Hand Gesture Recognition using Random Forest Classifier

By Mahrang Saeed & Shifabanu Shaikh
EE Department, San Jose State University

Abstract— **One of the examples of human-machine interaction (HMI) is gesture recognition system with EMG signals. EMG signals can be applied to various machine learning algorithms and can be utilized in various applications. In this paper, we use pre-recorded EMG signals in the Random Forest Classifier to note its accuracy and compare it with High-Dimensional (HD) Classifier. Principle Component Analysis is carried out to see if performance can be improved. We recorded 97.00% accuracy result within few trials for five hand gesture recognition while HD gave 96.64%**.

I. Introduction

Hand gestures are not only important for communication between humans but also nowadays, it is very useful with human-machine interaction (HMI). HMI has varied applications with evolving technology such as poly-articulated prosthetic hands, gaming interfaces, and many more. Thus, an efficient and accurate system which recognizes the gestures is required. Researchers are looking to employ onboard systems to improve the control and sophistication of prosthetic hands by using various machine learning approaches that read and react to nerve signals transmitted through the arm.

The practice of tracking the natural electric impulses sent by the brain to control the individual muscles is known as electromyography or EMG. EMG signals can be recorded by various means of electrodes. One of the means of recording is using a dense array of sensors with full muscle coverage. [1] These signals are already efficiently recorded in one of the research papers we referred. They then pre-processed the EMG signals and converted it into high-dimensional vectors and trained the classifier. They recorded 96.64% accuracy in three trials.

Here, we use Random Forest classifier with the same raw data recorded in HD computing [1] and try to compare the accuracy results. Random Forest, also known as Random Decision Forests, is an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or classification or mean predication or regression of individual trees. [2]

To improve performance, we carried out Principal Component Analysis. [3] It is a feature reduction process and also a feature extraction technique. It helps in combining the input variables in a certain format so that we can ignore the least valuable variables from the total number of variables and use only the most valuable variables in application. All the variables or features are independent to one another. Clusters of correlated features can be converted into principal components. The principal components are in the vector form which includes the features. [4] The principle components are perpendicular to each other. The model is trained using these principle components instead of features directly.

This paper briefs about the preprocessing of raw data taken from HD classifier, accuracy results of Random Forest model, improvement of performance using Principle Component Analysis, and comparison with the HD Computing Model.

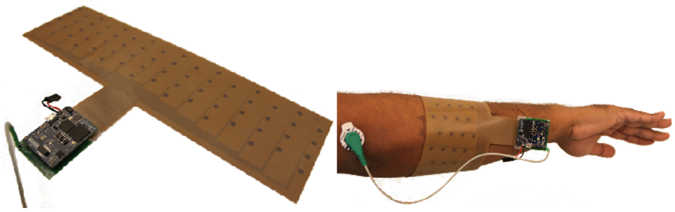

Fig. 1. Flexible electrode array with attached wireless bio-signal acquisition device (left) positioned on the arm (right).

Fig. 2. Dataset gestures (a) with the associated normalized activity maps for different sessions (Subject 1 shown)

II. Raw Data

The gesture signals were taken from HD computing paper. Recorded EMG signals were of three healthy males recorded in three different sessions. Each session had its own training as well as testing signals. We only utilized the signals of subject 1 which were recorded in the first session in our study. Signals were recorded by a wearable device with 64 electrodes. The training data set and testing data set for session 1 were recorded 30 minutes apart. Figure 1 shows the array of electrodes and placement of electrodes when recording the signal. [1]

Figure 2 shows the 5 gestures: rest, fist, raise, lower, open. [1] Every gesture started or ended with the rest gesture. Thus, in total 5 gestures were recorded. These signals were recorded in 10 trials. Each gesture took 5 seconds of recording time.

Each trial lasted for 28 seconds. 1000 samples were collected every second. Thus, the sampling rate was 1 kHz and every trial contained 28,000 samples. The sample size was reduced to 14,995 by utilizing only the 3 middle seconds of each 5 second gesture, excluding the last 3 seconds.

III. Pre – Processing

The raw signals that we acquired is a combination of the required EMG signals, noise and other harmonics. However, the wanted signal is the high frequency signal envelope of EMG potential. To remove the unwanted signals attached to the EMG signals, pre-processing was performed.

Figure 3 shows the pre-processing steps. The data is scaled up by a factor of 10. This is done before application of filters to improve overall accuracy. The notch filter at 60 Hz with Q-factor of 50 removes the power line interference while the DC offset and drift are removed using 8^{th} order Butterworth bandpass filter having frequency range as 1-200Hz.

For extracting the envelope, the absolute value is taken and a moving average filter of size 100 is applied. The pre-processing ends with normalization of the extracted data.

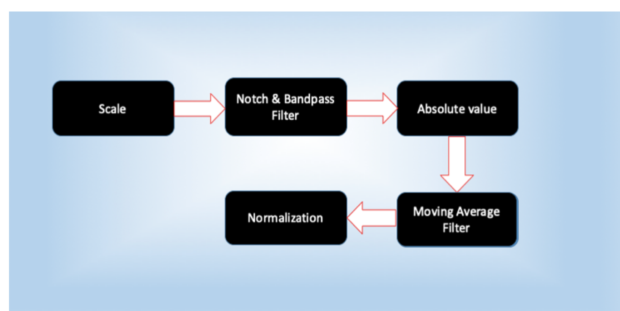

Fig. 3. Pre-processing of the raw signals.

IV. Random Forest Classifier

Ten batches were created using the dataset from subject 1 session 1. All ten batches used all 10 test trials as the test set. Batch 1 used only trial 1 from the training set as the train set. Each successive batch used 1 additional trial from training set as train set. Batch 10 used all 10 training trials.

For each batch, the Random Forest model with the best performance was obtained. First, GridsearchCV with 5-fold cross validation was used to find the best hyper-parameters to use in a Random Forest model. The hyper-parameters are bootstrap, criterion, and number of trees. Then, Random Forest models with different number of trees with the best hyper-parameters were trained to find the number of trees that produced the best performance. Number of trees tried were from 5 – 100 in increments of 5. The result was a Random Forest model with the best hyper-parameters and the best number of trees that gave the best performance.

Using the best Random Forest model, we then apply Principal Component Analysis (PCA) to find the best number of principal components to see if using principal components will yield us a better test accuracy. PCA performs feature reduction. In our case, we have 64 features, but some of them may predict the outcome better than others. PCA will determine which features are more important than others. PCA clusters the correlated features into principal components, denoted by Z. Each principal component is a vector comprised of features. Each principal component is perpendicular to the other principal components. When using PCA, the model is trained on the principal components, which are comprised of the features.

We tried different number of principal components ranging from 4 – 64 in increments of 4 to find the best number of principal components to use in our model. If using PCA yielded us better performance, we used the best Random Forest model with the best number of principal components. If PCA did not improve performance, we did not use principal components in our model. The result was the best Random Forest model for each batch.

IV. Classification Results

We built models for each of the 10 batches. Each batch contained x number of training data set and a fixed number of testing data set. The training data set changed from 1 to 10 trials while the number of testing trials was fixed to 10. We noted the results of the models for each batch. Maximum performance of 97.0% is achieved using batch 8, which contained 8 training trials. The best Random Forest model for this batch contained 65 trees and did not use principle components. When only 1 trial was used, applying PCA improved performance by 4.77% using 8 principle components. When 9 trials were used, PCA only improved performance by 0.17%. Thus, applying PCA only improved the performance of batches 1 and 9.

In HD classification, a maximum performance of 96.64% was achieved when only using 3 trials with 5 samples per trial, whereas Random Forest used 8 trials with 14,995 samples per trial to achieve maximum performance of 97.0%. Thus, Random Forest classifier only slightly improves accuracy over HD classifier but it uses significantly larger training set size. Also, Random Forest classifier only improves performance by 0.55% when using 8 trials vs 7 trials.

Table 1 shows the results we achieved for all 10 batches. As shown in the table, the highest accuracy is achieved when 8 training trials are used. After plotting the graph of different number of trials used in each batch versus test accuracy, we note that we got maximum performance using 8 trials. Here we did not use principle components because PCA only degraded the performance of the model. After performing PCA on 8 trials, it yielded 94.86% accuracy, which was lower than before PCA.

A plot of the number of trees tried for 8 training trials versus test error is shown in Figure 4. We can note that minimum test error is achieved when 65 trees are used.

Figure 5 shows the highest test accuracy achieved when number of training trials used are varied from 1-10.

Figure 6 shows the highest accuracy achieved for each number of trials used before and after PCA. From this plot, we can see whether principle components should be used or not in our Random Forest model.

No. of training trials used	1	2	3	4	5	6	7	8	9	10
No. of trees	90	80	30	60	30	30	20	65	40	85
test accuracy before PCA	85.80	92.46	95.47	96.30	96.67	96.73	96.45	97.00	96.14	96.48
No. of principle components	8	40	8	28	32	32	32	64	40	40
test accuracy after PCA	90.57	90.78	91.47	94.36	95.47	93.31	95.51	94.86	96.31	94.18
highest test accuracy	90.57	92.46	95.47	96.30	96.67	96.73	96.45	97.00	96.31	96.48

Table 1. Accuracy Results

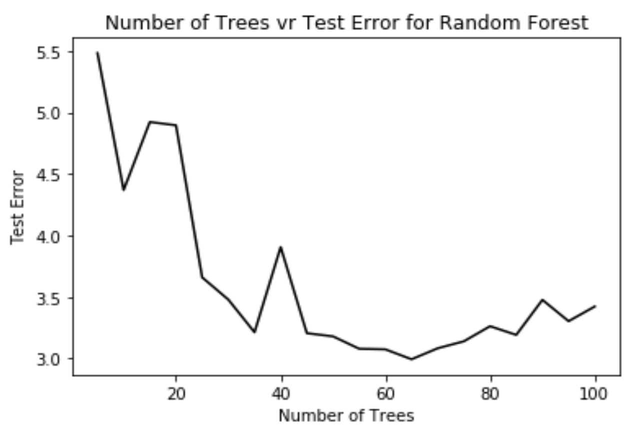

Fig. 4. Number of Trees versus Test Error plot for 8 trials

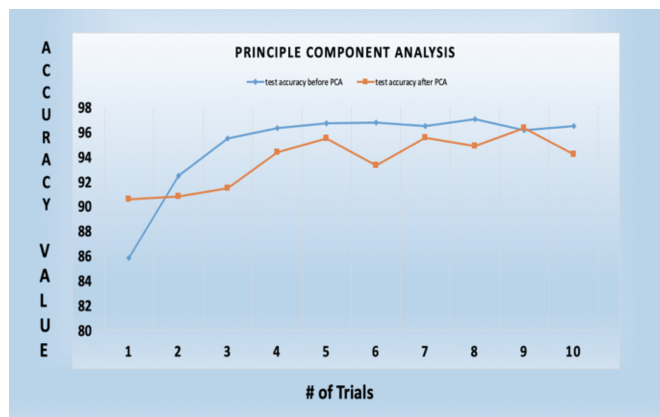

Fig. 6. Principle Component Analysis Results

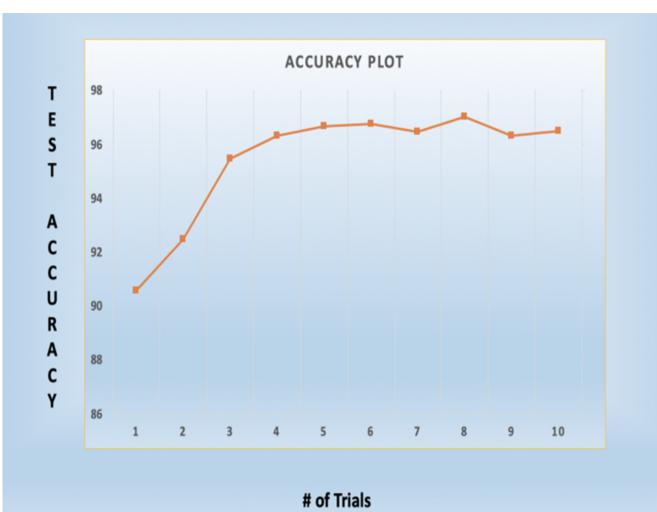

Fig. 5. Graphical representation of Accuracy results

V. Conclusion

Accuracy and efficiency is the main concern for EMG based applications such as hand gesture recognition system. A robust entity is required to successfully apply these signals to various human-machine interface (HMI) implementation. We presented hand gesture recognition system using Random Forest classifier, then we compared the results with that of an HD classifier. Random forest classifier improves accuracy by 0.36% over HD classifier, but it requires larger training data.

Acknowledgment

The authors would like to thank Professor Abbas Rahimi for his support and guidance throughout the project.

References

1. An EMG Gesture Recognition System with Flexible High-Density Sensors and Brain-Inspired High-Dimensional Classifier by Ali Moin, Andy Zhou, Abbas Rahimi, Simone Benatti, Alisha Menon, Senam Tamakloe, Jonathan Ting, Natasha Yamamoto, Yasser Khan, Fred Burghardt, Luca Benini, Ana C. Arias, Jan M Rabaey.
2. Random Forest wikepedia.com.
3. Principle Component Analysis wikepedia.com
4. Forests for randomized trees - A user - guide on scikit.
5. A One – Stop Shop for Principal Component Analysis by Matt Brems April 17, 2017.
6. Using onboard AI to power quicker, more complex prosthetic hands by Conor Hale October 3, 2019.
7. Artificial Intelligence Improves Control of Prosthetic Hands, UT Dallas Researchers, led by Mohsen Jafarzadeh, demonstrate how AI improves control of prosthetic hands at IEEE International Symposium. Newswire.com