Assignment 2 & 3 Report

CSE 572 Data Mining

Fall 2018

Submitted to:

Professor Ayan Banerjee

Ira A. Fulton Schools of Engineering

Arizona State University

Submitted by:

Ashm Walia

Harshit Laddha

Manav Bagai

Raj Buddhadev

Shubham Mittal

(Group - 2)

Table of Contents

1. Introduction

The aim of this project is to develop a system that can understand human activities such as eating using fork or spoon. For Assignment 2, the projected data is divided into two parts - 60 % for each user as training data and 40% for each user as testing data. For the 3rd assignment, the data from 60% users is taken as testing and rest data 40% is taken as testing data. After that Decision Trees, Support Vector Machines and Neural Networks are used to predict the particular action per user. We applied Grid Search over hyperparameters with 4 fold Cross validation for hyperparameter selection.Each machine is trained with the training data and then test data is used to report accuracy. Accuracy is reported for each of the 30 users in terms of Precision, Recall and F1 score.

2. Team Member

- Ashm Walia - awalia6@asu.edu
- Harshit Laddha - hladdha@asu.edu
- Manav Bagai - mbagai@asu.edu
- Raj Buddhadev - rbuddhad@asu.edu
- Shubham Mittal - smitta21@asu.edu

3. Feature Extraction and Classification

In this phase, we evaluate the performance of different models (using Precision, Recall and F1 Score) when classifying actions of different users. Data from different sensors is collected for 32 users and is then preprocessed and divided into 2 different activities.

Feature matrix with 127 features is then generated from this data by applying various signal processing techniques such as FFT, RMS, STD and AVG. Dimensions of this feature matrix is reduced to 20 by applying PCA on the standardized feature matrix. The new acquired features which are linear combination of the original features are used as an input to the classification model.

Projected matrix obtained after the PCA is divided into training set and testing test. Decision Trees, Support Vector Machines and Neural Networks are then

used to train the model. The test data is then tested on this model and performance of the trained model is obtained using accuracy metrics such as precision, recall, f1-score.

3.1 Techniques

Classification

Classification is the problem of identifying to which of a set of categories does a new observation belongs, on the basis of a training set of data containing observations whose category is known.

Decision Trees

A decision tree is a predictive model that uses a tree-like model of decisions and determine their possible consequences. All interior nodes correspond to one of the input variables while the edges represent each possible of these inputs. Leaves of the tree represent the possible outcome, which can be reached with a particular set of input.

Support Vector Machines

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

Neural Networks

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input.

3.2 Performance Metrics

We'll use the following four parameters to define the following:

- True Positive (TP) - Positive class correctly classified by the classifier
- True Negative (TN) - Negative class correctly classified by the classifier
- False Positive (FP) - Negative class wrongly classified by the classifier
- False Negative (FN) - Positive class correctly classified by the classifier

Accuracy

Accuracy = (TP+TN)/(TP+TN+FP+FN)

Precision

Precision = (TP)/(TP+FP)

Recall

Recall = (TP)/(TP+FN)

F1 Score

F1 score = 2(precision*recall)/(precision + recall)

4. Implementation

Decision Tree

Input:

Data is generated using IMU and EMG sensors for eating using spoon and fork. For each duration present in ground truth directory following features are extracted- FFT, Mean, Median, Maximum, Minimum, and Standard Deviation for each column. A feature matrix is generated for both spoon and fork and PCA is applied over it . Feature matrix is then multiplied with the core matrix that we got as an output from PCA to obtain projection of the dataset onto the eigenvectors obtained from PCA. Only top 20 eigenvectors are considered as they explain the majority of variance in the data. The feature matrix is divided in the ratio of 60:40 for training and testing respectively. The decision tree is trained on the training set and tested on the testing set. Accuracy metrics are calculated on the results after testing and reported for each user and each gesture.

Support Vector Machine

Input:

Features are extracted from the raw data of the sensors for the 2 different activities of each user (group). PCA is then applied on the feature matrices of different activities and different users. Feature matrix is then multiplied with the output of the PCA to obtain projection of the dataset onto the eigenvectors obtained from PCA. Out of all the eigenvectors, only top 20 eigenvectors are considered as they explain the majority of variance in the data. The feature matrix obtained after PCA for different activities are combined resulting in 1 matrix per user.

Multilayer Fully Connected Neural Network

Input:

Features are extracted from the raw data of the sensors for the 2 different activities of each user (group). PCA is then applied on the feature matrices of different activities and different users. Feature matrix is then multiplied with the

output of the PCA to obtain projection of the dataset onto the eigenvectors obtained from PCA. Out of all the eigenvectors, only top 20 eigenvectors are considered as they explain the majority of variance in the data. The feature matrix obtained after PCA for different activities are combined resulting in 1 matrix per user. Since its a multinomial classification using binary classifier, 1 vs all classification method has been used. So, the classification technique is applied 2 times as there are 2 classes, each time labelling actions corresponding to a gesture as 1 and remaining as 0. The feature matrix is divided in the ratio of 60:40 for training and testing respectively. The feed forward network is trained on the training set and tested on the testing set. Accuracy metrics are calculated on the results after testing and reported for each user and each gesture.

5. Results - Assignment 2 User Dependent Analysis

Decision Tree:

4 fold Cross Validation Grid search over the following hyperparameters:

criterion: gini , entropy
max_depth: 4, 5, 7, 9, 11

Best Parameters Selected using average F1 Score:

criterion: entropy
max_depth: 7

Overall Results:

precision_score: 0.6205250596658711
recall_score: 0.6032482598607889
f1_score: 0.611764705882353

Neural Network:

4 fold Cross Validation Grid search over the following hyperparameters:

Hidden_layer_sizes: (12,5),(10,5),(8,5)
activation: ['relu', 'logistic']
solver: ['adam']
learning_rate_init: [0.001, 0.003, 0.006, 0.01]

max_iter: [1000, 1500]

Best Parameters Selected using average F1 Score:

activation: logistic
hidden_layer_sizes: (12, 5) Two hidden layers with Size 12 and 5.
Learning_rate_init: 0.003
Max_iter: 1000
solver: adam

Overall Results:

precision_score: 0.5992292870905588
recall_score: 0.7215777262180975
f1_score: 0.6547368421052632

Support Vector Machine:

4 fold Cross Validation Grid search over the following hyperparameters:

C: [1.0, 0.8]
kernel: ['linear', 'rbf']
gamma: ['auto', 'scale'] // only used for rbf

Best Parameters Selected using average F1 Score:

C: 0.8
Kernel: linear

Overall Results:

precision_score: 0.593952483801296
recall_score: 0.6380510440835266
f1_score: 0.6152125279642058

Results for each user for all the three classifiers:

User	Classifier	Precision	Recall	F1
user09	decision_tree	0.7142857143	0.5	0.5882352941
user10	decision_tree	0.5416666667	1	0.7027027027
user11	decision_tree	0.3636363636	0.2666666667	0.3076923077

user12	decision_tree	0.6	0.4	0.48
user13	decision_tree	0.6923076923	0.9	0.7826086957
user14	decision_tree	0.9166666667	0.7333333333	0.8148148148
user16	decision_tree	0.6875	0.7333333333	0.7096774194
user17	decision_tree	0.3333333333	0.3333333333	0.3333333333
user18	decision_tree	0.5833333333	0.7	0.6363636364
user19	decision_tree	0.6470588235	0.7333333333	0.6875
user21	decision_tree	0.75	1	0.8571428571
user22	decision_tree	0.5	0.8666666667	0.6341463415
user23	decision_tree	0.6363636364	0.4666666667	0.5384615385
user24	decision_tree	0.9333333333	0.9333333333	0.9333333333
user25	decision_tree	1	0.4	0.5714285714
user26	decision_tree	0.7142857143	0.6666666667	0.6896551724
user27	decision_tree	0.9	0.6	0.72
user28	decision_tree	0.5652173913	0.8666666667	0.6842105263
user29	decision_tree	0.6363636364	0.4666666667	0.5384615385
user30	decision_tree	0.6363636364	0.4666666667	0.5384615385
user31	decision_tree	0.7142857143	0.3333333333	0.4545454545
user32	decision_tree	0.3333333333	0.2307692308	0.2727272727
user33	decision_tree	0.5333333333	0.5333333333	0.5333333333
user34	decision_tree	0.5	0.4666666667	0.4827586207
user36	decision_tree	0.4	0.1333333333	0.2
user37	decision_tree	0.5	0.7333333333	0.5945945946
user38	decision_tree	0.6875	0.7333333333	0.7096774194
user39	decision_tree	0.8125	0.8666666667	0.8387096774
user40	decision_tree	0.6875	0.7333333333	0.7096774194
user41	decision_tree	0.4	0.4	0.4
user09	neural-net	0.7272727273	0.8	0.7619047619
user10	neural-net	0.4230769231	0.8461538462	0.5641025641

user11	neural-net	0.4375	0.4666666667	0.4516129032
user12	neural-net	0.6666666667	0.8	0.7272727273
user13	neural-net	0.6153846154	0.8	0.6956521739
user14	neural-net	0.6875	0.7333333333	0.7096774194
user16	neural-net	0.55	0.7333333333	0.6285714286
user17	neural-net	0.5416666667	0.8666666667	0.6666666667
user18	neural-net	0.5	0.8	0.6153846154
user19	neural-net	0.625	1	0.7692307692
user21	neural-net	0.6666666667	0.6666666667	0.6666666667
user22	neural-net	0.5555555556	1	0.7142857143
user23	neural-net	0.5625	0.6	0.5806451613
user24	neural-net	0.7222222222	0.8666666667	0.7878787879
user25	neural-net	1	0.8	0.8888888889
user26	neural-net	0.7333333333	0.7333333333	0.7333333333
user27	neural-net	0.4545454545	0.3333333333	0.3846153846
user28	neural-net	0.6	1	0.75
user29	neural-net	0.5238095238	0.7333333333	0.6111111111
user30	neural-net	0.6875	0.7333333333	0.7096774194
user31	neural-net	0.6923076923	0.6	0.6428571429
user32	neural-net	0.5714285714	0.6153846154	0.5925925926
user33	neural-net	0.75	0.8	0.7741935484
user34	neural-net	0.6153846154	0.5333333333	0.5714285714
user36	neural-net	0.5833333333	0.4666666667	0.5185185185
user37	neural-net	0.3529411765	0.4	0.375
user38	neural-net	0.5652173913	0.8666666667	0.6842105263
user39	neural-net	0.7857142857	0.7333333333	0.7586206897
user40	neural-net	0.65	0.8666666667	0.7428571429
user41	neural-net	0.4705882353	0.5333333333	0.5
user09	svm	0.5454545455	0.6	0.5714285714

user10	svm	0.44	0.8461538462	0.5789473684
user11	svm	0.75	0.4	0.5217391304
user12	svm	0.75	0.8	0.7741935484
user13	svm	0.6923076923	0.9	0.7826086957
user14	svm	0.6666666667	0.6666666667	0.6666666667
user16	svm	0.6315789474	0.8	0.7058823529
user17	svm	0.380952381	0.5333333333	0.4444444444
user18	svm	0.5	0.6	0.5454545455
user19	svm	0.6666666667	0.9333333333	0.7777777778
user21	svm	0.5	1	0.6666666667
user22	svm	0.5357142857	1	0.6976744186
user23	svm	0.6428571429	0.6	0.6206896552
user24	svm	0.8235294118	0.9333333333	0.875
user25	svm	1	0.6666666667	0.8
user26	svm	0.75	0.4	0.5217391304
user27	svm	0.4	0.1333333333	0.2
user28	svm	0.5	0.6666666667	0.5714285714
user29	svm	0.4	0.2666666667	0.32
user30	svm	0.75	0.8	0.7741935484
user31	svm	0.3333333333	0.1333333333	0.1904761905
user32	svm	0.4444444444	0.3076923077	0.3636363636
user33	svm	0.6666666667	0.6666666667	0.6666666667
user34	svm	0.7857142857	0.7333333333	0.7586206897
user36	svm	0.5555555556	0.6666666667	0.6060606061
user37	svm	0.3333333333	0.2	0.25
user38	svm	0.5454545455	0.8	0.6486486486
user39	svm	0.875	0.4666666667	0.6086956522
user40	svm	0.5652173913	0.8666666667	0.6842105263
user41	svm	0.6	0.8	0.6857142857

6. Results - Assignment 3 User Independent Analysis

Decision Tree:

4 fold Cross Validation Grid search over the following hyperparameters:

criterion: gini, entropy
max_depth: 4, 5, 7, 9, 11

Best Parameters Selected using average F1 Score:

Criterion: gini
max_depth: 7

Overall Results:

precision_score: 0.462478184991274
recall_score: 0.5662393162393162
f1_score: 0.5091258405379443

Neural Network:

4 fold Cross Validation Grid search over the following hyperparameters:

Hidden_layer_sizes: (12,5),(10,5),(8,5)
activation: ['relu', 'logistic']
solver: ['adam']
learning_rate_init: [0.001, 0.003, 0.006, 0.01]
max_iter: [1000, 1500]

Best Parameters Selected using average F1 Score:

activation: relu
hidden_layer_sizes: (12, 5) // Two hidden layers with size 12 and 5
Learning_rate_init: 0.01
Max_iter: 1000
solver: 'adam'

Overall Results:

precision_score: 0.48516320474777447
recall_score: 0.6987179487179487
f1_score: 0.5726795096322241

Support Vector Machine:

4 fold Cross Validation Grid search over the following hyperparameters:

C: [1.0, 0.8]
kernel: ['linear', 'rbf']
gamma: ['auto', 'scale'] // only for RBF Kernel

Best Parameters Selected using average F1 Score:

C: 1.0
gamma: 'auto'
kernel: rbf

Overall Results:

precision_score: 0.4844827586206897
recall_score: 0.6004273504273504
f1_score: 0.5362595419847327

Results for each user for all the three classifiers:

User	Classifier	Precision	Recall	F1
user10	decision_tree	0.447761194	0.8108108108	0.5769230769
user11	decision_tree	0.375	0.075	0.125
user16	decision_tree	0.2857142857	0.15	0.1967213115
user17	decision_tree	0.5	0.55	0.5238095238
user18	decision_tree	0.40625	0.7428571429	0.5252525253
user19	decision_tree	0.5063291139	1	0.6722689076
user21	decision_tree	0.4868421053	0.925	0.6379310345

user26	decision_tree	0.5333333333	0.6	0.5647058824
user27	decision_tree	0.4782608696	0.275	0.3492063492
user28	decision_tree	0.5555555556	1	0.7142857143
user29	decision_tree	0.4186046512	0.45	0.4337349398
user32	decision_tree	0.2580645161	0.2222222222	0.2388059701
user10	neural-net	0.4084507042	0.7837837838	0.537037037
user11	neural-net	0.5714285714	0.1	0.170212766
user16	neural-net	0.5483870968	0.425	0.4788732394
user17	neural-net	0.4461538462	0.725	0.5523809524
user18	neural-net	0.4666666667	1	0.6363636364
user19	neural-net	0.5063291139	1	0.6722689076
user21	neural-net	0.5	1	0.6666666667
user26	neural-net	0.4888888889	0.55	0.5176470588
user27	neural-net	0.5	0.025	0.04761904762
user28	neural-net	0.5	0.975	0.6610169492
user29	neural-net	0.5064935065	0.975	0.6666666667
user32	neural-net	0.5	0.8888888889	0.64
user10	svm	0.4545454545	0.9459459459	0.6140350877
user11	svm	0.3333333333	0.05	0.08695652174
user16	svm	0.6	0.225	0.3272727273
user17	svm	0.5	0.725	0.5918367347
user18	svm	0.4571428571	0.9142857143	0.6095238095
user19	svm	0.5063291139	1	0.6722689076
user21	svm	0.5063291139	1	0.6722689076
user26	svm	0.6842105263	0.325	0.4406779661
user27	svm	0.75	0.075	0.1363636364
user28	svm	0.5	0.825	0.6226415094
user29	svm	0.5147058824	0.875	0.6481481481
user32	svm	0.2564102564	0.2777777778	0.2666666667

7. Conclusion

We used same models and same hyperparameter search techniques with F1 Score for comparing results of User dependent and User independent analysis.

For both User Dependent Analysis(Assignment 2) and User Independent Analysis(Assignment 3), Neural networks performed best then SVM and then Decision trees.

According to the user dependent analysis, the accuracy metrics is consistent across all the users, activities and classification techniques. There are a few outliers which are having unsatisfactory values in the accuracy metrics for few combinations. However, the overall accuracy metrics for the user dependent analysis is satisfactory. This also suggests that the feature extraction and feature selection done in Assignment 1 is accurate for us to get satisfactory results.

As in **User dependent analysis(Assignment 2)** users for which testing is done are also included in training data, the evaluation metrics are similar during training and testing. Also, overall metrics(across all users) are similar for training as well as testing data which shows that models didn't overfit.

User Independent Analysis(Assignment 3): In this analysis the testing users' data is not present during training there is considerable decrease in evaluation metrics in comparison to user dependent analysis across all the three classifiers(decision tree, svm, neural-net).

Moreover there is significant drop from training evaluation metrics to testing evaluation metrics as testing data users were completely new.

Still since the change evaluation metrics for each user across different classifiers is mostly consistent we can say that these classifiers converged well.