**Not to be distributed outside SPAN-PACER Lab**

**Injury Recovery Prediction**

This repository performs injury recovery prediction based on features obtained from tensor decomposition method. Number of features are first selected by setting the acc\_vs\_samples to False. The obtained number of features is then used to test the setup for acc\_vs\_samples set to True. Tucker decomposition is used for feature generation while, MLP is used to learn non-linear approximation and kNN to form a baseline.

**Requirements**

Python =2.7

Scipy

Numpy

Matplotlib

Sklearn

Tensorly

**Dataset**

This dataset is made up of three classes. Each class comprises of samples of a subject walking, during different stages of recovery from injury. The stages chosen for data were 10% recovered, 50% recovered and 95% recovered. The data consists of tri-axial accelerometer readings for the sensor mounted on heel of the subject. The data is organized in the form of tensor and each tensor is organized in the .mat file namely, recovered10.mat, recovered50.mat and recovered95.mat. The dataset has been re created. That is, the chopping procedure for each step has been re defined in accordance to the last meeting of Fall semester attended by Dr. Ivan Puchades, Tristan Scott and Kshitij Bichave.

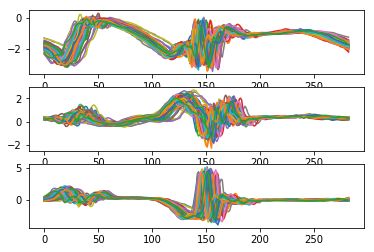


Fig. 1: Recovered - 10% data. Subplot 1: X axis, subplot 2: Y axis, subplot 3: Z axis. On the x axis of each subplot data points. On the y axis of each plot is acceleration (g).

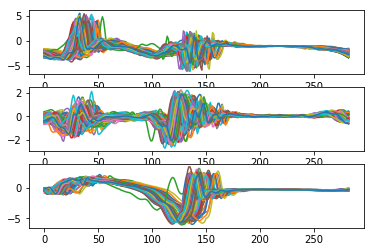


Fig. 2: Recovered - 50% data. Subplot 1: X axis, subplot 2: Y axis, subplot 3: Z axis. On the x axis of each subplot data points. On the y axis of each plot is acceleration (g).

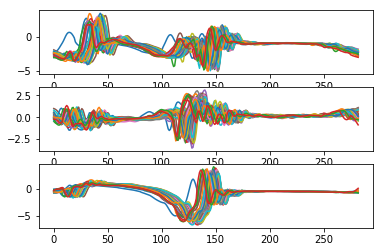


Fig. 3: Recovered - 95% data. Subplot 1: X axis, subplot 2: Y axis, subplot 3: Z axis. On the x axis of each subplot data points. On the y axis of each plot is acceleration (g).

**Method**

Run main.py to obtain results. To obtain results for acc\_vs\_reduced\_dimension set acc\_vs\_samples = False. To obtain results for acc\_vs\_samples, set acc\_vs\_samples = True. The classifiers choosen are Multi Layer Perceptron(MLP) and k-Nearest Neighbors (kNN). MLP are best to obtain non linear approximation while using multiple hidden layers while the kNN nearest neighbors represent the model based on it distance to the k nearest neighbors.