Deep Dark Fantasy on MathWork Challenge

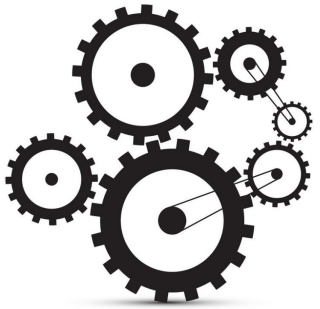
Yuzhe Yang*, Siyuan Liu, Laiyuan Zhang, and Sicheng Yao

University of Southampton

School of Electronic and Computer Science

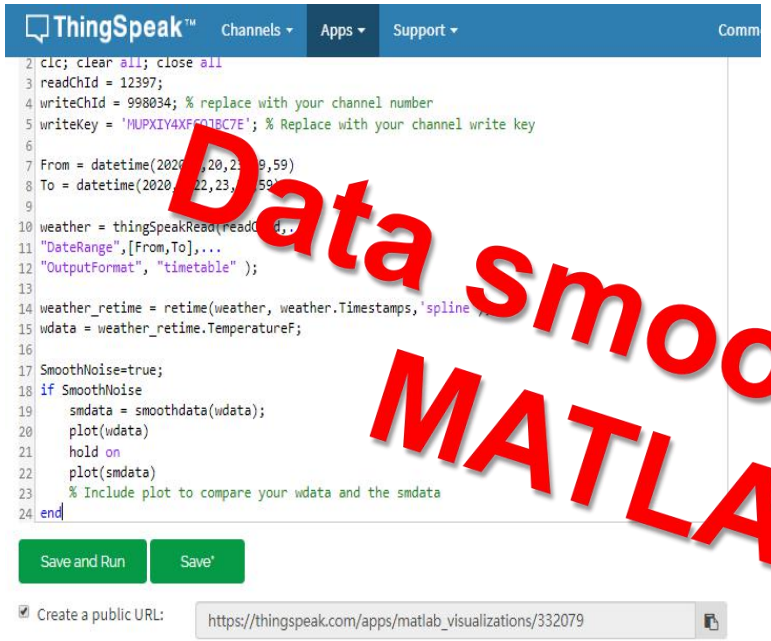
23/02/2020

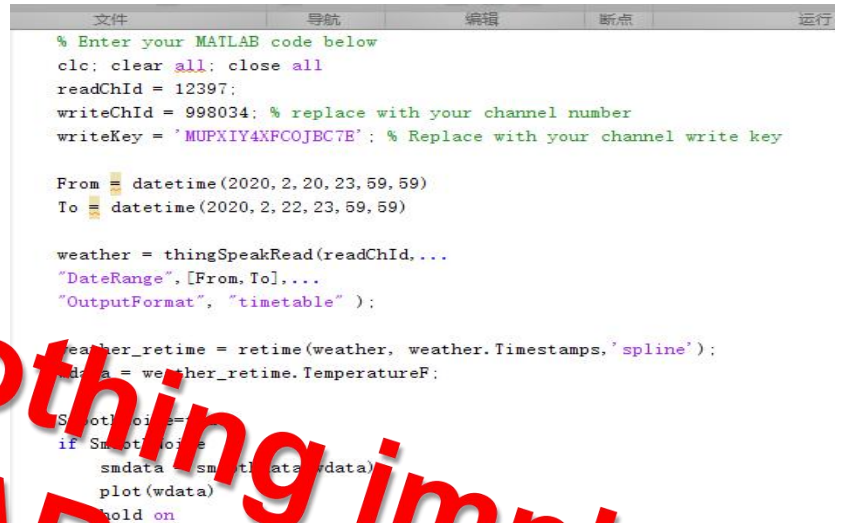
What we have done so far?

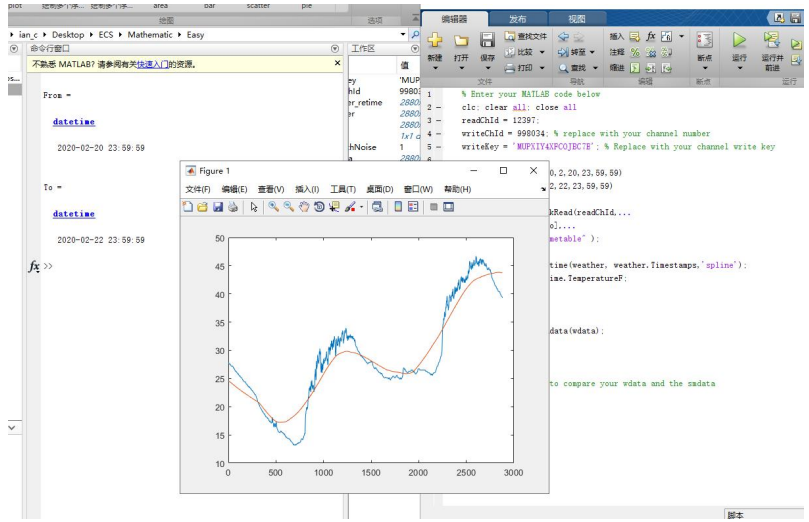


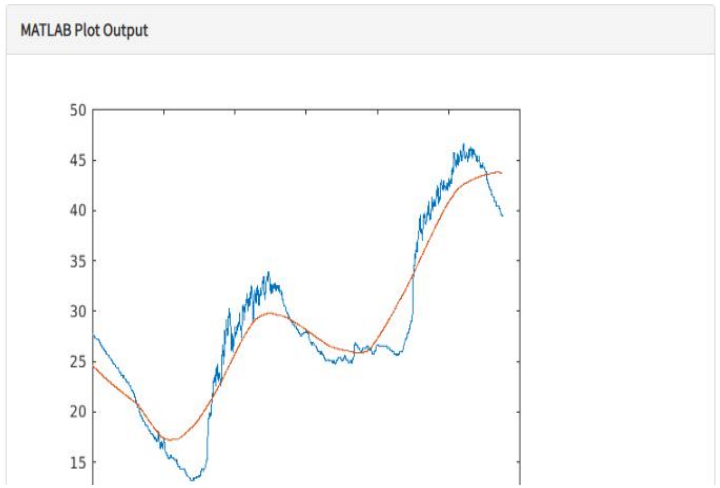
- 1. We have finished challenges.
- 2. We collect our own training and testing data via MATLAB mobile.
- 3. We have visualized our newly build-up data and analyze potential features.
- 4. We use the build-in machine learning algorithm to classify human activities.
- 5. We adapt our model into MATLAB mobile to analyze the real time data.
- 6. Have eaten many delicious food and drinks and played many games!

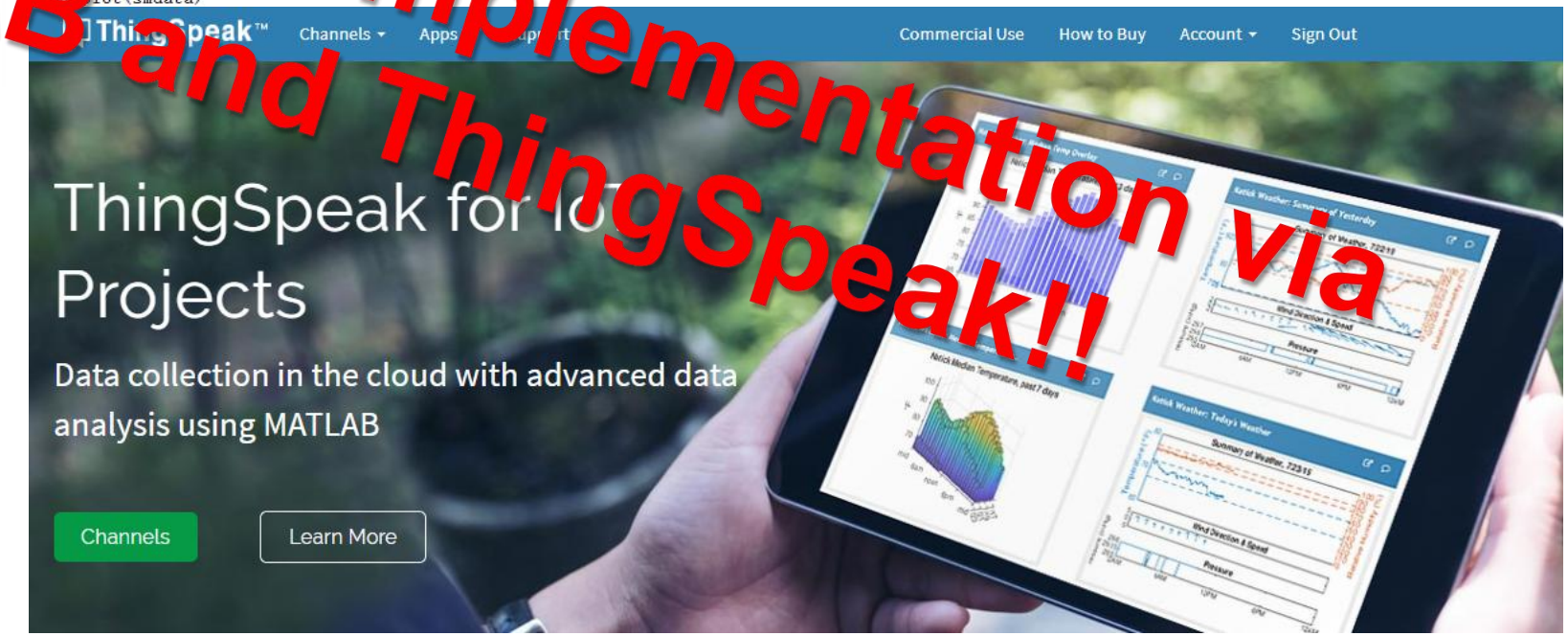
Data smooth in Easy Challenge





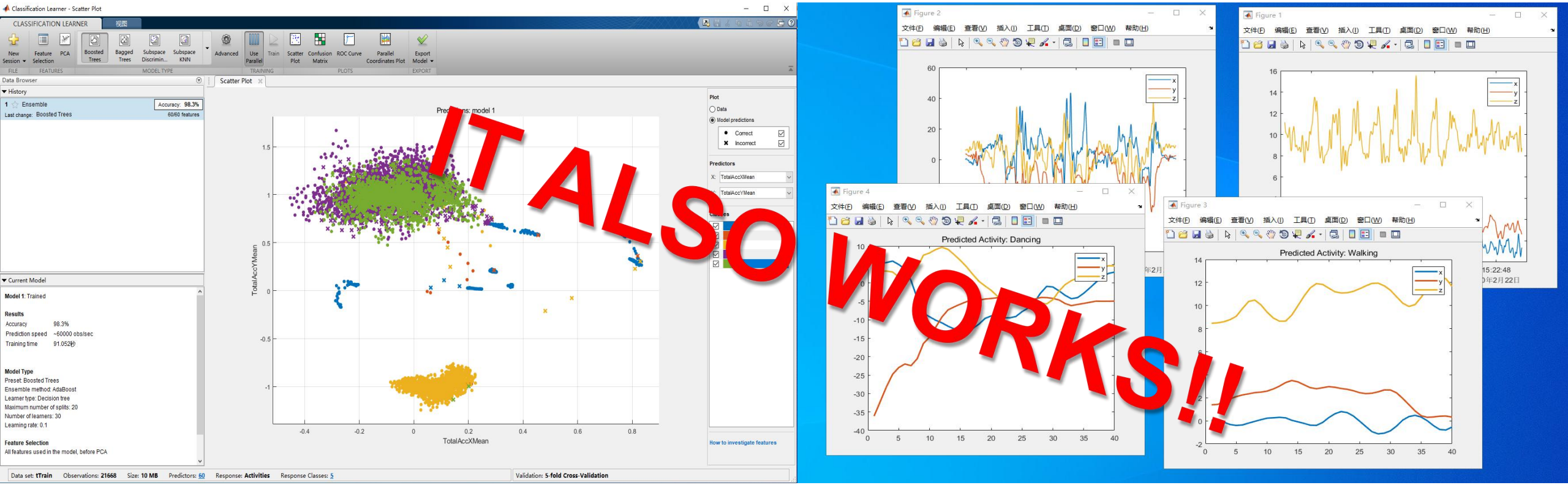






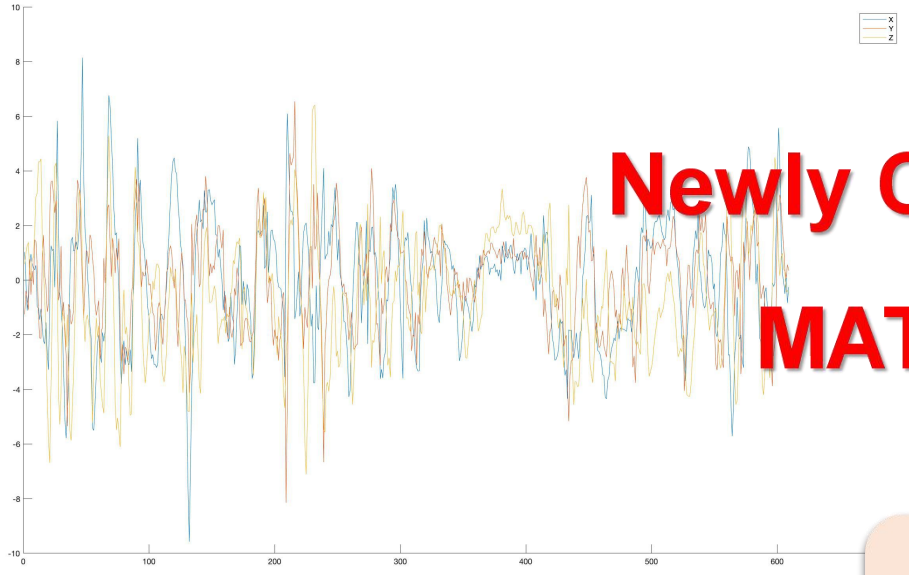
Data smoothing implementation via MATLAB and ThingSpeak!!

Train on provided dataset and predict on our raw data via MATLAB

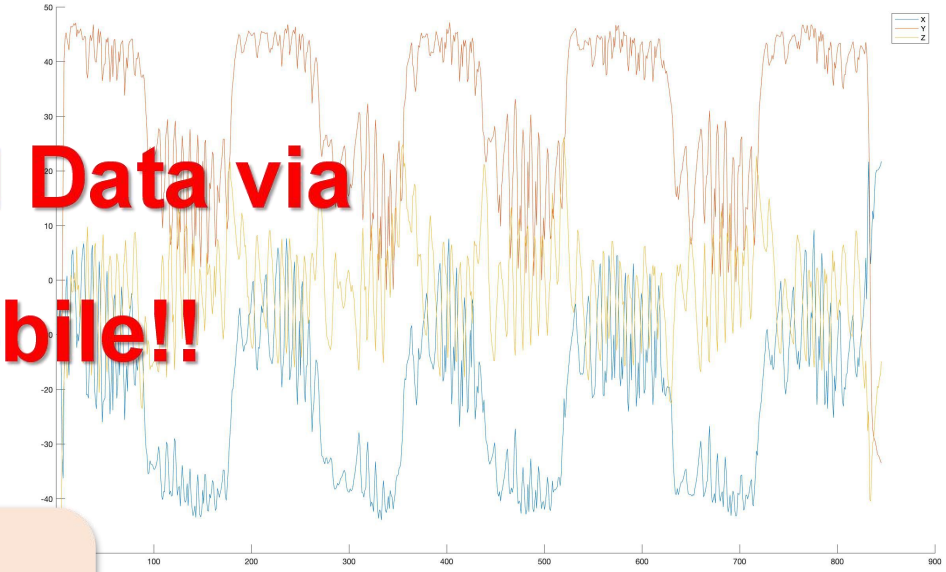


Training and Predicting

Newly Collected Data via MATLAB Mobile!!

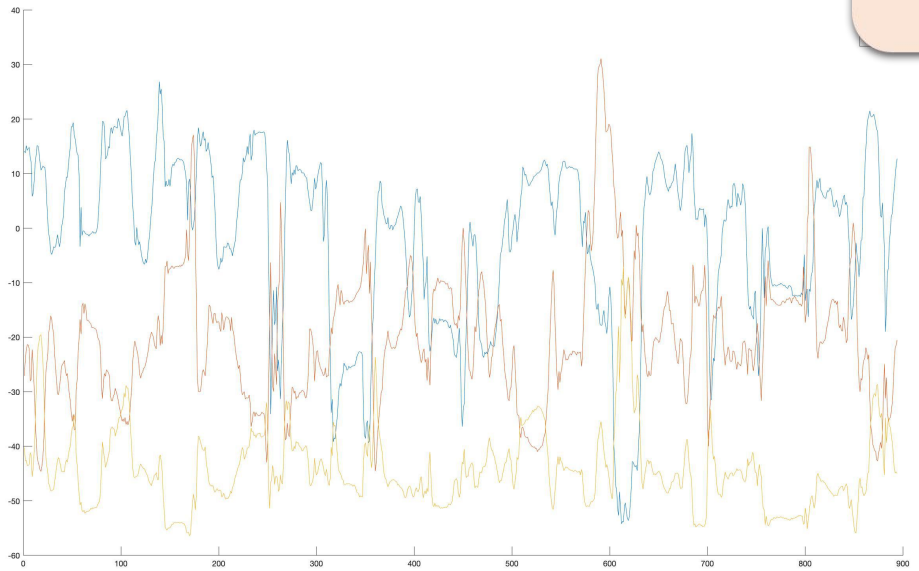


Dancing Magnitude

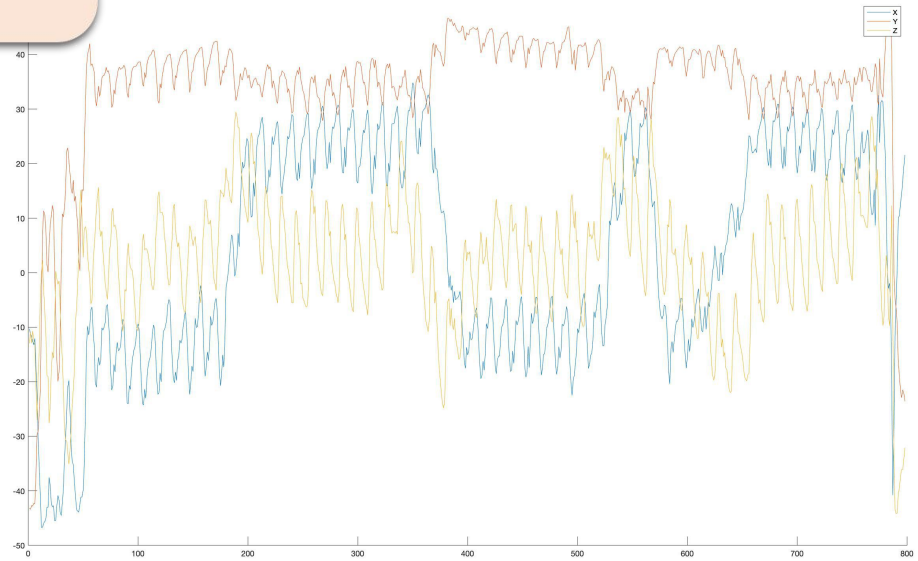


Running Magnitude

Magnitude



Sitting and Standing Magnitude



Walking Magnitude

Classification Learner - Scatter Plot

CLASSIFICATION LEARNER | 视图

New Session | Feature Selection | PCA | Boosted Trees | Bagged Trees | Subspace Discrimin... | Subspace KNN | Advanced | Use Parallel | Train | Scatter Plot | Confusion Matrix | ROC Curve | Parallel Coordinates Plot | Export Model

FILE | FEATURES | MODEL TYPE | TRAINING | PLOTS | EXPORT

Scatter of our newly collected data

Data Browser

History

1.1 Ensemble Last change: Boosted Trees	Accuracy: 85.3% 3/3 features
1.2 Ensemble Last change: Bagged Trees	Accuracy: **92.6%** 3/3 features
1.3 Ensemble Last change: Subspace Discriminant	Accura...5.0% ...tures
1.4 Ensemble Last change: Subspace KNN	Accura...7.9% ...tures
1.5 Ensemble Last change: RUSBoosted Trees	Accura...2.1% 3/3 features
2.1 Linear Discriminant Last change: Linear Discriminant	Accuracy: 55.2% 3/3 features
2.2 Quadratic Discriminant Last change: Quadratic Discriminant	Accuracy: 7...2% 3/3 features
3.1 KNN Last change: Fine KNN	Accuracy: 91.9% 3/3 features
3.2 KNN Last change: Medium KNN	Accuracy: 90.0% 3/3 features
3.3 KNN Last change: Coarse KNN	Accuracy: 86.6% 3/3 features
3.4 KNN Last change: Cosine KNN	Accuracy: 88.4% 3/3 features
3.5 KNN Last change: Cubic KNN	Accuracy: 90.7% 3/3 features
3.6 KNN Last change: Weighted KNN	Accuracy: 92.0% 3/3 features

92.6%

our newly collected data

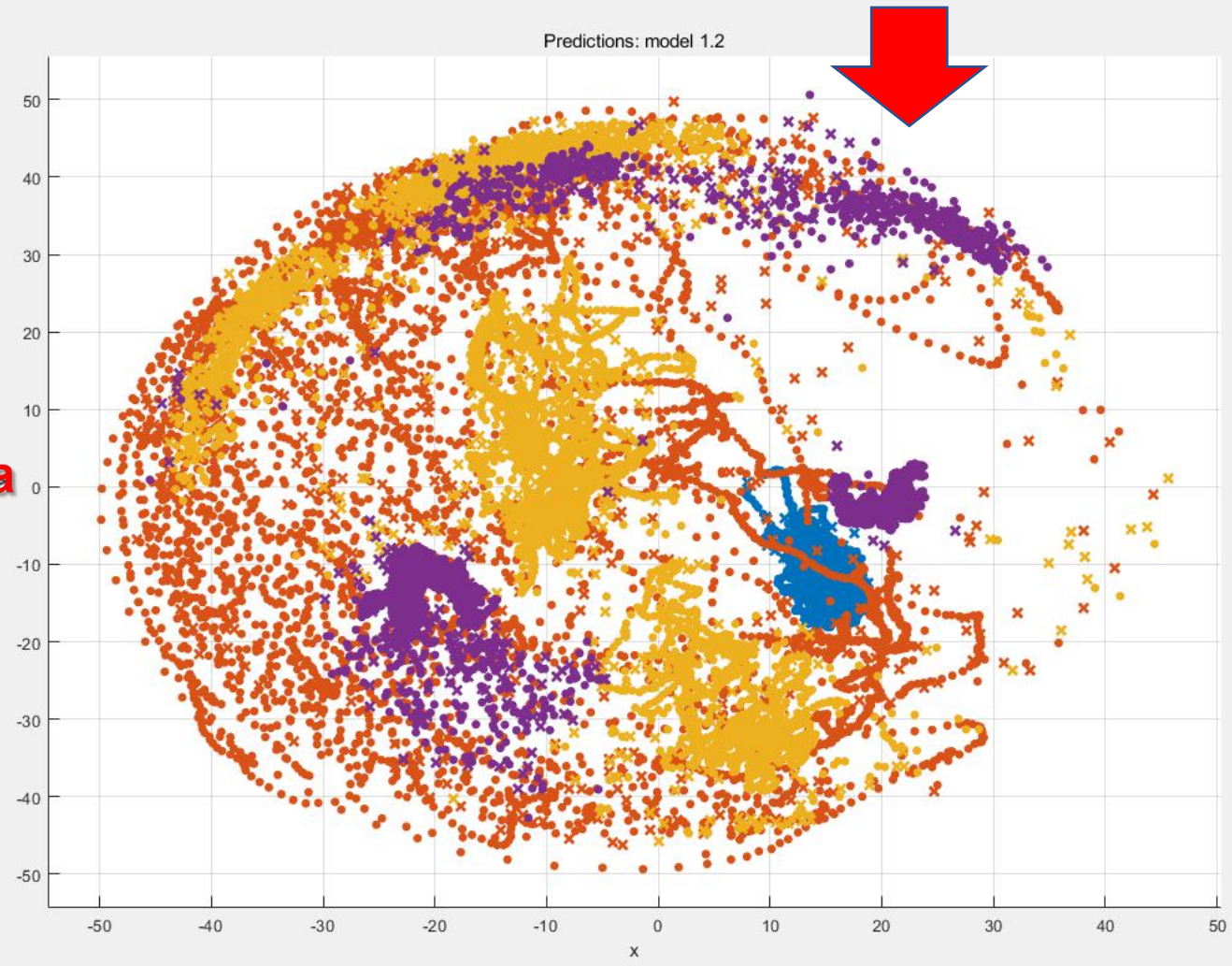
Current Model

Model 1.2: Trained

Results

Accuracy 92.6%

Scatter Plot



Plot
- Data
- Model predictions
  - Correct
  - Incorrect

Predictors
- X: x
- Y: y

Classes
- 1
- 2
- 3
- 4

How to investigate features

Data set: **tTrain_ours** Observations: **20000** Size: **627 kB** Predictors: **3** Response: **Activities** Response Classes: **4**

Validation: **5-fold Cross-Validation**

在这里输入你要搜索的内容

英 23:57 2020/2/22

Human Action Classification via “Sylvan Gated Bagged Trees”

```
clear all; clc; close all; % Keep super clean.
interval = 8; % How long it is between two detections, in seconds.
decetion_times = 8; % How many times do you want to detect.
trainedClassifier = load('trainedModelBagTree'); % Load .mat file from MATLAB Deive.
for external_loop = 1:1:decetion_times+1
    if external_loop ~= 1 % Skip the first loop for initialisation.
        [a,t] = magfieldlog(m); % a is the input matrix for the model.
        % The size of a should be equal to interval * frequenc,
        % however due to the limited performance on mobile devices,
        % it is apt to be marginally less than the estimation value.
        clear m; % Empty the log variable in every loop for saving memory.
        scattered_coe = abs((max(a)-min(a))/mean(a))
        disp(datestr(now))
        a = array2table(a);
        featlabels_test = {'x', 'y', 'z'};
        a.Properties.VariableNames = featlabels_test;
        pred = trainedClassifier.trainedModelBagTree.predictFcn(a);
        pred = pred';
        pred = pred(:)';
        tab_pred = tabulate(pred);
        if scattered_coe >= exp(-1)
            disp(tab_pred)
        else
            disp('User lost himself.');
        end
    end
    m = mobiledev; % Turn on and initialise the receiver.
    m.MagneticSensorEnabled = 1; % Ensure the magnetic sensor is on.
    m.Logging = 1; % Start logging.
    tnit_time=clock; % Start timing.
    % Timer.
    for inner_loop = 1:1:1073741824 % Very bad design.
        if etime(clock,tnit_time)>interval
            break
        end
    end
    fprintf('\n')
end
```

Gated Function:

if (max_value + min_value)/mean_value >1/e

- **Motivation:**

Use nature principle to solve nature problems

- **Contribution:**

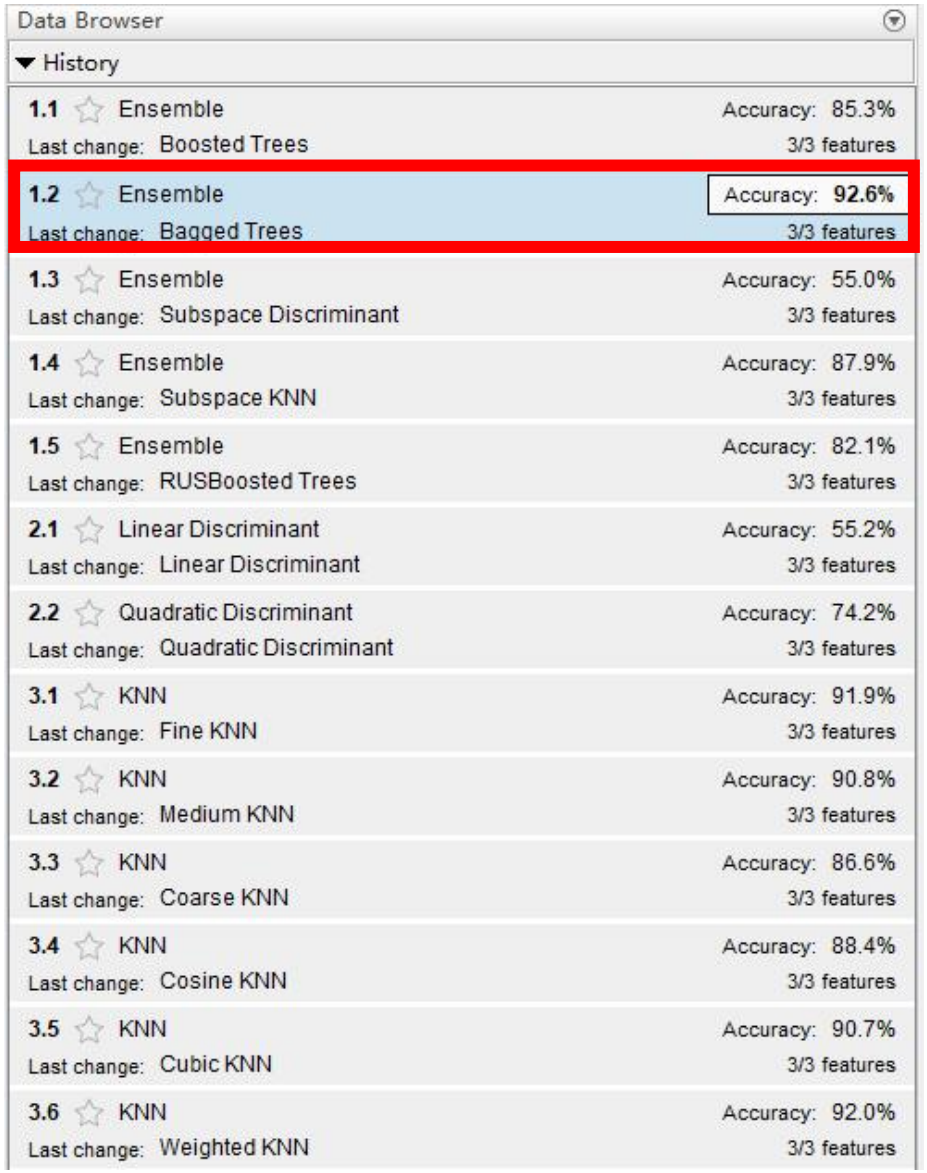
Eliminate misclassification

Results illustration

```
    0.7163

23-Feb-2020 11:20:13
    1.0000    4.0000    5.1948
    2.0000   20.0000   25.9740
    3.0000   53.0000   68.8312


scattered_coe =

    3.4422

23-Feb-2020 11:20:24
    1.0000         0         0
    2.0000   74.0000   92.5000
    3.0000         0         0
    4.0000    6.0000    7.5000


scattered_coe =

    0.2766

23-Feb-2020 11:20:32
User lost himself.


scattered_coe =

    3.7991

23-Feb-2020 11:20:45
    1.0000         0         0
    2.0000   63.0000   81.8182
    3.0000   11.0000   14.2857
    4.0000    3.0000    3.8961

 >> Enter command here...
```

Running Results on MATLAB mobile

Confusion Matrix



Cross-Validation Accuracy from Different Models

Acknowledge!