The iCanClean Algorithm: How to Remove Artifacts using Reference Noise Recordings

Ryan J. Downey1* and Daniel P. Ferris1

1 J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida, Gainesville FL, USA

*Disclosure: R. J. Downey is the inventor of a pending patent related to the concepts described in the present manuscript.

Abstract

Data recordings are often corrupted by noise, and it can be difficult to isolate clean data of interest. For example, mobile electroencephalography is commonly corrupted by motion artifact, which limits its use in real-world settings. Here, we describe a novel noise-canceling algorithm that uses canonical correlation analysis to find and remove subspaces of corrupted data recordings that are most strongly correlated with subspaces of reference noise recordings. The algorithm, termed iCanClean, is computationally efficient, which may be useful for real-time applications, such as brain computer interfaces. In future work, we will quantify the algorithm’s performance and compare it with alternative cleaning methods.

1. Introduction

Noise affects the fidelity of data recordings in many experiments. Algorithms for removing noise could lead to better scientific conclusions from experimental studies. There are many options for removing noise from data recordings, such as frequency-based filtering [1], adaptive filtering [2], wavelets [3], [4], independent component analysis [3], [5], and other blind source separation techniques [3], [4]. However, there is no single algorithm which is optimal for all scenarios. Efficacy of these approaches varies depending on the makeup of the recordings (e.g., transient versus static noise, large versus small amplitude noise, etc.). Here we describe a novel algorithm, termed iCanClean, that may have applications in high-density electroencephalography and other biomedical recording modalities.

2. The iCanClean Algorithm

The iCanClean algorithm consists of four steps. First, canonical correlation analysis is used to identify candidate noise components that project onto both the corrupted data recordings as well as the reference noise recordings. Second, a subset of noise components is selected for removal. Third, the projection from the bad components to the data channels is calculated. Fourth, the projected noise components are directly subtracted from the data channels. This process can be applied both to a large window of data and to a smaller sliding window to deal with static and/or transient noise.

Additional mathematical detail is subsequently provided, with key equations written in MATLAB format using built-in MATLAB functions. Variables are bolded. Functions are italicized. A list of variables is provided in Table 1, along with their description and dimensions.
### Table 1: List of variables

| Symbol         | Description                                      | Dimensions          |
|----------------|--------------------------------------------------|---------------------|
| A              | Unmixing matrix (data)                           | $N_{\text{Data}} \times N_{\text{Comp}}$ |
| B              | Unmixing matrix (noise)                          | $N_{\text{Noise}} \times N_{\text{Comp}}$ |
| BadCompActivity| Noise components selected for removal            | $T \times N_{\text{Bad}}$ |
| BadCompList    | Indexed list of bad components                   | $N_{\text{Bad}} \times 1$ |
| $N_{\text{Bad}}$| Number of noise components selected for removal  | Scalar              |
| $N_{\text{Comp}}$| Number of candidate noise components            | Scalar              |
| $N_{\text{Data}}$| Number of data channels                          | Scalar              |
| $N_{\text{Noise}}$| Number of noise channels                         | Scalar              |
| ProjectionMatrix | Relates noise component space to data channel space | $N_{\text{Bad}} \times N_{\text{Data}}$ |
| ProjectedNoise | Bad noise components projected onto the data channels | $T \times N_{\text{Data}}$ |
| R              | Correlation between each $(U_i, V_i)$ pair        | $N_{\text{Comp}} \times 1$ |
| $T$            | Number of time points (samples)                  | Scalar              |
| Thresh         | User defined threshold for identifying bad components | Scalar              |
| U              | Candidate noise components (calculated from data channels) | $T \times N_{\text{Comp}}$ |
| V              | Candidate noise components (calculated from noise channels) | $T \times N_{\text{Comp}}$ |
| X              | Corrupted data recordings                        | $T \times N_{\text{Data}}$ |
| $X_{\text{Clean}}$ | Cleaned data                                   | $T \times N_{\text{Data}}$ |
| $X_{\text{MC}}$ | Mean centered data                              | $T \times N_{\text{Data}}$ |
| Y              | Reference noise recordings                       | $T \times N_{\text{Noise}}$ |
| $Y_{\text{MC}}$ | Mean centered noise                             | $T \times N_{\text{Noise}}$ |

**Definitions**

Let $X$ be the corrupted data recordings the user wishes to clean with dimensions $T \times N_{\text{Data}}$, where $T$ is the number of time points (or samples) and $N_{\text{Data}}$ is the number of data channels. Similarly, let $Y$ be the reference noise recordings with dimensions $T \times N_{\text{Noise}}$, where $N_{\text{Noise}}$ is the number of reference noise channels.

**Step 1**

Given corrupted data to clean ($X$) and reference noise recordings ($Y$), canonical correlation analysis is used to identify latent sources of noise (i.e., candidate noise components) in common to both $X$ and $Y$.

$$[A, B, R, U, V] = \text{canoncorr}(X, Y);$$
A is an unmixing matrix that converts corrupted data recordings to candidate noise components as \( U = X_{MC}^*A \), where \( X_{MC} \) is the mean centered data. Similarly, \( B \) is an unmixing matrix that converts reference noise recordings to a second set of candidate noise components as \( V = Y_{MC}^*B \). Finally, \( R \) is a vector which contains the correlation between each unique \((U_i, V_i)\) pair, where \( U_i \) and \( V_i \) are the \( i \)th columns of \( U \) and \( V \), respectively. The number of candidate noise components identified, \( N_{Comp} \), depends on the rank of the data. Specifically, \( N_{Comp} = \min( \text{rank}(X), \text{rank}(Y) ) \). Therefore, \( N_{Comp} \leq \min( N_{Data}, N_{Noise} ) \).

With the iCanClean approach, we use corrupted data recordings as one set of inputs to canonical correlation analysis and reference noise recordings as the second set of inputs. Canonical correlation seeks to find the subspaces of these two datasets which are maximally correlated with each other. Because the corrupted data recordings and reference noise recordings both contain noise, canonical correlation analysis will identify hidden noise components that are common to both datasets.

Canonical correlation analysis returns candidate components in ranked order. Thus, the noise component pair with the strongest relationship (i.e., largest \( R^2 \) correlation) appears first \((U_1, V_1)\). The next noise component pair \((U_2, V_2)\) has the second largest \( R^2 \) correlation and is independent of the first component pair, and so forth. The noise components identified by canonical correlation do not depend on how strongly the noise sources project onto the data channels or noise channels. Thus, both large and small noise sources are identified. In a subsequent step, candidate noise components will be appropriately scaled to match the channels being cleaned.

**Step 2**

A bad subset of components is identified. We suggest a basic thresholding technique where all components with an \( R^2 \) value \( \geq \) a user-defined threshold are rejected. Alternative approaches can be employed as well, for example, by examining the frequency content of the candidate noise components. The user has a choice of whether to select \( U \) or \( V \), or a combination of \( U \) and \( V \) as their noise components. The best choice may vary by each specific application. As an example, we assume here that the user wishes to use mixtures (or subspaces) of the data channels to clean the data channels themselves (i.e., use \( U \) to clean \( X \)).

\[
\text{BadCompList} = \text{find}( R.^2 \geq \text{Thresh} );
\]

\[
\text{BadCompActivity} = U( :, \text{BadCompList} );
\]

**Step 3**

The projection from the noise components onto the channels is calculated. We recommend using a least squares solution (e.g., using matrix division in MATLAB). Alternative options for calculating the projection include applying a Moore-Penrose pseudoinverse to the \( A \) and/or \( B \) unmixing matrices, but we have found that it does not perform as well in practice.

\[
X_{MC} = X - \text{mean}(X);
\]

\[
\text{ProjectionMatrix} = \text{mldivide}( \text{BadCompActivity}, X_{MC} );
\]

\[
\text{ProjectedNoise} = \text{BadCompActivity} * \text{ProjectionMatrix};
\]
Step 4

The projected noise is subtracted from the corrupted data channels.

\[ X_{\text{Clean}} = X - \text{ProjectedNoise}; \]

Because the noise components are calculated as linear mixtures of the original recordings (\(X\) and/or \(Y\)) and because the projection onto the channels is also linear, iCanClean noise cancellation resembles a spatial filter. This filter can be static or time-varying depending on the application.

3. Discussion

In the present work, we described a novel noise canceling algorithm which cleans corrupted data using reference noise recordings. The iCanClean algorithm has potential for cleaning various biomedical signals such as electroencephalography and electromyography. The algorithm is computationally efficient, which may be useful for real-time applications such as brain computer interfaces. In future work, we will examine iCanClean’s ability to remove artifacts from electroencephalography using an electrical head phantom, similar to [6], where ground-truth signals are known. This will allow us to quantify its performance relative to alternative cleaning methods.

4. References

[1] J. Gotman, J. R. Ives, and P. Gloor, “Frequency content of EEG and EMG at seizure onset: Possibility of removal of EMG artefact by digital filtering,” Electroencephalogr. Clin. Neurophysiol., vol. 52, no. 6, pp. 626–639, 1981, doi: 10.1016/0013-4694(81)91437-1.

[2] D. Steyrl, G. Krausz, K. Koschutnig, G. Edlinger, and G. R. Müller-Putz, “Reference layer adaptive filtering (RLAF) for EEG artifact reduction in simultaneous EEG-fMRI,” J. Neural Eng., vol. 14, no. 2, p. 026003, 2017, doi: 10.1088/1741-2552/14/2/026003.

[3] D. Safieddine et al., “Removal of muscle artifact from EEG data: Comparison between stochastic (ICA and CCA) and deterministic (EMD and wavelet-based) approaches,” EURASIP J. Adv. Signal Process., vol. 2012, no. 127, 2012, doi: 10.1186/1687-6180-2012-127.

[4] M. Rakibul Mowla, S.-C. Ng, M. S. A. Zilany, and R. Paramesran, “Artifacts-matched blind source separation and wavelet transform for multichannel EEG denoising,” Biomed. Signal Process. Control, vol. 22, pp. 111–118, Sep. 2015, doi: 10.1016/j.bspc.2015.06.009.

[5] O. Dimigen, “Optimizing the ICA-based removal of ocular EEG artifacts from free viewing experiments,” Neuroimage, vol. 207, p. 116117, Feb. 2020, doi: 10.1016/j.neuroimage.2019.116117.

[6] N. Richer, R. J. Downey, W. D. Hairston, D. P. Ferris, and A. D. Nordin, “Motion and Muscle Artifact Removal Validation Using an Electrical Head Phantom, Robotic Motion Platform, and Dual Layer Mobile EEG,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 28, no. 8, pp. 1825–1835, Aug. 2020, doi: 10.1109/TNSRE.2020.3000971.