

# Surface Detection by Robot Movements

Marian Dumitrascu

March 19, 2019

## Introduction

For this project I choosed a Kaggle.com open competition project. This is *CareerCon 2019 - Help Navigate Robots*. Here is the description of the project from Kaggle:

*In this competition, you'll help robots recognize the floor surface they're standing on using data collected from Inertial Measurement Units (IMU sensors). We've collected IMU sensor data while driving a small mobile robot over different floor surfaces on the university premises. The task is to predict which one of the nine floor types (carpet, tiles, concrete) the robot is on using sensor data such as acceleration and velocity. Succeed and you'll help improve the navigation of robots without assistance across many different surfaces, so they won't fall down on the job.*

The task is challenging, but I believe I can obtain a decent result using techniques and tools learned in this course.

## Report Structure

1. First I will describe data provided and the output expected
2. Then, will perform data analysis, visualization and get insights
3. pre-process and transform data
4. decide the model to use, measure its performance and tune it
5. perform the final data prediction and show the results, submit to Kaggle
6. draw some conclusions

## Report Project

I also keep this project on GitHub here: [https://github.com/mariandumitrascu/ph125\\_9\\_HelpRobotsNavigate](https://github.com/mariandumitrascu/ph125_9_HelpRobotsNavigate)  
Data loaded by the R scripts is kept on an AWS public S3 bucket, to be easily loaded.

## Data Description

Input data from Kaggle consists in 4 files:

- *X\_train.csv* and *X\_test.csv* - the input data, covering 10 sensor channels and 128 measurements per time series plus three ID columns:
  - *row\_id*: The ID for this row.
  - *series\_id*: ID number for the measurement series. Foreign key to *y\_train/sample\_submission*.
  - *measurement\_number*: Measurement number within the series.

The orientation channels encode the current angles how the robot is oriented as a quaternion (see Wikipedia). Angular velocity describes the angle and speed of motion, and linear acceleration components describe how the speed is changing at different times. The 10 sensor channels are:

- *orientation\_X*
  - *orientation\_Y*
  - *orientation\_Z*
  - *orientation\_W*
  - *angular\_velocity\_X*
  - *angular\_velocity\_Y*
  - *angular\_velocity\_Z*
  - *linear\_acceleration\_X*
  - *linear\_acceleration\_Y*
  - *linear\_acceleration\_Z*
- *y\_train.csv* - the surfaces for training set.
    - *series\_id*: ID number for the measurement series.
    - *group\_id*: ID number for all of the measurements taken in a recording session. Provided for the training set only, to enable more cross validation strategies.
    - *surface*: the target for this competition.
  - *sample\_submission.csv* - a sample submission file in the correct format.

## Data Analysis

I will make the following assumptions about observations:

- all observations are made using the same robot
- the interval between 128 observations for each series is always the same.
- the surface is a plane, no stairs, hills or valleys

From a physicist perspective there are three forces involved: gravitation force, robot propulsion force, and friction force. Gravitation force is constant. Friction force depends on the surface by a coefficient and propulsion is an unknown variable. We need to basically determine the friction coefficient based on a movement pattern. Moving objects will travel longer if the surface has a lower friction than on a surface with higher friction. On the other side, changing direction can be trickier on a surface with higher friction.

## First Insights

```
x <- c(1,3,7,9)
diff(x^2)
```

```
## [1] 8 40 32
```

**Data Distribution**

**Data Pre-Processing**

**Quaternion to Euler**

**More Data Analysis**

**Fit The Model**

**One-vs-One or One-vs-All**

**Results and Submit the Data**

**Conclusion**

One of the most important outcome of this prroject is that I learned a few things in addition to what was presented in the course.

**Reference**

1. Q2EA: Convert from rotation Quaternions to Euler Angles. Q2EA converts from Quaternions (Q) to Euler Angles (EA) based on D. M. Henderson (1977). Q2EA.Xiao is the algorithm by J. Xiao (2013) for the Princeton Vision Toolkit. <https://rdr.io/cran/RSpincalc/man/Q2EA.html>
2. Understanding Quaternions. <http://www.chrobotics.com/library/understanding-quaternions>
3. Understanding Euler Angles. <http://www.chrobotics.com/library/understanding-euler-angles>
4. Tune Machine Learning Algorithms in R (random forest case study) by Jason Brownlee. <https://machinelearningmastery.com/tune-machine-learning-algorithms-in-r/>
5. Classification with more than two classes, from Introduction to Information Retrieval, Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze,, Cambridge University Press 2008 <https://nlp.stanford.edu/IR-book/html/htmledition/classification-with-more-than-two-classes-1.html>

[http://www.starlino.com/imu\\_guide.html](http://www.starlino.com/imu_guide.html)