Assignment: SDLC for human activity recognition project

COMP255 Human activity recognition @MQ S2, 2019 (updated on 11th Aug 2019)

This project aims to develop a human activity recognition IoT application to evaluate students' knowledge in SDLC. This is an individual assignment. The project tasks shall be carried out individually.

Commences: Week 3.
Assignment due: 5pm, Friday of week 7 (13th Sep).
Value: 15%

Overview

The recognition of human activities has become a task of high interest for medical, military, and security applications. For instance, patients with diabetes, obesity, or heart disease are often required to follow a well-defined exercise routine as part of their treatments [5]. Therefore, recognizing activities such as walking, running, or cycling becomes quite useful to provide feedback to the caregiver about the patient's behavior. Likewise, patients with dementia and other mental pathologies could be monitored to detect abnormal activities and thereby prevent undesirable consequences [6].

In such IoT applications, proper software engineering and data engineering are especially important to manage the software development life cycle and help make data useful for machine learning models. Many software engineers are primarily interested in aggregating raw data and making it into useful, ordered and structured data formats. A typical flowchart of sensor-based human activity recognition as shown in Figure 1.

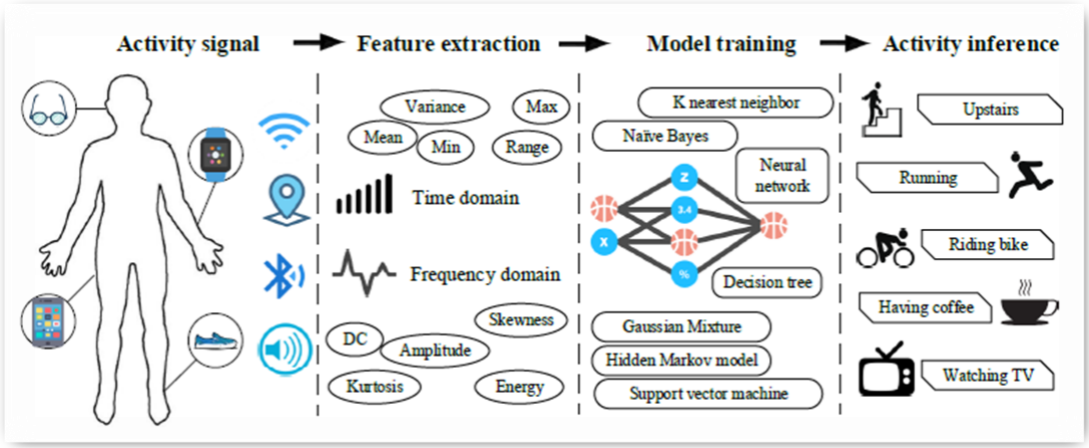

Figure 1. A typical flowchart of sensor-based human activity recognition

This assignment involves the following subtasks:

1. Use Agile to manage this IoT application development (e.g., develop backlog, create sprint, and monitor the sprint progress). The backlog and each sprint along with each week's sprint progress burndown chart shall be recorded in the final submission document.
2. Based on the given workshop materials, create python code to load data and extract corresponding features from the given dataset.
3. Test and evaluate the two given machine learning models (KNN and SVM) and application in general and record the test results and evaluation summary in the final submission document.
4. Refactor the source code according to the design pattern lecture and make the code easier to understand and extensible. The code shall be managed by GitHub and will be reviewed for this along with GitHub version control history.

The sourcing data is from a public dataset (Dalia dataset [1], which contains 6 sensors' data for 19 activities), refining that data and cleaning them up, and extracting significant features through statistical analysis for use in artificial intelligence and machine learning systems.

An example code is provided for reference. You may need to learn the use of Python libraries Numpy [2] and Pandas [3]. Machine learning modules using Scikit-learn [4] are given though having some understanding of them is recommended (we will only cover the basics of it to avoid course overlapping).

Recommended Sprints

The human activity recognition IoT system are recommended to be developed in four sprints.

1. Data loading and preprocessing: In this stage, based on the workshop materials provided, you need to firstly visualize the sensor data to get some idea of the underlying human activity pattern. Based on the given codes, apply the signal filtering and visualize the cleaned data.
2. Feature engineering for sensor data: In this stage, you need to extract features from the cleaned sensor data. In the example code, min, max, and mean values of three accelerometers in the wrist sensor are extracted as features of each human activity. In this assignment, you need to focus on feature engineering (try to extract more features from more sensors based on the Week 3 lecture note, and research how different features influence the performance of human activity recognition based IoT application). Then, you could use the GIVEN code to construct training datasets. In this stage, you could train different GIVEN machine learning models based on training feature set. The code of recognition models is GIVEN, where KNN and SVM classifier are used to learn human activity recognition.
3. Testing: After training a model, you should evaluate and test the application. Classification accuracy is a simple metric to measure the performance of a trained model. In addition, confusion matrix could clearly show the performance of our model on the recognition of each activity (Testing of Machine learning models and confusion matrix will be covered in week 4 lecture notes) . The two evaluation metrics are also GIVEN in the example code.
4. Code refactoring and Version Control: The given example code reflects the state of the art engineering for IoT. Please refactor the code to make the code easily to read/understand (e.g., comments) and extensible (those techniques for design pattern and software refactoring taught in the unit). The changes shall be reflected in the GitHub version control.

[1] https://www.mad.tf.fau.de/research/activitynet/daliac-daily-life-activities/
[2] https://www.numpy.org/devdocs/user/quickstart.html
[3] https://pandas.pydata.org/pandas-docs/stable/
[4] https://scikit-learn.org/stable/documentation.html
[5] Y. Jia, “Dietetic and exercise therapy against diabetes mellitus,” in Second International Conference on Intelligent Networks and Intelligent Systems, pp. 693–696, 2009.
[6] J. Yin, Q. Yang, and J. Pan, “Sensor-based abnormal human-activity detection,” IEEE Trans. Knowl. Data Eng., vol. 20, no. 8, pp. 1082– 1090, 2008.

The use of example code:

1. download the DaLiAc dataset from the link of reference 1. Unzip all files

into a folder called ‘dataset’ and then put the example code ‘har.py’ and the ‘dataset’ folder at the same directory.

2. Install Python3.x and libraries Numpy, Pandas, Scikit-learn and Matplotlib following the guidance in the weekly labs.

3. Run har.py

At the end of this assignment, you should submit your code (a new file or example code where you add your functions) and a report.

The structure of the report is:

Suggested headings (max. 10 pages; 10pt-12pt font size in single line spacing)

- Student details: name and SID

- Project title (you are free to give a cool name as the project can be used for many purposes)

- Introduction: description of the project.

- SCRUM Sprint and Design: give description of each key component and system architecture (can follow the given diagram but can’t be exactly same). Give description of the backlog, each sprint created and weekly sprint progress chart (burndown chart).

- Implementation: description of technologies and techniques used with respect to each of system components/functionalities described in the Design.

- Evaluation: description of experiments and discussion of results

- Discussion: Challenges, limitations and open issues.

- Version Control: give screen shop of the GitHub version control log

- Summary/conclusion: summary and/or concluding remarks

- References including Bitbucket project repository/wiki

Marking rubric:

3 marks: Agile Management and System Design

- The design is compliant with the project requirement and detailed – 1 mark
- Reasonable backlog design – 1 mark
- Reasonable Sprint design and progress – 1 mark

3 marks: Data engineering & Feature engineering

- Correct python code to load data – 1.5 marks
- Correct python code to extract features from data – 1.5 marks

3 marks: Testing and Evaluation report

- Correct python code to construct training and testing set and test given machine learning models – 1.5 marks
- Reasonable evaluation report – 1.5 marks

3 marks: Code Refactoring and Version Control

- Elegance of code – 1 mark
- Maintainability (e.g., comments) – 1 mark
- Reasonable Version control history (e.g., Screen shot from GitHub) – 1 mark

3 marks: Summary and overall clarity of the report

- Insightful summary for the project from software engineering perspective – 1 mark
- English (e.g., grammar, typos, readability, etc.) – 1 mark
- Structure including references/Length – 1 mark