Soft Magnetic Elastomers for Continuous Force and Location Estimation in Real-time

Tess Hellebrekers, Nadine Chang, Keene Chin, Michael J. Ford, Oliver Kroemer, and Carmel Majidi

1 Repository content details

script

- *calibrate5X1Board.py*: script collect data from figure 8 motion and calcuate scales, offsets, and affine transforms.
- *sampleXYZ_Continuous.py*: script that controls uArm Swift Pro and Arduino Zero to collect magnetic signals over continuous surface.
- *preprocess5X1_XYZTF.py*: script that takes raw data from txt file and outputs filtered data into pickle file.
- *nn_regression_F*: script evaluating final NN model for F estimation on collected data.
- *nn_regression_F_gridsearch.py*: script evaluating sweep across parameters to determine best parameters for force on collected data.
- *nn_regression_XYZ*: script evaluating final NN model for XYZ estimation on collected data.
- *nn_regression_XYZ_gridsearch.py*: script evaluating sweep across parameters to determine best parameters for XYZ on collected data.
- *collectContinuousData.py*: script for collect hand written data from magnetometer.
- *mag_class.py*: script for preprocessing data and training SVMs.
- *run_demo.py*: script for running trained SVM in a real-time demo.

data

- *[SampleName].txt*: raw data for that [SampleName] collected over the surface.

- *[SampleName] SVD.pkl*: filtered data for that [SampleName] collected over the surface.

- *[SampleName] 5X1 Board Calibration.npz*: holds raw data from calibration, affine transforms, offsets, and scales for the reference board to the main board.

- *base_data_batch*: all data batches of baselines, which were averaged to get average baseline signal.

- *collect_data_batch{#}*: data batch on each digit $0-9$. batch$(n-1)$ correlates to digit n (ex: batch1 contains data for digit 0).

- *digits_idx_final.pkl*: pickle contains index for each digit signal series from collect_data_batch files.

- *digits_signals_final.pkl*: pickle contains the corresponding signal series for each digit sample.

- *best_est.joblib*: best svm estimator saved.

2 Details on preprocessing and SVMs

We demonstrate a simple task of classifying digits through the soft skin, which illustrates the ability of identifying meaningful change through temporal space. We first collect a set of digits data from the magnetometer by drawing the numbers 0 through 9. The data was collected at 50 Hz. In order to extract the signals correlated with the digit, we performed a number of preprocessing steps. For the following steps, we assumed our raw magnetometer signal to be S_i^d for digit $d \in \{0, , 9\}$ and time $i \in \{1, , t\}$. We denoted the raw magnetometer signal without any interference as S^b, defined by averaging signals of resting states over a set of 5 experiments.

1) We defined a positive signal $S_j^d = S_i^d$ if $\delta(S_i^d, S^b) > .2S^b$. Note that due to filtering, time j was no longer consecutive after this step. 2) For each S_j^d, if $\delta(j, j \pm 1) > 3$ time steps away, we deemed element S_j^d to be noise and was removed. 3) We clustered the neighboring time data points together, where a bucket B consists of a series of non-consecutive $S_{j:j+n}^d$. A bucket was determined if $\delta(j+n, j+n+1) > 7$. 4) Different digits required different amount of time to write. In order to compensate for this variable, we select a fixed time length l. Based on the median and $80th$ percentile time lengths for all buckets B, we set $l = 19$, since the maximum median is 19 and maximum $80th$ percentile is 21.8, close to our median. 5) For each unique bucket $S_{j:j+n}^d$ and $l = 20$, we set an anchor $a = \text{midpoint}(i, i+n)$. To make a consecutive series, all time points i from $a - l/2$ to $a + l/2$ were selected as the final time points, such that the final data point for digit d is $X^d = S_{a-l/2:a+l/2}^d$.

The final data X^d for all d digits were utilized to train, cross validate and test a classification model. Each X^d was flattened into a one dimensional vector. We performed grid search over parameters ($l1$ or $l2$ penalty, $C = \log_{10} n$ with $n \in (0, , 10)$) for a linear SVM, with five fold cross validation on training and testing splits and an additional five fold cross validation on training and validation split. Our final model was chosen with a squared hinge loss, $l2$ penalty, and $C = 0.02636$. Our final accuracy is 92.86% on our held-out test set.