Gesture Recognition via Electromyograph Signals

1. Introduction

Skeletal muscles produce electrical activity known as *myoelectric impulses* upon contraction. These myoelectric impulses can be detected and recorded using an electromyograph (EMG). EMGs are currently used to study a number of neuromuscular conditions.

Two electromyograph categories exist currently. The first type of EMG, introduced in 1950, is the *intramuscular EMG*. The intramuscular EMG uses subcutaneous sensors implanted in the skeletal muscle through thin wires. The second EMG type is the *Surface* electromyograph ("sEMG"). The sEMG uses sets of two or more electrodes placed on the surface of the skin. The sEMG represents a compromise between the invasive nature and greater precision associated with its intramuscular counterpart.

Superficial qualities of the subject being measured can have a detrimental affect on the quality of sensor readings. Some examples of these qualities are perspiration, body-fat content, or shifting of sensor pads [1].

A more recent technology trend is that of wearable technology or *wearables* [2]. Modern wearables typically perform a variety of tasks including providing information such as text message alerts and capturing biological data such as the user's heart rate. Wearables have been depicted in SciFi novels and movies for decades but have only recently become common to consumers. As discussed in section 3, quality wearable devices combined with reliable and cost-effective myoelectric sensing technology may lead to a new generation of intuitive human-computer interface ("HCI") devices.

2. Problem Statement

As sEMG devices decline in size and cost, they have become a focus for researchers interested in the information they might provide.The problem of classifying motion as captured with sEMG sensors is not a new one. Researchers have published articles as early as 1967. Indeed, the number of publications on the topic has grown considerably in the last 20 years [3]. In particular, the problem of classifying movements is now popular enough that several datasets of recorded signals and subject characteristics have been generated and published [4].

As stated above, there have been many successful attempts to classify gestures using EMG sensor data. Common methods employed to this task are linear discriminant analysis, artificial neural networks, support-vector machines and more [3]. Despite this success, consumers have been slow to adopt HCI devices for a number of reasons. Two major factors represent hurdles to continued advances in this area:

1) Limited computational resources - Available processing power and on-board memory are restricted to what can be embedded in a necessarily low-profile device. The artificial neural network ("ANN") is a prime example of a high-performing algorithm that has performed will in-sample but does not generalize well due to these constraints.

2) Availability of training data - The amount of training data required to achieve satisfactory results without overfitting is limited to the quality and configuration of equipment of a single study [5]. While the NinaPro database offers a number of relevant data to be used in training gestures classifiers, the data is generated across differing experiments where test subjects make differing gestures while wearing sensor arrays in differing configurations. As a result, the amount of available training data is not linear in the number of new studies being performed.

This project seeks determine whether it is possible to identify latent features common to various classes of gestures while removing variance associated with superficial qualities of the test subject and minor differences in quality and configuration of the sEMG sensor array. The figure 1 depicts a proposed data pipeline for this purpose.

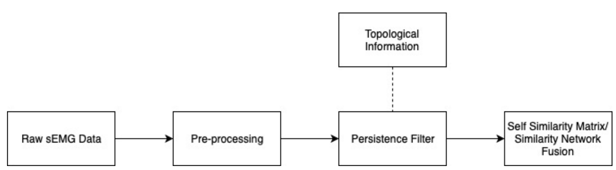

Figure 1: Proposed data pipeline

This pipeline consists of two primary components:

1) A filtering component capable of remove superficial noise inessential to classifying gestures.

2) A classification component capable of identifying a gesture. Ideally this classification will be performed as early as possible in the gesture cycle.

Hypotheses, testing, and construction of these components are described in greater detail in section 4. If the suggested pipeline is feasible, training data captured across studies may be utilized in classification with minimal information loss.

3. Previous Work

The primary work that inspired this project is *Latent Factors Limiting the Performance of sEMG-Interfaces* by Lobov, *et al.* [1]. The research team in this study set out to train both a linear discriminant analysis ("LDA") classifier and an artificial neural network ("ANN") classifier capable of identifying gestures. The classifier would then be used to assist users in playing a simple game via the *Myohalmic armband*, a wearable HCI device equipped an sEMG sensor array. Subjects were able to control the game with some success using gestures. However, performance was far better when using a mouse or joystick. The sEMG training data was smoothed using a root mean square (RMS) moving average.

More recently, Phinyomark *et al.* found applications of TDA for EMG data similar to those generated in [1]. They sampled overlapping windows of EMG signals and then reduced the dimension of these windows via principal component analysis. The *mapper* algorithm was applied to the resulting vectors to create a topological network representing the feature space of EMG signals. Signal amplitude and power, nonlinear complexity and frequency information, and time-series structure were determined to be the most influential features for identifying gesture patterns [6].

The preferred classifier to be utilized in the pipeline in figure 1 utilizes self-similarity matrices ("SSMs") and is based on the similarity network fusion ("SNF") method described in [8]. As a brief summary, SSMs are symmetric matrices whose entries represent a measure of similarity between two points in a time-ordered point cloud ("TOPC"). This measure of similarity may be a distance metric such as the L_2 distance or a similarity measure like the gaussian kernel. A set of SSMs generated from differing signal sources can be normalized and thought of as a transition matrix which can be utilized to randomly walk over a connected graph representing the TOPC. If sufficient iterations are performed, a stationary distribution will be reached and the resulting matrix will also be a similarity matrix which captures information from all of the individual matrices. This approach to clustering seems appropriate to the problem at hand because it was developed to handle sensor data generated from differing sources much like the 8 sensors in my data set (section 3). Additionally, this technique does not need the large, homogeneous training data sets required to adequately fit ANNs or other models with large parameter sets. In the paper, the authors use the SNF method on 2 and 3 modalities. My project attempts the method with 5. As we will see, this is a resource intensive endeavor.

4. **Data**

The *EMG data for gestures Data Set* from the UCI Machine Learning Repository [10] was utilized for this project. This data set was collected by Lobov, *et al.* to conduct research for [1].

This data set consists of 36 subjects performing a series of six distinct gestures. Four of these gestures are simple motions including moving the hand up, down, left, or right (figure 2). The final two motions are complex motions created by a combination of two simple motions (e.g. moving the hadn up and to the left simultaneously).

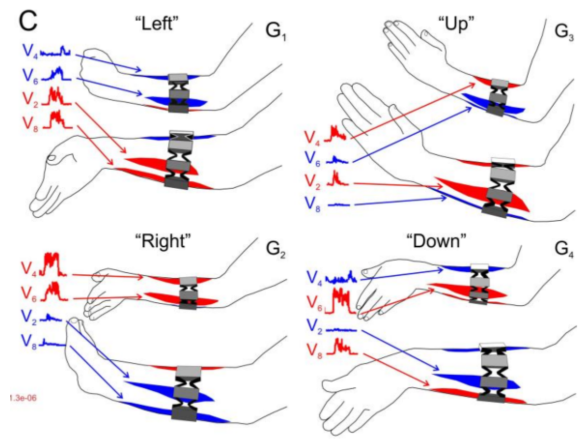

Figure 2: Simple gestures captured by Lobov *et al.* [1]

Subjects performed all gestures four times for approximately 1.5 to 3 seconds for each. The motions were captured by a MyoThalmic bracelet placed as depicted in figure 2. Of the eight evenly spaced sensors, five were situated directly on five main muscles - *flexor carpi radialis*, *flexor carpi ulanis extensor carpi radialis longus*, *extensor digitorum*, *extensor carpi ulanis*, *flexor*, and *palmaris longus*.

To make computation more tractible only the five primary channels were considered in constructing the proposed pipleline (figure 1). Similarly the two complex gestures were removed from the dataset leaving only the four simple gestures. The result is a data set of approximately 720 matrices of size $t \times 6$. The 7 columns of each matrix are time (in milliseconds), sensor readings 2 4, 5, 6, and 8, and a gesture label column.

A notable data issue discovered during exploratory analysis is a large disparity in the amount of time that each subject performed their gesutres. Some subjects performed gestures for close to the full $3s$ asked of them. However, it was much more common for a subject to perform a gesture for approx. $2s$ while some subjects performed a gesture for as little as $1.7s$. Another issue with the dataset is imprecision in the sEMG sensors used. Signals registered in by the MyoThalmic bracelet are truncated towards their peaks and troughs. However, the dataset continues to include readings during the peak and trough periods using the last recorded value. Curves with very "blocky" appearaces are the result (figure 3).

5. **Analysis Methods & Pipeline**

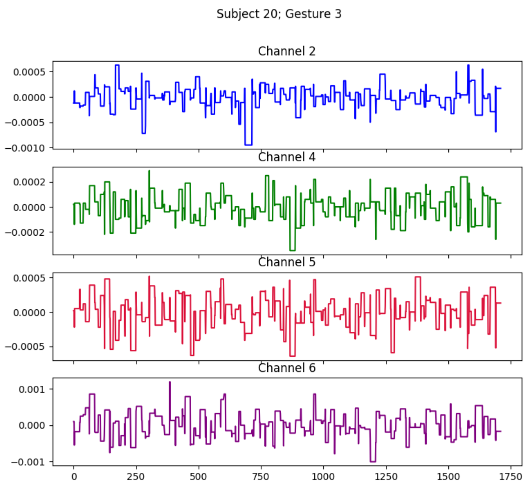

Figure 3: Subject 20 gesture 3. Duration (ms) is on the horizontal axis.

1) *Constructing the Filtering Mechanism*

The first step in constructing a filtering mechanism was determining features that were informative with regards to the gesture being performed. A 20×20 persistence image was generated with a spread parameter of $\sigma = 1 \times 10^{-5}$ over the surface formed by the 5 sEMG channels. This was performed for all 720 simple gestures. the 20×20 pixel parameter was chosen for modeling purposes described below. The spread parameter was chosen to enusre some identifiability remained between gestures. Some samples of these persistence images are deicted below (figure 4). Only the $H1$ homology class is considered for computability purposes.

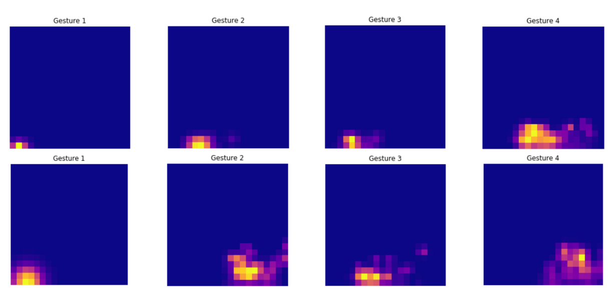

Figure 4: Two samples of persistences images for each gesture

As a first step in determining which features are important and which can be removed from the data, the persistence image vectors were normalized, added within their gesture classes, and the resulting vectors were normalized again. This provides a sort of empirical distribtion within ech gesture class. The resulting images are depicted in figure 5.

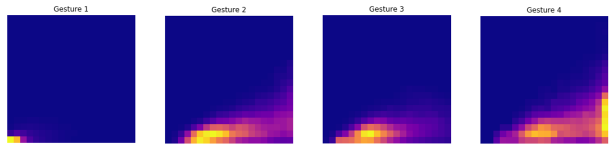

Figure 5: Normalized combinations of persistence images for each gesture

Interestingly, when used to classify the gestures by their persistence images, 57.1% in-sample accuracy was achieved. This is surprisingly high for a heuristic approach to classification as they expected accuracy for a random guess is only 25%. However, this classifier is very fragile. Because some subjects contribute more to the sum total, when single subject is left out of the fitting process this heurisitc classifier may have 0% for all of that individual subject's gestures. Alternatively, some subjects were fit with perfect accuracy when left out of the training sample. This provides evidence that commonalities do exist across subjects with regards to gesture performance.

The approach just described is interesting, but determining important features requires a more rigorous approach. This rigor is achieved by applying an L_1 normalized multi-class logistic regression to the persistence images. The L_1 penalty is included in order to achieve a more sparse set of parameters with the hopes of creating a more aggressive filter. The inverse persistence images formed by the regression coefficients (one for each of the 400 pixels) are arranged and presented below.

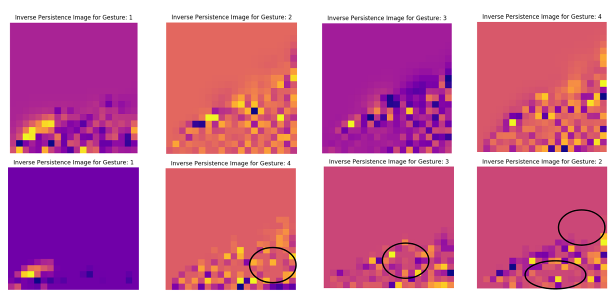

Figure 6: Inverse persistence images of non-sparse (top row) & sparse (bottom row) regression coefficients

From figure 6 we can see that the sparsity induced by the L_1 penalty is mostly induced in the middle and high persistence 1-cycles. This discovery represents a problem for the construction of a persistence filter because it shows that the information provided by the low persistence 1 cycles is important to retain. Because of the elder rule, we cannot remove more persistent cycles without also removing those cycles with also removing less persistent cycles.

It is possible that by inlcuding the full gesture cycle, as opposed to just the beginning or just the end of the cycle, we have included information about *two* gestures - the motion to make the gesture and the motion caused by releasing the gesture and returning to a neutral state. This may have a sort of "mirror image" effect making gestures more challenging to recognize. Further research will be done to test this hypothesis as well as to determine commonalities among gestures which are misclassified by the regression models.

2) *Constructing the Classifier*

Methods from topological data analysis will be utilized for feature extraction. Persistence diagrams allow us to inform decisions about high-dimensional geometric properties of data without the information loss associated with other techniques for visualizing features (e.g. principle component analysis). Persistence images use a weighting function to emphasize pertinent topological features of a signal and de-emphasize noise. This approach should outperform the typical moving average approach [3] to pre-processing sEMG signals which "bake-in" signal noise and in some cases smooth over potentially important characteristics like medium-sized amplitudes.

Important topological characteristics identified in this way will be used to encode a filter capable of differentiating useful signal from statistical noise. This represents the first stage in the pipeline. The filter will work by discarding the irrelevant components of the sEMG modalities as identified in the previous paragraph and keeping those components that provide useful information. For example, this might mean removing all periods whose absolute amplitude is less than some threshold which corresponds to imprecision in the sEMG sensor bracelet.

The next stage of the pipeline will feature a classifier capable of differentiating gestures based on their vectorized persistence diagrams or images. A supervised method will first be utilized here to test the separability of the gesture classes. Following tests of supervised learners and separability of the vectors, unsupervised methods of clustering will be employed. The intention of using unsupervised methods is to further provide evidence that large training samples and complex machine learning algorithms are not strictly necessary for gesture recognition. Currently, the preferred classifier for this stage in my pipeline is the SSM described in section 2 paragraph 3.

6. **Software**

My exploratory work with the data set thus far has utilized Python's standard packages for data analysis including matplotlib and NumPy. Additionally, I have been working with libraries that comprise the Scikit-TDA such as ripser, persim, and cechmate. I have already created a series of specialized functions for performing the analysis and building the pipeline described in the previous section. Looking forward the functions in Scikit-TDA should be sufficient for my needs throughout this project.

The supervised and unsupervised classifiers in the second stage of my pipeline will be implemented via the scikit-learn library. Additional libraries will be employed as needed for these purposes. The SSM approach [7][8] to unsupervised clustering will likely require that I develop several functions to meet my specific needs. This should be feasible using NumPy and SciPy, but may require some optimization for run-time efficiency. This can likely be performed using a number of Python-based C/ C++ optimization tools. Time permitting, this may also provide an opportunity to develop and employ functions in Julia to take advantage of the high-performance numerical analysis features found there.

7. **Risk Mitigation**

Current exploratory analysis is promising. However, there are three primary areas which could slow progress:

1. *High computational complexity* - As exploratory analysis continue there may come a point where computing various complexes and persistence diagrams becomes computationally infeasible. This scenario is low risk as a significant amount of computation has already been performed without issue. The libraries described in the previous section are well optimized and the data set is not overly complex. Nevertheless, should computation become an issue the code and data can quickly be transferred to Duke's distributed servers and docker containers. Thus development will continue while other components are tested and tasks are performed.
2. *Poor classifier performance* - This is also low risk. There are many alternative methods for classifying gestures. The real risk in this area is the risk of over-fitting a particular classifier to the data set resulting in poor out-of-sample performance. The goal of the project is to find a classifier that is invariant to the user who is making the gesture. Over-fitting the data would result in a failure to achieve this goal. To prevent over-fitting, I will employ a modified form of k-fold cross validation that will remove all gestures performed by k subjects from the data set. In the case of a supervised algorithm, I will fit the model to the training data and test on the hold out data. This cross validation method can be used to test the quality of unsupervised

methods as well. For the unsupervised algorithms, I will generate clusters from the training data and then see how well the hold-out data fits into these clusters via a cluster purity metric or the like.

3. *Inability to identify important features* - This is the greatest risk to the project goals. Inability to detect latent features in the modalities could be due to a number of factors including data quality, sensor imprecision, etc. If this scenario arises, various transformations will be employed on the data set in an attempt to uncover these features. Should that fail to mitigate the problem, I will explore alternative sEMG data sets.

References

[1] Lobov, Sergey, et al. Latent Factors Limiting the Performance of sEMG-Interfaces. Sensors, vol. 18, no. 4, June 2018, p. 1122., doi:10.3390/s18041122.

[2] Tehrani, Kiana, and Andrew Michael. Wearable Technology and Wearable Devices: Everything You Need to Know. Wearable Devices Magazine, WearableDevices.com, March 2014. Web.

[3] Peerdeman B, Boere D, Witteveen H, Huis in t Veld R, Hermens H, Stramigioli S, Rietman H, Veltink P, Misra S. Myoelectric forearm prostheses: State of the art from a user-centered perspective. J Rehabil Res Dev. 2011;48(6): 71938. DOI:10.1682/JRRD.2010.08.016

[4] Atzori, M.et al.Electromyography data for non-invasive naturally-controlled robotic hand prostheses.Sci. Data1:140053 doi: 10.1038/sdata.2014.53 (2014).

[5] Phinyomark, A.; Scheme, E. EMG Pattern Recognition in the Era of Big Data and Deep Learning. Big Data Cogn. Comput. 2018, 2, 21.

[6] Phinyomark Angkoon, Khushaba Rami N., Ibez-Marcelo Esther, Patania Alice, Scheme Erik and Petri Giovanni Navigating features: a topologically informed chart of electromyographic features space14J. R. Soc. Interface

[7] Tralie, Christopher J., et al. Geometric Cross-Modal Comparison of Heterogeneous Sensor Data. 2018 IEEE Aerospace Conference, 2018, doi:10.1109/aero.2018.8396789.

[8] Tralie, Bendich, & Harer. Multi-Scale Geometric Summaries for Similarity-Based Sensor Fusion. 2019 IEEE Aerospace Conference, 2019, doi:10.1109/aero.2019.8741399.

[9] Holt, G.A., et al. Gesture Recognition Using Skeleton Data with Weighted Dynamic Time Warping. Proceedings of the International Conference on Computer Vision Theory and Applications, 2013, doi:10.5220/0004217606200625.

[10] EMG Data for Gestures Data Set. UCI Machine Learning Repository: EMG Data for Gestures Data Set, https://archive.ics.uci.edu/ml/datasets/EMG data for gestures.