

Decomposition of Muscle Activity for Sensorimotor Neuroscience

Final Report

Daniel King, Jasmine Ortega, Rada Rudyak, Rowan Sivanandam

Mentor: Dr. Alexi Rodriguez-Arelis

Partner: Dr. Jean-Sébastien Blouin, Sensorimotor Physiology Laboratory, UBC

2022-06-21

Contents

Executive Summary	1
Introduction	1
Data Science Methods	1
Data Product and Results	3
Conclusions and Recommendations	3
References	3

Executive Summary

Introduction

Data Science Methods

The partner wants us to replicate the blind source separation algorithm used in Negro et al. (2016) to decompose raw EMG signals into their constituent motor unit spike trains. This algorithm provides advantages to other blind source separation algorithms in that it is an experimentally validated algorithm that allows for the decomposition of multi-channel invasive EMG and non-invasive EMG data.

Essentially, the blind source separation algorithm is an unsupervised machine learning model that iteratively extracts separation vectors from the data. Separation vectors are vectors that, when applied to the pre-processed data, separate a single motor unit spike train from unwanted noise and other motor unit spike trains. A motor unit spike train contains the firing times of a single motor unit, and it is the mixture of these. This is done through broadly three steps:

1. The data is pre-processed.
2. Latent component analysis (LCA) is performed to estimate a separation vector.
3. The refinement process is used to increase the quality of the estimated separation vector.

Steps two and three are repeated for a pre-determined amount of iterations of the overall blind source separation algorithm.

The first step of the algorithm is data pre-processing. Pre-processing occurs in a four step process. First the data is band-pass filtered. Then each channel is centred by subtracting the mean of each channel. Then each channel is extended by a certain number of time-delayed versions of that channel. Finally, the data is whitened, so that the covariance matrix of the channels and their extensions is equal to the identity matrix. Band-pass filtering removes noise, and increases the amount of unique motor units identified. Centering

the data makes the whitening process faster and is necessary for LCA to find all of the separation vectors. Extending the data is done for the purpose of converting the mixture of signals from a convolutive mixture to a linear instantaneous mixture, which is necessary for latent component analysis to work (Negro et al. 2016; Weenink 2012). Whitening is done to computationally simplify and speed up the convergence of the LCA step (Hyvärinen, Karhunen, and Oja 2001).

The pre-processed data then goes through the LCA step. The LCA step is based off of independent component analysis, where the separation vectors would be obtained by maximizing their statistical independence from each other. However, since the motor unit spike trains are extended along with the observations, they cannot be fully independent from each other. This is why LCA extracts separation vectors by maximizing sparsity instead of independence. Intuitively, this is done because a singular motor unit spike train will always be more sparse than the combination of multiple motor unit spike trains (Negro et al. 2016). First, the separation vector is initialized by using a time instance of high activity in the whitened data, as these time instances are likely to correspond to multiple motor unit firings. Using a contrast function that measures sparsity of the motor unit spike train, the estimated separation vector is iteratively updated. In each iteration, the estimated separation vectors are orthogonalized against previously accepted separation vectors and normalized, to increase the number of unique motor unit spike trains that are extracted. The LCA step converges when the separation vector no longer changes, within a tolerance.

After the separation vector is extracted, it goes through the refinement step. This is done because the latent component analysis may converge to unreliable estimates and through refinement the quality of the estimate is increased. The refinement step is an iterative algorithm that maximizes the regularity of the motor unit spike train. This process is carried out under the assumption that singular motor units will fire off action potentials at a much more regular rate than combinations of motor units (Negro et al. 2016). First, the motor unit spike train is estimated by applying the separation vector to the pre-processed data. Then the firing times are determined by applying the peak-finding algorithm from Virtanen et al. (2020). Of these firing times, the instances that correspond to small peaks in the spike train are separated away from the large peaks using the KMeans algorithm from Pedregosa et al. (2011). The firing times corresponding to small peaks are discarded as they likely correspond to the firing occurrence of more than one motor unit (Negro et al. 2016). The information from the accepted firing times are used to update the separation vector. The iterative refinement process converges once the coefficient of variation of the time between firings increases, which in other words means the regularity decreases.

Once the refinement process is done, the refined separation vector is accepted based on a user-defined threshold of either the silhouette score between the signal and the noise or the pulse-to-noise ratio. The accepted separation vectors correspond to the motor units that the blind source separation algorithm extracts from the raw signal.

A further improvement to the algorithm that we we did not have time to implement would be a re-learning feature. The user would run the algorithm on a sample of the data, and based on that sample identify firing times that are inaccurate. Then they would run the algorithm on the rest of the data and based on the firing times identified as inaccurate, the algorithm would no longer make similar mistakes in the rest of the decomposition. Implementing this feature would be quite complex because it is unclear how this would be done. One idea was that we somehow change the initialization of the separation vectors so that they no longer identify the false firing times when applied to the pre-processed data. However, since the separation vector changes throughout the LCA and refinement processes, it would be hard to control the effect that it actually has on the estimated firing times. Another approach would be to influence the KMeans algorithm so that the threshold for the small peaks cluster includes the peaks that were as false firings, in the hopes that future peaks of similar size are also false firings. The downside to this approach would be that we may increase the amount of incorrect identifications of large peaks as small peaks, which are discarded.

The stakeholders affected by our blind source separation algorithm are researchers and those that would be affected by their research. This is why it is important our algorithm works properly so that researchers results are accurate and do not affect the general public adversely down the line. For example, if someone uses `EMGdecompPy` and obtains inaccurate results, and these results are used to inform a neuromuscular diagnosis down the line, it could greatly affect someone's life. We have not had the chance to thoroughly

validate our algorithm, as the debugging process took a great deal of time. We have only received qualitative results, obtained by visually comparing MUAP shapes identified by **EMGdecompPy** and those Hug et al. (2021) identified using the DEMUSE tool. There are concerns with this approach as the DEMUSE software uses a similar but different algorithm than Negro et al. (2016). The DEMUSE is a highly validated software in comparison to OT Bioelettronica, and therefore the partner wishes to compare our results to theirs.

Data Product and Results

Our final data product consists of two elements. The first element is the python package, **EMGdecompPy**, and its associated [GitHub repository] containing the replicated Negro et al. (2016) blind source separation algorithm. The second element is a Jupyter notebook containing a workflow leading up to an interactive dashboard visualizing the decomposition results.

Creating a python package allows anyone who would like to decompose EMG signals into constituent motor units to simply download the package off of pip and run the **decomposition** function with the desired hyper-parameters on their data. Using a python package to deliver our blind source separation algorithm not only increases accessibility but also readability of the code by breaking the blind source separation algorithm into many different functions with informative docstrings. By giving the partner access to the GitHub repository for **EMGdecompPy** we allow them to view the source code and even edit it if they desire to add more functionality in the future. These attributes of our data product satisfy the partner’s desires to replicate the blind source separation algorithm from Negro et al. (2016) without experimental duration limits, while providing them with easy-to-understand open source code which means they will understand how their experimental results are derived.

The Jupyter notebook provides the partner with an interface in which to use **EMGdecompPy**. This notebook further facilitates our package’s ease-of-use, as loading in the data, running the algorithm, and visualizing the data, will be laid out in a step-by-step process. Essentially, the Jupyter notebook will eliminate all confusion in how to use the package, and can be re-used to analyze different data, or to change hyper-parameters.

Conclusions and Recommendations

References

- Hug, Francois, Simon avrillon, Alessandro Del Vecchio, Andrea Casolo, Jaime Ibáñez, Stefano Nuccio, Julien Rossato, Aleš Holobar, and Dario Farina. 2021. “Analysis of motor unit spike trains estimated from high-density surface electromyography is highly reliable across operators,” February. <https://doi.org/10.6084/m9.figshare.13695937.v1>.
- Hyvärinen, Aapo, Juha Karhunen, and Erkki Oja. 2001. “What Is Independent Component Analysis?” In *Independent Component Analysis*, 158–61. Wiley.
- Negro, Francesco, Silvia Muceli, Anna Margherita Castronovo, Ales Holobar, and Dario Farina. 2016. “Multi-Channel Intramuscular and Surface EMG Decomposition by Convolutional Blind Source Separation.” *Journal of Neural Engineering* 13 (2): 026027. <https://doi.org/10.1088/1741-2560/13/2/026027>.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. “Scikit-Learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12: 2825–30.
- Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, et al. 2020. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.” *Nature Methods* 17: 261–72. <https://doi.org/10.1038/s41592-019-0686-2>.
- Weenink, David. 2012. “Blind Source Separation.” *Blind Source Separation*. https://www.fon.hum.uva.nl/praat/manual/blind_source_separation.html.