Arizona State University

Ira A. Fulton Schools of Engineering

CSE 572 Data Mining

Activity Recognition Project Portfolio Report

Arizona State University

Shashi Kiran Chilukuri

ASU ID: 1217746764

Date Due: March 03, 2020
Date Submitted: March 03, 2020

Table of Contents

Introduction

The Activity Recognition project comes under CSE 572: Data Mining course for 2020 Spring A semester. The aim of this project is to develop a computing system that can understand the human activities, specifically, to identify human eating action when mixed with other unknown activities. To develop this system, 30 user recorded wristband sensors activity data was provided along with the actual activity information of each of those users. Using this real-world raw data, need to build the models that can accurately predict an eating activity when given with other activity. To get to this, this project is divided into five phases:

- Phase 1: Data Cleaning and Organization
- Phase 2: Feature Extraction
- Phase 3: Feature Selection
- Phase 4: User dependent Analysis
- Phase 5: User independent Analysis

In phase 1, using given raw wristband sensor data and corresponding actual activity information of each of the users, need to establish features and ground truth by performing the data cleaning and organization.

In phase 2, need to select and implement five existing feature extraction methods such as Fast Fourier Transform, Discrete Wavelet Transform, a set of statistical features (min, max, avg, std, RMS, energy function), etc. The aim of this task is to use features that show clear distinction between eating vs non-eating actions.

In phase 3, need to reduce the feature space and keep only those features which show maximum distance between the two classes (eating and non-eating). To achieve this, need to use Principal Component Analysis technique.

In phase 4, perform user dependent analysis i.e. split the new feature set obtained from PCA output at user level into two parts, training and test data. Then combine these training (and also testing) data of all the users to obtain the train and test data respectively. Now, train three machine learning models with the training data and check the performance of all three models with the test data. Three models to use are Decision Trees, Support Vector Machines (SVM), Artificial Neural Networks (ANN).

In phase 5, perform user independent analysis i.e. split the new feature set obtained from PCA output of all users into two parts, training and test data. Now, train three machine learning models with the training data and check the performance of all three models with the test data. Three models to use are Decision Trees, Support Vector Machines (SVM), Artificial Neural Networks (ANN).

Finally, need to report the accuracy metrics of Precision, Recall, F1 score results from each of three models performed on user dependent and independent data.

Solution

Before we get into the solution, here is the list of resources used to perform this project:

Programming language	Python
IDE	Jupyter Notebook
Python Libraries	os, glob, numpy, pandas, matplotlib, sklearn, keras, Tensorflow
Models/Methods	Sklearn's Train test split, Random sampler, Standard scaler, PCA, Decision tree, SVM, Accuracy score, Confusion matrix, Classification report. Keras's Neural network methods: Sequential, Dense, Adam optimizer.

Extraction Methods	Min, Max, Mean, Standard Deviation, Root Mean Square (RMS), Fast Fourier Transform (FFT)

Here is the solution for each of the phases:

Phase 1: Data Cleaning and Organization

Using the sample frame numbers from the actual activity information (video data), picked up the eating activity as well as non-eating rows separately from activity sensor data for each of the users. As a result, two feature arrays one for eating and another for non-eating are captured for all 30 users. Then combined all the users eating arrays and also, non-eating arrays to establish combined eating and non-eating arrays. And each of these arrays got eight features namely emg1, emg2, emg3, emg4, emg5, emg6, emg7, emg8 and with dimension of N x 8, where N is number of data points.

Phase 2: Feature Extraction

In this phase, new features are extracted by applying the five extraction methods on the original eight features. The five extraction methods used are Maximum, Mean, Standard Deviation, Root Mean Square, Fast Fourier Transform (FFT). The intuition for using such extracted features is to have interesting new features that can help differentiate eating vs non-eating activity. Here is the algorithm to extract features:

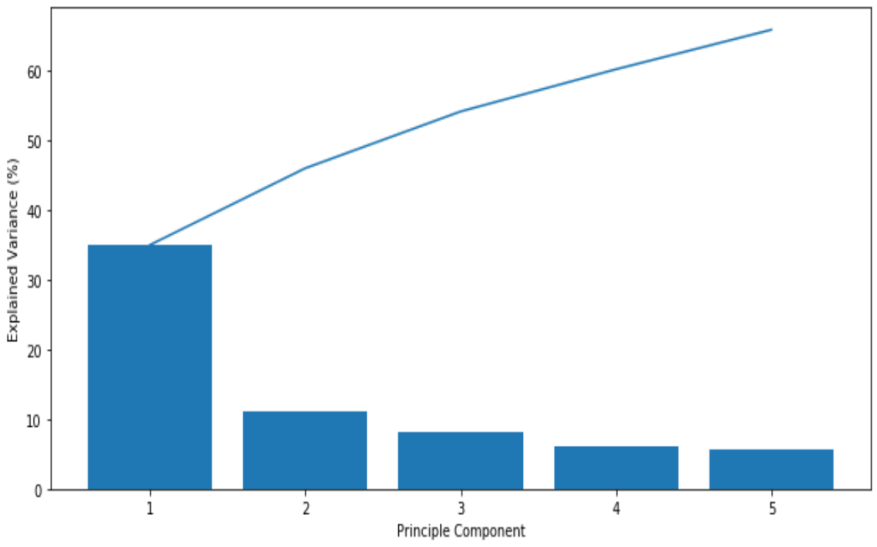
For each extraction method (Max, Mean, STD, RMS, FFT)

a. For Eating activity
   i. Group by n rows (n= 200) at a time
   ii. Apply the method
   iii. Concatenate the new features on every iteration
b. For Non-eating activity
   i. Group by n rows (n= 200) at a time
   ii. Apply the method
   iii. Concatenate the new features on every iteration

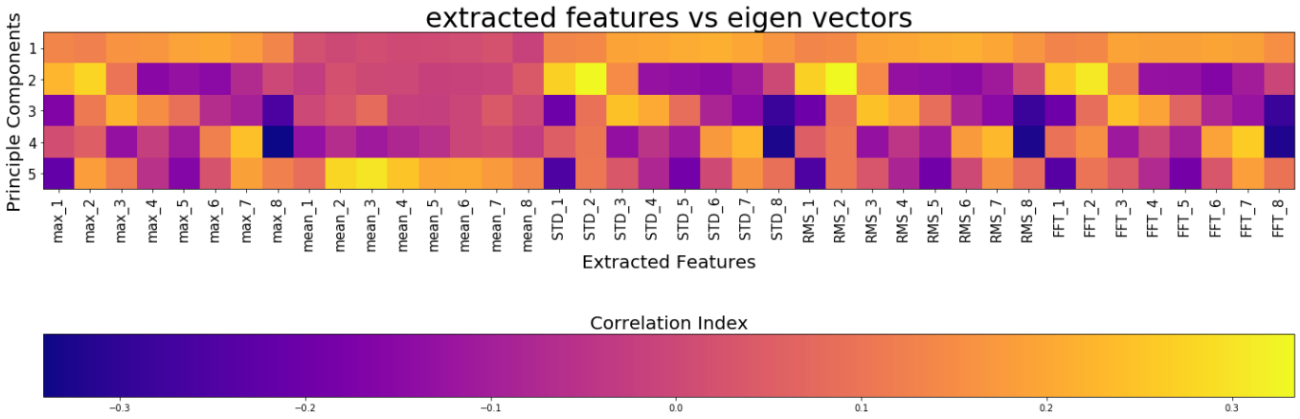
This algorithm returns two arrays of extracted features, one for eating and another for non-eating with dimension of N X 40 (where N is number of data points) each.

Phase 3: Feature Selection

In this section, reduced the feature space from 40 features and kept only top 5 features that has shown maximum variance between the two classes (eating vs non-eating). For reducing the dimension of feature space, used Principal Component Analysis (PCA). To perform PCA, two feature extracted arrays (from above) are combined and applied feature scaling method from skearn library to normalize the data. Then, on this scaled data, applied PCA technique with five components as parameter. Here is the plot that provides details about the principle component and their explained variance.



Application of PCA technique not only helped in reducing the feature array dimension from N x 40 to N x 5 (where N is number of data points), but also captured 66% of total variance with just 12.5% of features. Here is the plot to show how the 40 features are compressed to form 5 components using PCA



Phase 4: User dependent Analysis

In this phase, for each user, PCA applied data will be split into train (60%) and test (40%) data, and then combined all the train data and test data respectively before applying three machine learning models. The models used in this phase are Decision Tree, Non-linear SVM, Artificial Neural Network.

Decision Tree: For computing Decision Tree model, used Sklearn's "DecisionTree Classifier" method and used sklearn's grid search method to obtain optimum hyperparameters for the data. These optimum parameters (Criterion: 'gini', max_depth = 8) are then used to build the model.

Support Vector Machine: For computing non-linear (rbf or gaussian kernel) SVM model, used sklearn's "SVC" method and used sklearn's grid search method to obtain optimum hyperparameters for the data. These optimum parameters (C: 100, gamma = 0.01, kernel = 'rbf') are then used to build the model.

Neural Network: For computing artificial neural network, used keras's "Sequential" and "Dense" methods. For this model, used 2 hidden layers with 'relu' activation function and since output is binary, used 'sigmoid' activation function to predict the results. Other hyperparameters used are Adam optimizer with learning rate 0.01, loss function = "binary_crossentropy", batch size = 100, and number of epochs = 20.

Here are the results of all three models:

Metrics	Decision Tree	Non-Linear SVM	Neural Network
Confusion Matrix	`[[535 115]` `[210 440]]`	`[[544 106]` `[179 471]]`	`[[544 106]` `[172 478]]`
Training set accuracy	87%	84%	83%
Testing set accuracy	75%	78%	79%
Precision	76% (weighted avg)	78% (weighted avg)	79% (weighted avg)
Recall	75% (weighted avg)	78% (weighted avg)	79% (weighted avg)
F1-score	75% (weighted avg)	78% (weighted avg)	79% (weighted avg)

Phase 5: User independent Analysis

In this phase, combined user PCA data is split into train (60%) and test (40%) data before applying three machine learning models. The models used in this phase are Decision Tree, Non-linear SVM, ANN.

Decision Tree: For computing Decision Tree model, used Sklearn's "DecisionTree Classifier" method and used sklearn's grid search method to obtain optimum hyperparameters for the data. These optimum parameters (Criterion: 'gini', max_depth = 10) are then used to build the model.

Support Vector Machine: For computing non-linear (rbf or gaussian kernel) SVM model, used sklearn's "SVC" method and used sklearn's grid search method to obtain optimum hyperparameters for the data. These optimum parameters (C: 100, gamma = 0.01, kernel = 'rbf') are then used to build the model.

Neural Network: For computing artificial neural network, used keras's "Sequential" and "Dense" methods. For this model, used 2 hidden layers (hidden layer 1 with 20 nodes and hidden layer 2 with 10 nodes) with 'relu' activation function and since output is binary, used 'sigmoid' activation function to predict the results. Other hyperparameters used are Adam optimizer with learning rate 0.01, loss function = "binary_crossentropy", batch size = 100, and number of epochs = 20.

Here are the results of all three models:

Metrics	Decision Tree	Non-Linear SVM	Neural Network
Confusion Matrix	[[582 49] [85 546]]	[[603 28] [57 574]]	[[593 38] [48 583]]
Training set accuracy	96%	93%	94%
Testing set accuracy	89%	93%	93%
Precision	90% (weighted avg)	93% (weighted avg)	93% (weighted avg)
Recall	89% (weighted avg)	93% (weighted avg)	93% (weighted avg)
F1-score	89% (weighted avg)	93% (weighted avg)	93% (weighted avg)

Results

Based on the metrics from user dependent and user independent analysis, we can clearly see that all the three models (Decision Tree, non-linear SVM, ANN) performed much better in predicting user activity when used with user independent data. Also, when we compare metrics among the models, non-linear SVM and artificial neural network performed almost similar and both of them performed better than Decision tree model. In conclusion, one can safely select either artificial neural network or non-linear SVM to predict the user activity and these models will do the best when used on user independent data for this activity recognition.

Contributions

This is an individual project. All the tasks, right from understanding the project requirements to reporting the results are performed individually by me. But here, I would like to give some credit to Prof. Salman and his teaching assistants for providing guidance and support throughout the project life-cycle.

Lessons Learned

Here are some of the lessons learnt from this project:

- Getting to work with real world raw sensor data.
- Getting to know how to handle data inconsistencies when working with raw data.
- Getting to know how the extractions methods can help to distinguish the class data.
- Finally, this project helped to strengthen my understanding on PCA dimensionality reduction technique and also on Decision Tree, SVM and Artificial Neural Network machine learning models.