IOWA STATE UNIVERSITY
Department of Computer Science

Prediction of the individual with over-weighted based on the center of mass acceleration data during the gait cycle.

Yu-Pin Liang

Apr.30, 2021

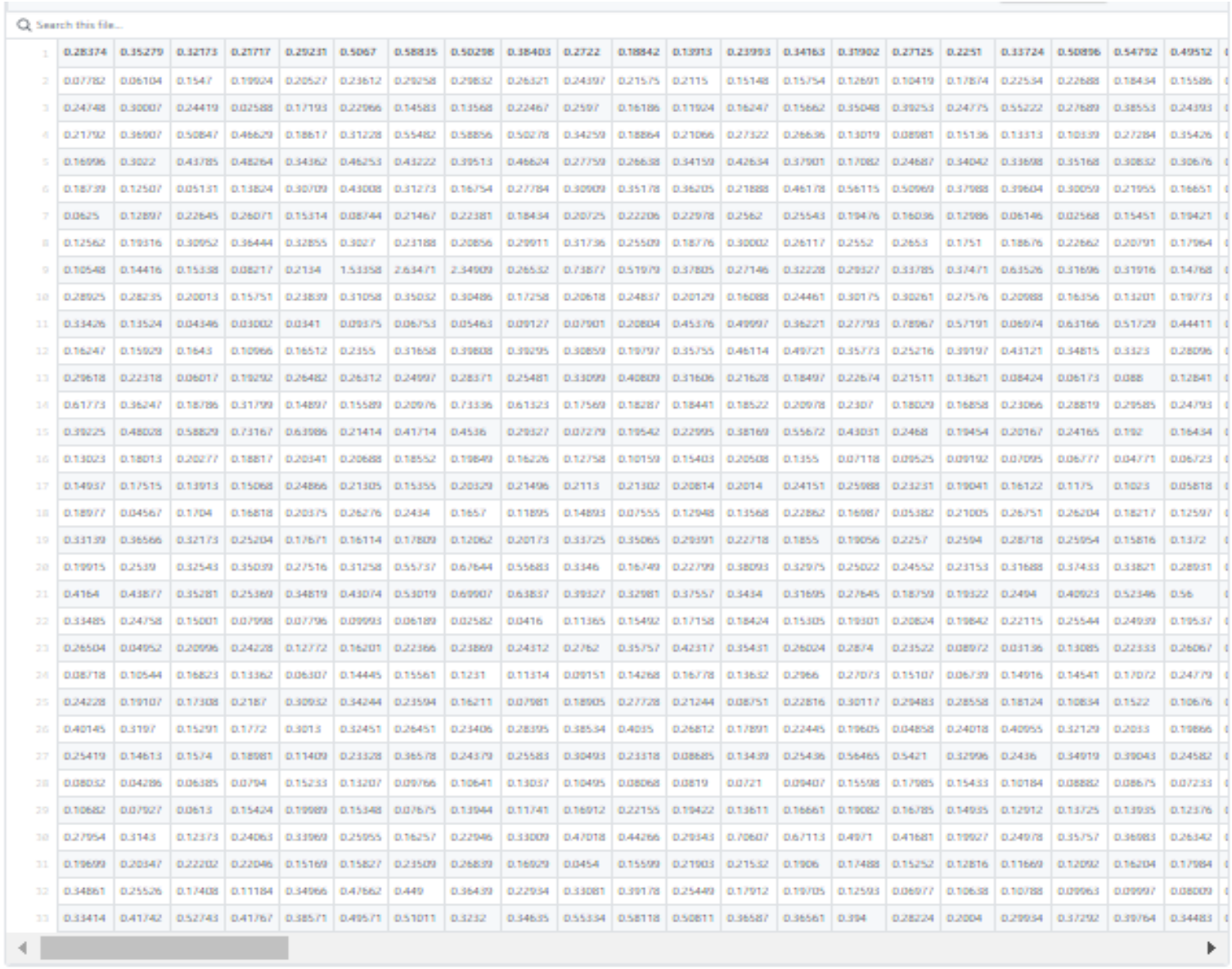
Background

- What is the background of your task?
  - The prevalence of obesity is increasing over the decade.
- What kind of data you have?
  - Acceleration time-series data from IMU, A single IMU place at L5
- What problem you want to solve?
  - Use wearable sensor detect the obesity from gait data

Datasets

- What is the dataset?
  - Acceleration time-series data from IMU
- What features you use?
  - Acceleration X,Y,Z
- What is the expected output by your machine learning models.
  - Obesity or Not (0,1)

Datasets



Related work

- In Biomechanics, lots of researchers were still engaging in running kinematics. As the growing human age, the running kinematics changed due to the decrease of the muscle mass. This kind of change was detected by using classical inferential statistics. However, the traditional methodology was time-consuming and with insufficient sensitivity. Data mining techniques have been applied in recent biomedical studies to solve this problem using a more general approach which is like general ML model.

Your method

- Here you talk about your methods
  - Compare the accuracy between SVM model (without consider time-stamp)with LSTM (consider time-stamp)
- Data processing
  - Sampling rate : IMU data 128 Hz
  - Filtered with a 2nd order, zero-lag, and low-pass Butterworth filter with a 12 Hz cutoff frequency. 80% of Training data, 20% of Testing Data

Algorithm details- SVM

```
import numpy as np
import sklearn, sklearn.datasets, sklearn.utils, sklearn.model_selection, sklearn.svm
import pandas as pd
from sklearn import metrics
from sklearn.metrics import classification_report


url = 'https://raw.githubusercontent.com/yu-pin-liang/COMS-574-ML_finalProject/main/dataset.csv'
data = pd.read_csv(url,sep='delimiter', header=None,engine='python',delimiter = ',')  # Dataset
X = np.array(data)

y  = X[:,101]
X = X[:,0:100]

X, y = sklearn.utils.shuffle(X, y, random_state=1)
X_train, X_test, y_train, y_test = sklearn.model_selection.train_test_split(X, y, test_size=0.2, random_state=1)

clf = sklearn.svm.SVC(kernel='rbf',random_state=1)


clf.fit(X_train, y_train)


#print (X_train, X_test, y_train, y_test)

y_pred = clf.predict(X_test)







print("Accuracy:",metrics.accuracy_score(y_test, y_pred))



print(classification_report(y_test, y_pred))
```

Algorithm details- LSTM

```
43
44 -    inputSize = 3;
45 -    numHiddenUnits = 100;
46 -    numClasses = 2;
47
48 -    layers = [ ...
49          sequenceInputLayer(inputSize)
50          bilstmLayer(numHiddenUnits,'OutputMode','last')
51          fullyConnectedLayer(numClasses)
52          softmaxLayer
53          classificationLayer]
54
55 -    maxEpochs = 20;
56 -    miniBatchSize = 6;
57
58 -    options = trainingOptions('adam', ...
59          'ExecutionEnvironment','cpu', ...
60          'GradientThreshold',1, ...
61          'MaxEpochs',maxEpochs, ...
62          'MiniBatchSize',miniBatchSize, ...
63          'SequenceLength','longest', ...
64          'Shuffle','never', ...
65          'Verbose',0, ...
66          'Plots','training-progress');
67 -    net = trainNetwork(XTrain,YTrain,layers,options);
68
69
70
71 -    numObservationsTest = numel(XTest);
72 -    for i=1:numObservationsTest
73 -        sequence = XTest{i};
74 -        sequenceLengthsTest(i) = size(sequence,2);
75 -    end
76 -    [sequenceLengthsTest,idx] = sort(sequenceLengthsTest);
77 -    XTest = XTest(idx);
78 -    YTest = YTest(idx);
79
80
81
82 -    miniBatchSize = 6;
83 -    YPred = classify(net,XTest, ...
84          'MiniBatchSize',miniBatchSize, ...
85          'SequenceLength','longest');
86
87 -    acc = sum(YPred == YTest)./numel(YTest)
```

Experimental setups

- Learning rate:0.001
- Number of epochs : 100
- Hardware: CPU

Experimental results

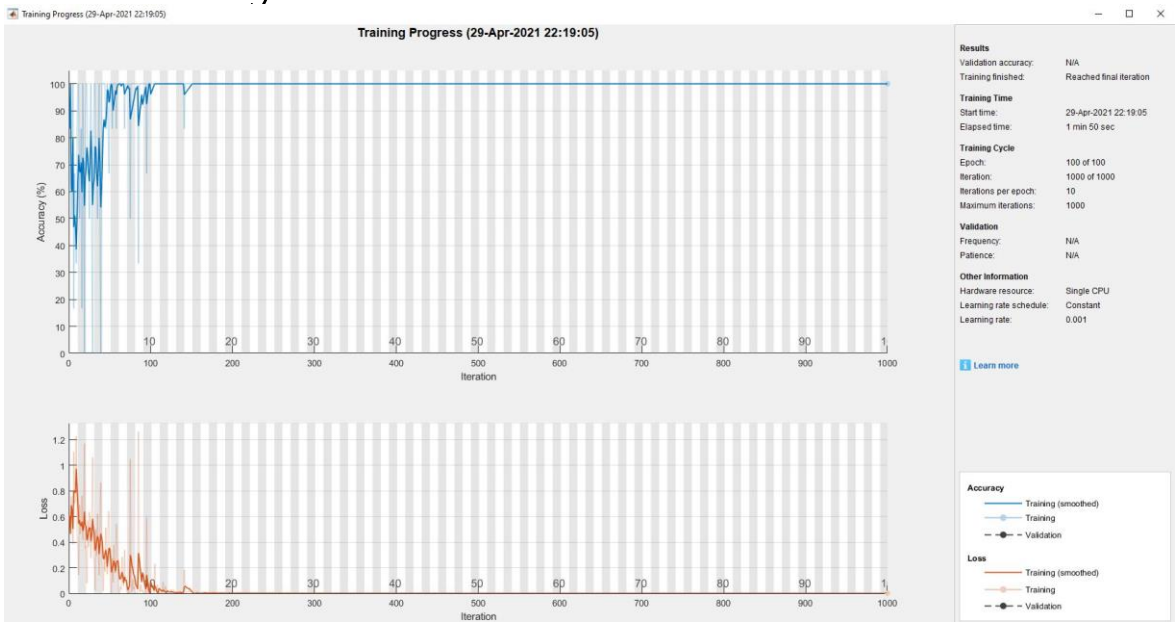
Accuracy from SVM

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))


print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.8571428571428571
```

Accuracy from LSTM




```
86
87 -    acc = sum(YPred == YTest)./numel(YTest)
88
```

Command Window

```
acc =

    0.9333
```

Experimental results

- Are the results meet your expectations?
  - LSTM had higher accuracy rate compare with the SVM
  - Both of them have high accuracy

Conclusion and Future Work

1. Add more features to make this model increased the accuracy.
2. Learn how to predict the time-series result (predict the muscle force at each moment), not only just a classifier