Blind Source Separation using Machine Learning Approaches

Tung-Cheng Su
School of Music
Carnegie Mellon University
Pittsburgh, PA 15213
tungches@andrew.cmu.edu

Kaiwen Lan
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
klan@andrew.cmu.edu

Ziyun Liu
School of Music
Carnegie Mellon University
Pittsburgh, PA 15213
ziyunliu@andrew.cmu.edu

Yaguang Li
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
yaguangl@andrew.cmu.edu

Abstract

Blind source separation (BSS) is widely used to identify motor unit (MU) sources from high-density electromyography (HDEMG) data. In this project, we use various filters in the preprocessing period and employ Fast Independent Component Analysis (FastICA) algorithm to separate mixed signals and accurately determine the number of MUs by removing duplicate MUs in the postprocessing phase. This combined approach improves the identification of MUs from HDEMG data.

1 Introduction and Research Background

1.1 Introduction

BSS is the problem of finding one or more base sources from the given mixed signals, under the assumption that all the sources are independent, and with very little or no priori knowledge of the sources. In neurobiology, BSS is essential in recovering multiple neuron motor action potentials from the mixed electrical signals generated by the conduction of neuron motors and muscle fibers. The problem is to identify unique MUs that composed the signals, with the assumption that the neuron motors are independent. The experiment HDEMG data is collected based on the tibialis anterior muscle measurement of human wrist flexion. Two sets of increasing force data and one set of steady force data collected by the 8 × 8 electrodes on the forearm skin are going to be analysed. The significance of this problem is that it provides us with a novel way to non-invasively examine potential central (CNS) and peripheral (PNS) nervous systems diseases by measuring the surface electromyogram on the skin [4]. The result can also be used for pathophysiological investigations and diagnostic researches.

1.2 Related Work

Several different methods can be used to decouple mixed signals to get individual motor units. Garcia *et el.* [3] combined joint approximate diagonalization of eigen-matrices (JADE) [1] with FastICA [7] to get independent α-motor units from the original data. Garcia *et el.* [11] introduced sparsity in signal processing using sparse NMF and sparse component analysis (SCA) to separate source signal. Meanwhile, Jiang *et el.* [8] applied second-order blind identification (SOBI) algorithm to separate

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

mixtures of multiple sources, proving superior to previous methods. Muceli *et el.* [9] focused on forearm signal extraction, using NMF approach on the electromyogram recordings. In this research, the number of activation primitives remained 4 regardless of the number of electromyogram channels.

A significant improvement on BSS problem was the introduction of CKC algorithm [6], which did not require MUs to be temporally independent. It estimated MUs discharge patterns without knowing the mixing matrix. Furthermore, a gradient-based optimization algorithm was developed based on the framework of CKC approach [5]. The result of gradient CKC method had a higher number of reconstructed innervation pulse trains (IPT) and lower false positive rate comparing to the sequential probabilistic CKC [4]. This algorithm held without depending upon the spatially uncorrelation of the sources, which increased the robustness of the estimation of individual signals.

2 Dataset

The experiment HDEMG data is collected based on the tibialis anterior muscle measurement of human wrist flexion. Two sets of increasing force data and one set of steady force data collected by the 8 × 8 electrodes on the forearm skin are going to be analysed. The distance between inner electrode is 8.75mm. The inputs of the data is the electrode array that are attached to the forearm skin and the output of the data are electromyographic signals detected by the sensors. The dataset is considered to be incomplete, which means additional preprocessing and postprocessing techniques should be adopted to produce better results.

3 Experiments and Results

3.1 NMF Baseline

The NMF algorithm is nonparametric, efficient and applicable without any prior training or calibration [4], which serves as a baseline of our project. The basic idea of NMF is separating a non-negative data matrix into two non-negative matrices, one of which is the separated source. By minimizing the reconstruction error, NMF captures the underlying structure of the data and separates it into distinct components.

We utilize a Butterworth bandpass filter and a notch filter for the baseline method. The Butterworth bandpass filter employs cutoff frequencies to determine the range of frequencies that will be allowed to pass through the filter to the subsequent stages of analysis. The notch filter is designed to attenuate specific frequencies, typically associated with noise or interference that are present in the mixing data.

After putting the increasing force 1 dataset into a Butterworth bandpass filter and a notch filter, we apply NMF to the filtered data and get 8 different MU potential actions. After that, we use peak detection algorithm to detect the peaks of the potential actions. The inter-spike interval of different peaks are between 20-30 HZ, which is in accordance with biological measurements [2].

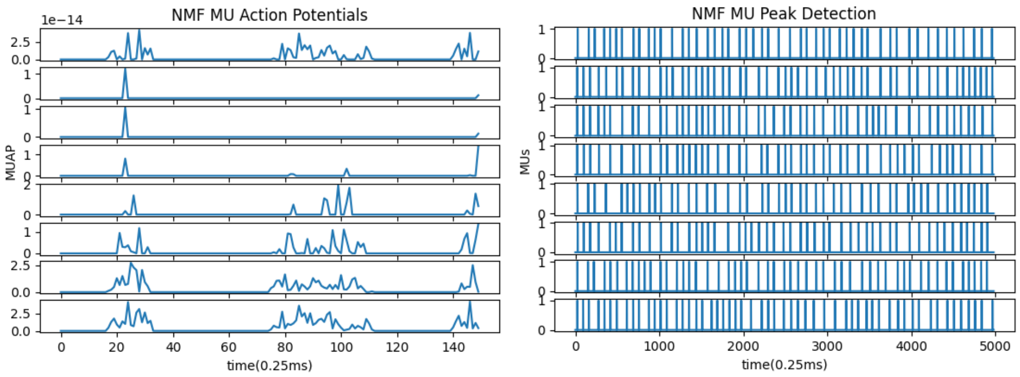

Figure 1: NMF results of increasing force 1: MUAPs and MU peaks

The NMF results seems to work, however, it has several weaknesses that needs to be further eliminated:

Number of MUs NMF can not fully determine the number of MUs based on the HDEMG data. We manually set the number of separated sources and feed it as a parameter to the NMF algorithm. Although the 8 MU action potentials lies inside the range of frequency, we have no idea if there are extra MUs that have not been detected by the NMF algorithm.

Postprocessing This algorithm has a second weakness related to its postprocessing phase. It lacks a comprehensive step that addresses the removal of duplicate MUs and the calculation of silhouette values for each individual MU. As a result, there is a possibility of having two sequential MUs that appear to be duplicated but actually originate from the same MU within the body.

3.2 FastICA Experiments

The FastICA algorithm is based on the principle of maximizing non-Gaussianity in the transformed signals. In each iteration, it computes a weight vector that represents the direction of the estimated independent component. The weight vector is updated by taking into account of the non-Gaussianity (e.g. approximation of negentropy) of the projected data onto the current estimation of the independent component. This update process continues until convergence, where the estimated components are as independent as possible. The algorithm seeks to find a linear transformation that maximizes the statistical independence of the components.

In order to achieve a better result in the FastICA algorithm, some additional steps need to be done before and after applying the FastICA algorithm:

Matrix Extension In order to effectively identify independent resources in the FastICA algorithm, we introduce the extended HDEMG data for analysis. The extended data copies the original signals based on the extension parameter, with each copy shifted to right for one sample. The reason for doing extension is that HDEMG signals typically exhibit temporal correlation among multiple components, indicating the presence of time-delay relationships between different components. By duplicating and delaying the original signal, the expanded matrix can better capture this temporal correlation, enabling the decomposition algorithm to more accurately separate different components.

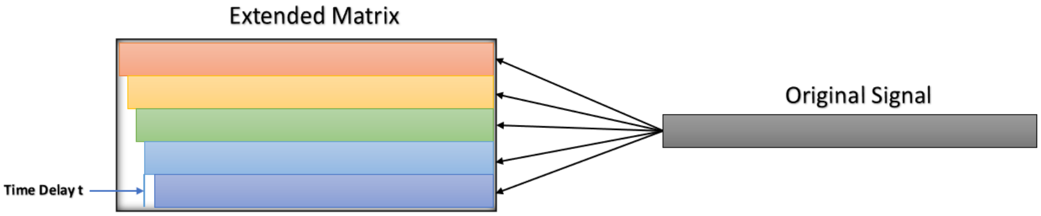

Figure 2: Extended Signals for Future Analysis

In addition, the centralization and whitening process is employed in ICA decomposition. Whitening is applied to transform the input signals into independent and unit-variance signals, creating a stronger basis for subsequent independent component analysis. By implementing whitening, the FastICA algorithm becomes more effective in separating the independent components within the input signals.

K-means++ After getting the independent components of the original data, we adopt a k-means++ clustering technique to identify the spikes of each independent component. Comparing with k-means algorithm, k-means++ is less likely to be affected by the initialization of cluster centroids, leading to better convergence and more accurate clustering results. In order to effectively cluster spikes from the mixed noises, a quadratic kernel function is used for the k-means clustering. Based on the non-linear property of quadratic function, the distance of a one-dimensional vector can be enlarged when mapping into a two-dimensional space, which is easier for the separation of useful signals from the noises.

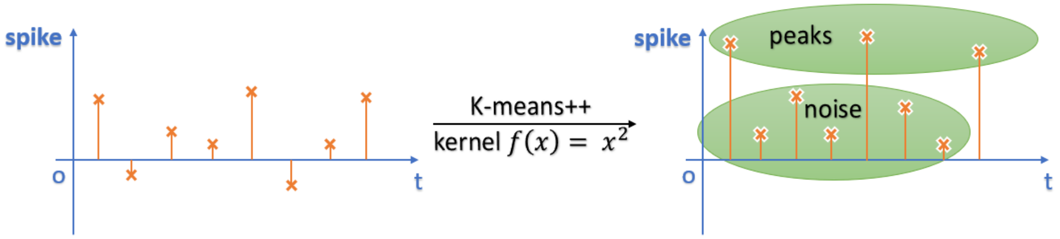

Figure 3: k-means++ with quadratic kernel function

Select Feasible MUs In the postprocessing stage, we apply additional criteria to select physiologically plausible sources. For example, one criterion may involve selecting sources with frequencies between 4 HZ and 35 Hz, as these frequencies are often associated with neural activity of interest.

Moreover, the spike trains within a good source may contain spikes that are very close to each other, with a high spike rate of more than 50 Hz. However, such closely spaced spikes are not typically observed in physiological spike trains. To address this, a typical approach is to choose only the spike with the higher peak value and remove redundant spikes. This helps to ensure that the resulting spike trains align more closely with the expected physiological characteristics.

Another important step in the postprocessing phase is the calculation of the silhouette value for each spike train. The silhouette value is a measurement of how different a sample is from other samples in its own cluster comparing with samples in other clusters. It provides a quantitative assessment of the compactness of clusters in the data. A higher silhouette value indicates that the samples within a cluster are closer to each other, exhibiting higher similarity, while being well-separated from samples in other clusters. On the other hand, a lower silhouette value suggests that the samples within a cluster are less similar to each other or are closer to samples in other clusters, indicating potential overlapping and ambiguity.

By calculating the silhouette value for each MU spike train, we choose MU spike trains with high silhouette values, as they exhibit stronger cohesion within the cluster and clear separation from other clusters. Only spike trains with a high silhouette value can be selected for further analysis and visualization.

3.3 FastICA Results

We set the extension parameter to be 4, and the number of iteration to be 100. We put each of the three sets of data into our FastICA model, iterating and computing the decomposed MUs on each of the dataset. The results are shown in Figure 4, 5 and 6:

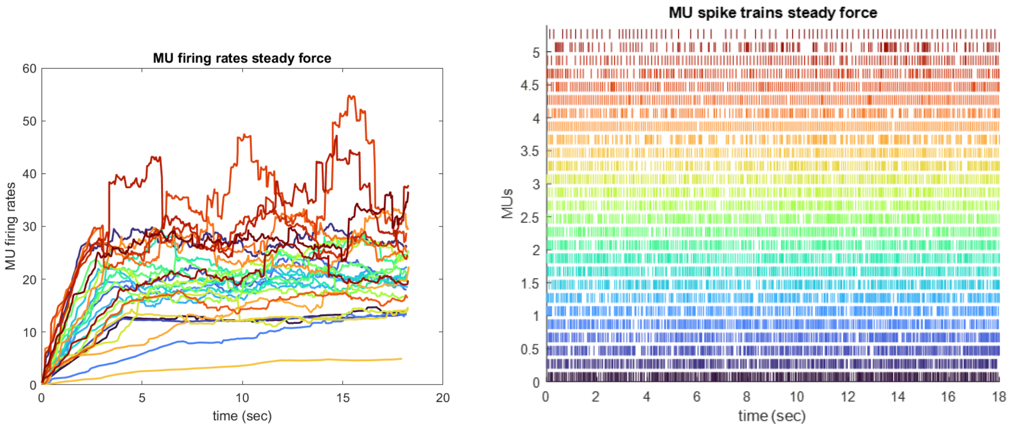

Figure 4: Steady force MU firing rates and MUs

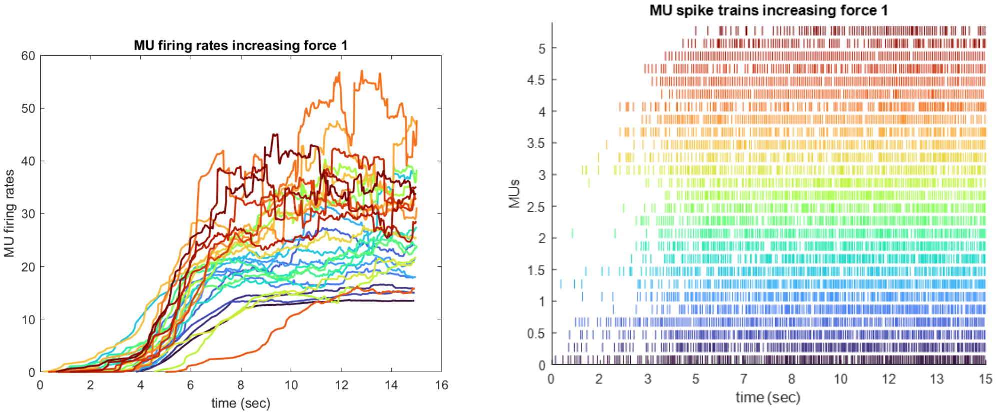

Figure 5: Increasing force 1 MU firing rates and MUs

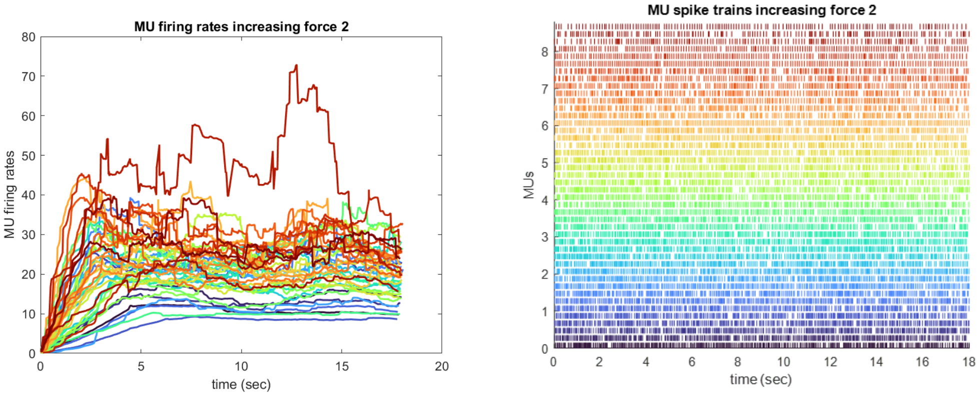

Figure 6: Increasing force 2 MU firing rates and MUs

The evaluation metrics we use is inter-spike intervals, since we rule out the independent components with frequency less than 4HZ or more than 35HZ in time series. We also remove duplicate MUs that are close with each other (e.g. the time between each spike is less than 20ms), and select only the MU spike trains with a silhouette value greater than 0.5. We get 27 MUs in the steady force 1 dataset, 27 MUs in the increase force 1 dataset and 44 MUs in the increasing force 2 dataset. This turns out to be a more exhaustive result than the NMF since it captures more unique MUs comparing to the previous NMF result. Part of out code is referred from this github project [10].

4 Discussion and Analysis

4.1 Discussions

FastICA Iteration Numbers We found that a small number of iteration can effectively improve the speed of the code, however, it may lose some information of independent MUs. An iteration number of 20 will only produce around 12-15 independent MUs on steady force, while a number of 200 will produce more than 30 independent components. An overly large iteration number will dramatically increase the computation complexity, while not making much improvement on the performance. Therefore, we choose 100 as the iteration number.

Comparison between Datasets From the results we found that the time stamps are different among different datasets. That's because the actual measurement time is different when collecting the HDEMG data. The steady force and increasing force 2 datasets have a total of 18 seconds, while the increasing force 1 has a total of 15 seconds. Time stamps are different when comparing these three datasets.

Another difference is the MU firing rates between the dataset. In steady force, since the muscle is at resting state, there are not many activities of the MUs. By the time of 3 second, most of the MUs have reach to their peak state, and the firing rates remain unchanged in the rest of the time.

Increasing force 1 is the case where the MU firing rates is increasing as time goes by. The MUs begin to activate at 4 second, having a dramatic increase from 4 second to 8 second, and continue to increase until 15 second. From the result figures we determine the actual activity of the tibialis anterior muscle, which can help us identify muscle movement as well as whether it is healthy or not.

The increasing force 2 dataset has an early activation at around 2 second, which is similar to the steady force dataset. However, the increasing force 2 dataset has a relatively higher firing frequency comparing with the steady force, indicating a potential increase of muscle activity. It is important to point out that the increasing force 2 dataset is different from the increasing force 1 dataset, indicating two different ways of muscle activities.

The 8 × 8 Electrodes The input array of the datasets may have some effect on the sequential analysis. In this case, we use an 8 × 8 electrodes, with each electrode has a distance of 8.75mm. This might give us some important information on the MU distributions since the electrodes in a single column may have recorded the same MU with a time delay. If we can capture the delayed MU in the corresponding electrode, we might be able to identify each MU without using FastICA algorithm. However, in our project, we did not use this information as well as the 8.75mm distance between different electrodes.

4.2 Future Improvements

Model Limitations The Independent Component Analysis (ICA) algorithm used in this study relies on the assumption of temporal independence among the sources and assumes that each Motor Unit (MU) discharge occurs independently of previous discharges [4]. The temporal dependencies and interactions among MUs may introduce complexities that challenge the effectiveness of ICA-based techniques in accurately identifying and separating different sources.

CKC and Gradient CKC An alternative approach of doing this problem is the CKC approach. The CKC algorithm directly estimate the MU discharge patterns without mixing data. This is a significant step forward in identifying sources directly from the original data. The gradient CKC uses a gradient optimization of the non-linear cost function of the estimated MU discharge pattern has been used to iteratively improve the MU identification [4]. It does not require the assumption of spatially uncorrelated between the sources to hold. In this way, the gradient CKC algorithm is more robust than the FastICA algorithm, which may produce better results.

References

[1] J. Cardoso. High-Order Contrasts for Independent Component Analysis. *Neural Computation*, 11(1):157–192, 01 1999.

[2] M. Roos D. Connelly, C. Rice and A. Vandervoort. Motor unit firing rates and contractile properties in tibialis anterior of young and old men. *Appl Physiol*, 1999.

[3] G.A. Garcia, R. Okuno, and K. Azakawa. A decomposition algorithm for surface electrode-array electromyogram. *IEEE Engineering in Medicine and Biology Magazine*, 24(4):63–72, 2005.

[4] A. Holobar and D. Farina. Blind source identification from the multichannel surface electromyogram. *Physiological Measurement*, 35:R143 – R165, 2014.

[5] A. Holobar and D. Zazula. Gradient convolution kernel compensation applied to surface electromyograms. In *International Conference on Agents*, 2007.

[6] A. Holobar and D. Zazula. Multichannel blind source separation using convolution kernel compensation. *IEEE Transactions on Signal Processing*, 55(9):4487–4496, 2007.

[7] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[8] N. Jiang and D. Farina. Covariance and time-scale methods for blind separation of delayed sources. *IEEE Transactions on Biomedical Engineering*, 58(3):550–556, 2011.

[9] S. Muceli, N. Jiang, and D. Farina. Extracting signals robust to electrode number and shift for online simultaneous and proportional myoelectric control by factorization algorithms. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(3):623–633, 2014.

[10] Shirazi S.Y. hdemg-decompostion (github.com/neuromechanist/hdemg-decomposition/tag/0.1). 2022.

[11] F.J. Theis and G.A. Garcia. On the use of sparse signal decomposition in the analysis of multi-channel surface electromyograms. *Signal Processing*, 86(3):603–623, 2006.