04
2. macOS Catalina 10.15
3. Windows 10

3. Installation Steps:

- Install MinGW and extract msys.
- Add path in Environment variables.
- Copy msys to MinGW.
- Rename mingw-makefile.exe to make (File present in C->MinGW->bin)
- Compile GSL library using ./configure command. (Navigate to lib gsl2.6 and use ./configure for compilation)

4. Running the Software

Note: Please ensure that all previous steps have been followed sequentially:

1. Step 1: Place nu-sense project to lib folder (C:\MinGW\msys\1.0\home\user\nu-sense)
2. Open terminal at nu-sense folder
3. Clean residual file: *make clean*
4. Build project : *make*
5. Naviagte to bin folder: *cd bin*
6. Run the main file: *./main on* ***macOS or*** *./main.exe on* ***Windows***

Input file screenshot:

Time (24h clock)	sensor_name	value
13.2	sens1	53.4
13.2	sens2	50.0
13.2	sens3	53.7
13.2	sens4	53.2
13.2	sens5	52.8
13.2	sens6	53.3
13.2	sens7	49.7
13.2	sens8	53.1
14.2	sens1	53.2
14.2	sens2	50.0
14.2	sens3	53.7
14.2	sens4	55.3
14.2	sens5	53.5
14.2	sens6	53.3
14.2	sens7	49.6
14.2	sens8	53.3
15.2	sens1	53.2
15.2	sens2	50.0
15.2	sens3	53.7
15.2	sens4	53.2
15.2	sens5	52.2
15.2	sens6	53.6
15.2	sens7	45.7
15.2	sens8	55.1

Output File Screenshot:

13.2	53.255083
14.2	53.387622
15.2	53.281054
16.2	53.055298

13.2	53.204129
14.2	53.387622
15.2	53.281054
16.2	53.055298

13.2	53.204388
14.2	53.395727
15.2	53.428331
16.2	52.438366

Sensor History Screenshot:

Sensor1 value are out of range!
Sensor2 was stucked.

Sensor1 value are out of range!
Sensor2 value are out of range!

Sensor1 value are out of range!
Sensor2 value are out of range!

Sensor1 value are out of range!
Sensor2 value are out of range!

Sensor1 value are out of range!
Sensor2 value are out of range!

Sensor1 value are out of range!
Sensor2 value are out of range!

Sensor1 value are out of range!
Sensor2 value are out of range!
---

5. Input and Output files:

- To change input file, navigate to data folder. Go to input data further. Replace existing "sample_input.csv" with new "sample_input.csv".
- To access output, go to output data in data folder, output will be in form of "output.csv".
- To check stuck sensors, navigate to sensor history folder, output of stuck sensors and out of range sensors, use "sensor_history.csv".

6. Testing

Testing is performed using MiUnit testing. Several functions were tested as unit. To run tests. Please follow these steps:

1. Place test input values in sample_input.csv
2.Navigate to nu-sense.
3. Proceed to test folder.
4. Use *"make clean"* to clean residual files.
5. Use *"make"* to build files.
6. Check output.csv for outputs and at all it will display results for test functions.

Test results screenshot:

```
Elliminate incorrect data:
Sigma of z_i=9.916794
Abs of sigma/sensor_number=1.239599
70 percent of Abs of sigma/sensor_number=0.867719
z-0 satisfies!  fabs(z[i])=1.413948>0.867719
z-1 satisfies!  fabs(z[i])=1.302777>0.867719
z-2 satisfies!  fabs(z[i])=1.381740>0.867719
z-3 satisfies!  fabs(z[i])=1.413948>0.867719
z-4 satisfies!  fabs(z[i])=1.414124>0.867719
z-5 satisfies!  fabs(z[i])=1.321176>0.867719
z-6 does not satisfy!  fabs(z[i])=0.242938<0.867719
z-7 satisfies!  fabs(z[i])=1.426142>0.867719

Sum of z_i disregarded 9.673855STEP 7-2: Weight coefficient Omegas:
omega0=0.146162, omega1=0.134670, omega2=0.142832, omega3=0.146162, omega4=0.146
180, omega5=0.136572, omega6=0.000000, omega7=0.147422,

FINAL STEP:
The fused output is 52.988225
ALL TESTS PASSED
Tests run: 6
Fused output was added in ../data/output_data/output.csv
```