**Modeling the Sleeping Behaviour of a Patient Using Supervised Learning Techniques**

Aseel AlOrbani, Razan AlFar, Dimuthu Hemachandra, other profs

**Abstract**

**Introduction:**

**Data:**

**Methods:**

The data from all the sensors have been obtained as text files and as the first step, the data was converted into *panda data frames* which is popular data structure in Python programming language that is used in data analysis. Since the amount of available data was noisy (55836 readings from 7 sensors per year), due to the simultaneous activation of sensors making it difficult to be used without been filtered. Therefore, an empty data frame was defined with *Day, Counter, Status* as fields and allocated 1440 rows corresponding to each minute of a given day (60 min x 24). Then a method was implemented to go through all the data and detect the time points where the patient goes to bed (ABS bed sensor activated), and when the patient wakes us (ABS bed sensor vacated) and used those time points to fill the empty data frame with 1s for time periods of sleeps and 0s when he is awake. This leads to a data set of two classes where 1 represents minutes where the patient is on the bed and 0 represents otherwise.

Also, it was observed that the patient has not been present at home on some days and comes home in an irregular pattern making the sensor readings not interpretable since all the sensors resets at midnight. Therefore, these days were also removed from our analysis. Figure () shows a graphical representation of the status vs the counter for the selected data. To model a classifier that can predict the sleeping behaviour of this patient, the data was first divided in to sets where the odd days are considered as training data and the even days as the test data for validation. Several classifiers were tested with these data to inspect which model performs the best.

Random Forest Classifier:

Out of all the available classifiers nowadays, we used the Random Forest (RF) classifier1 since it is one of the most widely used classifiers and has outperformed other classifiers in many aspects2 3 4. RF is a collection of decision trees where each decision tree within the forests is built with a different bootstrap sample drawn from the original data set and then splitting according to the best split found over a randomly selected subset of features independently at each node. Once the forest is built, the classification can be done by simply aggregating the votes of all trees. There are few hyper parameters that you can tune in RF to improve the test accuracy5. The number of trees(n\_estimators), the maximum depth of the tree (max\_depth), the minimum number of samples required to split an internal node (min\_samples\_split) and the number of features to consider when looking for the best split (max\_features) are some of them.

**Results:**

**Discussion:**

1. Breiman L. Random Forests. *Mach Learn*. 2001;45(1):5-32. doi:10.1023/A:1010933404324

2. Fernández-Delgado M, Cernadas E, Barro S, Amorim D. Do we need hundreds of classifiers to solve real world classification problems? *J Mach Learn Res*. 2014;15(1):3133-3181.

3. Treeratpituk P, Giles CL. Disambiguating Authors in Academic Publications Using Random Forests. In: *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries*. JCDL ’09. New York, NY, USA: ACM; 2009:39-48. doi:10.1145/1555400.1555408

4. A. Criminisi JS. *Decision Forests for Computer Vision and Medical Image Analysis*. Berlin, Germany: Springer International Publishing; 2013.

5. Probst P, Wright M, Boulesteix A-L. Hyperparameters and Tuning Strategies for Random Forest. *arXiv e-prints*. April 2018:arXiv:1804.03515.