DATA1030 Project Proposal -- Hand Gesture Recognition

Cangcheng(Joseph) Tang
Banner ID: B01628536

1.Obejective

- The target variable is different "Class" for gestures and "User". This project's main purpose is to train a model to recognize hand gestures based on the coordinates of 12 joints of the hand. There is also an exploratory objective: identifying users.

- As the target variable is categorical, not continuous, classification methods and models will be used to solve the problems above.

- Hand gesture recognition is a very powerful tool in interaction with an electronic device. Such technology allows hands-free control over smartphones, computers, or even smart household appliances. So remote controls will no longer be needed. Also, accessibility features with hand gesture recognition can do more than reading things very loudly.
  Many smartphone companies, including LG and Google, have already been doing this, and it is going to be a trend in human-computer interaction.

2. Dataset Description

- There are 78095 data points and 38 features:
  5 types of hand postures from 14 users were recorded using unlabeled markers on fingers of a glove in a motion capture environment. Due to resolution and occlusion, missing values are common.

- Data collection:
  A rigid pattern of markers on the back of a glove was used to establish a local coordinate system for the hand, and 11 other markers were attached to the thumb and fingers of the glove. 3 markers were attached to the thumb with one above the thumbnail and the other two on the knuckles. 2 markers were attached to each finger with one above the fingernail and the other on the joint between the proximal and middle phalanx.

- Features Information:
  - 'Class' - Integer. The class ID of the given record. Ranges from 1 to 5 with 1=Fist(with thumb out), 2=Stop(hand flat), 3=Point1(point with pointer finger), 4=Point2(point with pointer and middle fingers), 5=Grab(fingers curled as if to grab).
  - 'User' - Integer. The ID of the user that contributed the record. No meaning other than as an identifier.
  - 'Xi' - Real. The x-coordinate of the i-th marker position. 'i' ranges from 0 to 11.
  - 'Yi' - Real. The y-coordinate of the i-th marker position. 'i' ranges from 0 to 11.
  - 'Zi' - Real. The z-coordinate of the i-th marker position. 'i' ranges from 0 to 11.
    i = 1 Pinky Finger (Joint);
    i = 2 Pinky Finger (Nail);
    i = 3 Ring Finger (Joint);
    i = 4 Ring Finger (Nail);
    i = 5 Middle Finger (Joint);
    i = 6 Middle Finger (Nail);
    i = 7 Pointer Finger (Joint);
    i = 8 Pointer Finger (Nail);
    i = 9 Thumb (Metacarpophalangeal Joint);
    i = 10 Thumb (Interphalangeal Joint);
    i = 11 Thumb (Nail)

- Public Paper
  - *A. Gardner, J. Kanno, C. A. Duncan, and R. Selmic. 'Measuring distance between unordered sets of different sizes,' in 2014 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), June 2014, pp. 137-143.*
    In this paper, the authors introduced a new parameterizable metric, which measures the distance between unordered sets. This metric can be useful in premetrics, multisets, and multiple minimum-cost mappings from each set's perspective. Hand gesture recognition is used to demonstrate the function's usage in machine learning and pattern recognition.
  - *Gardner, A., Duncan, C.A., Kanno, J. and Selmic, R., 2014, October. 3d hand posture recognition from small unlabeled point sets. In 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 164-169). IEEE.*
    This paper compared the performance of various classification models when recognizing static hand postures. The researchers found that aggregate feature classifiers have balanced performance among different users, but their accuracy is not the best. And pseudo-rasterization is the one performed best among all the classification models tested. The difficulty of user classification is also stated.

3. Data Preprocess

- For the coordinate data, MinMaxScaler is used, because there are clear boundaries in coordinates. As the size of human's hands are limited.
- When predicting "Class", "User" will be dropped, and "Class" will be the target variable.
  Train, cv, and test will be split based on different Users.
- When predicting "User", "Class" will be dropped, and "User" will be the target variable.
  Train, cv, and test will be split based on different "Class" of gestures.

Unprocessed data

Class	User	X0	Y0	Z0	X1	...
1	0	54.263880	71.466776	-64.807709	76.895635	...
1	0	56.527558	72.266609	-61.935252	39.135978	...
1	0	55.849928	72.469064	-62.562788	37.988804	...
1	0	55.329647	71.707274	-63.688956	36.561862	...
1	0	55.142400	71.435607	-64.177303	36.175817	...

Preprocessed data for predicting gestures

"User" column is dropped.
MinMaxScaler is used to preprocess coordinate data

Class	X0	Y0	Z0	X1	Y1	...
1	0.5453203	0.6346098	0.2580551	0.6278134	0.5203832	...
1	0.5529020	0.6376009	0.2700179	0.5021060	0.6708459	...
1	0.5506325	0.6383580	0.2674044	0.4982869	0.6711944	...
1	0.5488899	0.6355092	0.2627143	0.4935364	0.6683312	...
1	0.5482628	0.6344933	0.2606805	0.4922512	0.6671603	...

Preprocessed data for predicting users

"Class" column is dropped.
MinMaxScaler is used to preprocess coordinate data

User	X0	Y0	Z0	X1	Y1	...
1	0.5453203	0.6346098	0.2580551	0.6278134	0.5203832	...
1	0.5529020	0.6376009	0.2700179	0.5021060	0.6708459	...
1	0.5506325	0.6383580	0.2674044	0.4982869	0.6711944	...
1	0.5488899	0.6355092	0.2627143	0.4935364	0.6683312	...
1	0.5482628	0.6344933	0.2606805	0.4922512	0.6671603	...

Code Part, to be deleted in PDF

[11]
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder,
MinMaxScaler, StandardScaler, LabelEncoder

df = pd.read_csv('Postures.csv')
df = df.drop(0) # Dropping initial dummy record
df = df.replace('?', np.NaN) # Replacing missing value with NaN
df.head()
```

	Class	User	X0	Y0	Z0	X1	
1	1	0	54.263880	71.466776	-64.807709	76.895635	42
2	1	0	56.527558	72.266609	-61.935252	39.135978	82
3	1	0	55.849928	72.469064	-62.562788	37.988804	82
4	1	0	55.329647	71.707275	-63.688956	36.561863	81
5	1	0	55.142401	71.435607	-64.177303	36.175818	81

5 rows × 38 columns

[12]
```
# Use MinMaxScaler to preprocess coordinate data
scaler_minmax = MinMaxScaler()
coord_column = df.columns[2:]
df[coord_column] = scaler_minmax.fit_transform(df[coord_column])
```

[13]
```
# Dropping User for predicting Class
df_pred_class_processed = df.drop(columns = "User")
df_pred_class_processed.head()
```

	Class	X0	Y0	Z0	X1	Y1	
1	1	0.545320	0.634610	0.258055	0.627813	0.520383	0.34
2	1	0.552902	0.637601	0.270018	0.502106	0.670846	0.43
3	1	0.550633	0.638358	0.267404	0.498287	0.671194	0.42
4	1	0.548890	0.635509	0.262714	0.493536	0.668331	0.41
5	1	0.548263	0.634493	0.260681	0.492251	0.667160	0.41

5 rows × 37 columns

[14]
```
# Dropping Class for predicting User
df_pred_user_processed = df.drop(columns = "Class")
df_pred_user_processed.head()
```

	User	X0	Y0	Z0	X1	Y1	
1	0	0.545320	0.634610	0.258055	0.627813	0.520383	0.34
2	0	0.552902	0.637601	0.270018	0.502106	0.670846	0.43
3	0	0.550633	0.638358	0.267404	0.498287	0.671194	0.42
4	0	0.548890	0.635509	0.262714	0.493536	0.668331	0.41
5	0	0.548263	0.634493	0.260681	0.492251	0.667160	0.41

5 rows × 37 columns