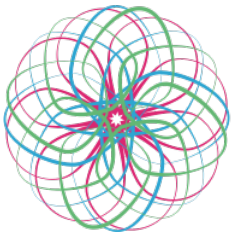

Automated EOG artifact correction in EEG data using Second Order Blind Inference

Documentation

Lisa Tostrams

Research project conducted at the
Centre for Human Drug Research

Supervisors:
dr. Y.M. Miranda (CHDR)
dr.ir. R.J. Doll (CHDR)

Leiden, January 2018

Contents

1 Introduction

One of the methods used in the Center for Human Drug Research (CHDR) to asses the effect of drugs on the central nervous system is the quantification of resting state electro-encephalography (EEG). For the duration of a few minutes, cortical activity is measured in subjects during periods with eyes closed and periods with eyes opened. From the calculated power spectrum of the recorded signals some indications about the state of the central nervous system can be extracted.

One of the issues while recording EEG is that the amplitude of EEG signals is relatively low in comparison to artifacts. Common artifacts in EEG recordings are caused by the propagation of ocular (EOG) and muscular activity through the skull. Without correction, the calculated power spectrum is contaminated by these artifacts. For example, ocular activity in EEG recordings results in an overestimation of the low-frequency components, while muscular activity is generally found in the high-frequency components.

Currently at CHDR, the Grubbs' test for outliers is used to detect artifacts in EEG recordings, and the contaminated parts are omitted from further analysis. As a result, there may be a loss of relevant information as the final endpoints are based on fewer data-samples. Alternatives use Machine Learning (ML) techniques for correcting signals such that only the artifact is removed.

In this project, an automated EOG correction technique was implemented and validated for the use in resting state EEG recordings. Based on a previous literature review of existing techniques, Second Order Blind Inference (SOBI) has been implemented and tested on real and simulated data as a proof of concept. This follow up project consisted of four tasks:

1. Cleaning and documenting the implemented algorithm
2. Optimizing the calculations
3. Optimizing and validating relevant parameters
4. Combining and documenting the results of point 1 – 3

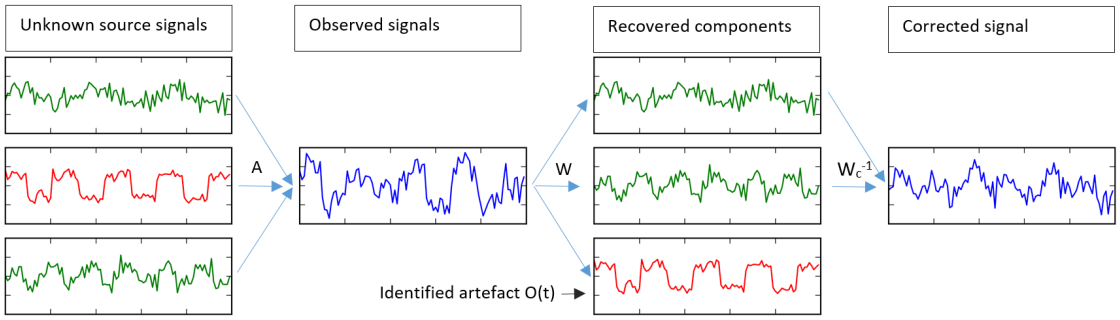

Figure 1: With the (unknown) original sources being true signals in the matrix B (green) and the artifact in O (red), X is a linear combination $X = A[B + O]$ with mixing matrix A. The source signals can be recovered from X with the unmixing matrix W with blind source separation. When the original sources are known, the artifact O can be selected and the new mixing matrix W_c^{-1} is defined with rows corresponding to the artifact set to zero. The corrected signal matrix C is constructed from the sources using W_c^{-1}.

2 Method

Blind source separation (BSS) methods are used in signal processing to recover the independent sources from observations of a linear mixture of the sources [8]. BSS use the observations $X(t)$ to generate the unmixing matrix W, which can be used to obtain the independent component signals. With the original sources being the matrix representations of the true signal B and the artifact O, X is a linear combination $X = A[B + O]$ with mixing matrix A. Given the observation matrix X, the task is to estimate both the mixing matrix A and the original sources $B + O$. Once the mixing matrix A is known, sources can be recovered from X with the unmixing matrix $W = A^{-1}$ with $B + O = WX$. With the original sources of $X(t)$ having been identified, they can be selected and removed, and the signal is reconstructed without the artifacts to produce the corrected signal $C(t)$ (see figure 1).

BSS techniques generally require more electrodes than expected signal sources, and more time points than the square of the number of electrodes. Either the length of the recording or the sampling frequency can be increased to satisfy this requirement. BSS techniques fall in the class of unsupervised machine learning and do not need training data sets. In principle, the methods do not automatically identify artifact components, but the analysis can be automated with the use reference channels.

2.1 Second order blind inference

Second order blind inference (SOBI) uses decorrelation across several time points as its main computational step [2]. SOBI considers the relationship between components at different time lags and insists that these are decorrelated as much as possible. First the data is pre-whitened and a set of cross-correlation matrices are calculated at different delays. Using the joint diagonalizer of the cross-correlation matrices, the sources can be recovered. SOBIs ability to resolve correlated activity is essential for ocular artifact detection since artifact signals coming from the two eyes are highly correlated [14].

The matrix of cross-correlations of the measured (whitened) signal $X(t)$ at time-lag τ is defined as

$$R(\tau) = \mathbb{E}[X(t)X(t-\tau)] = \int_{-\infty}^{\infty} X(t)\overline{X}(t-\tau)\,\mathrm{d}t \tag{1}$$

where $\overline{X}$ represents the complex conjugate. The unmixing matrix W is then computed as the

matrix that jointly diagonalizes (rotates) a set of p whitened cross-correlation matrices

$$\{R(\tau_i)|i = 1, ..., p\} \tag{2}$$

The diagonalization R_W of a matrix R by matrix W is defined as

$$R_W = WRW^{-1} \tag{3}$$

and is equivalent to the eigen-decomposition of R, where W is the matrix containing the eigenvectors and R_W the diagonal matrix containing the eigenvalues. For a set of matrices, the joint diagonalizer is therefore a form of an average eigenstructure. Computation time for calculating the joint diagonalizer is directly related to the number p of cross-correlation matrices.

The advantage of SOBI is that the time-delay of artifacts propagating over the scalp is considered in the identification of the components. SOBI is able to identify correlated activity over time if the right time-lags are chosen for computing the cross-correlation matrices. Determining the right lags is related to the expected delay of propagating signals from the front to the back of the scalp. Studies on effects of the chosen delays suggest a set of lags from 1 to 300 ms works best [11] [9]:

$$\begin{aligned}\tau \in \{&1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65,\\ &70, 75, 80, 85, 90, 95, 100, 120, 140, 160, 180, 200, 220, 240, 260, 280, 300\}\end{aligned}$$

This implies that the signals needs to be sampled at $1kHz$ or higher.

SOBI can be automated [7] by identifying components that correlate with components from EOG channel recordings. First, the EEG/EOG data is decomposed into a number of components equal to the number of sensors. Second, the sign on all lower and horizontal EOG channels is inverted (i.e. multiplied by -1) and the new data is also decomposed into components. The new components that invert compared to the old are flagged. In this step the components corresponding to EOG signals that do not propagate far from the EOG recording site are identified. Third, the components that correlate above a certain level with the lower and horizontal EOG data and components with high power in the low frequency band are flagged as well. This step identifies the components containing larger eye-movements and blinks that propagate far and are not inverted in the second step. The idea is that the components containing eye activity will correlate more strongly with the lower and horizontal EOG channels than with non-ocular components, since these EOG channels reflect primarily ocular motions. The correlation threshold is derived from relations between components originating near the eyes, as identified by geometric relationships contained in mixing matrix W^{-1}. Finally, the flagged components are removed from the data by setting the corresponding rows of the source matrix to zero before multiplying with mixing matrix W^{-1}.

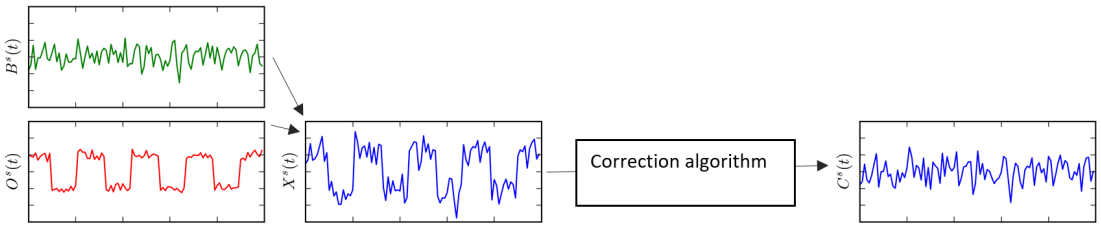

Figure 2: The simulated signal $X^s(t)$ is composed of the simulated 'true' signal $B^s(t)$ and the simulated ocular artifact $O^s(t)$. The corrected signal of the simulated data $C^s(t)$ is obtained by applying an artifact correction algorithm. The improvement in quality of the signal after correction can be directly computed since the original artifact free signal $B^s(t)$ is known.

3 Validation

The greatest challenge for determining the relative efficiency and performance of each artifact correction method is that in 'real' acquired data the uncorrupted desired signal is unknown. The majority of techniques are evaluated with simulated data and thus the validity of conclusions depends on the fidelity of the used model [10]. Methods can first be assessed and compared to other methods by using simulations, but recorded EEG should be used as a final testbed for evaluating the true performance, reliability and reproducibility [12].

3.1 Validation with simulated data

Simulated data can be used for cross-validation of methods, determining optimal values for certain parameters in terms of performance and computation time and as a first guide to validation of corrections. One advantage of using simulated data $X^s(t)$

$$X^s(t) = B^s(t) + O^s(t) \tag{4}$$

where the superscript ^{s} denotes simulated signals and with $B^s(t)$ being the simulated clean signal and $O^s(t)$ being the simulated ocular artifact, as opposed to measured data $X(t)$, is that the quality of the a priori signal and the signal after artifact removal $C^s(t)$ can be assessed directly through performance measures. For M channels and T measurements, each signal is a MxT matrix.

3.1.1 Normalized mean squared error

The normalized mean squared error (NMSE) measures the deviation between the corrected signal $C^s(t)$ and the simulated clean signal $B^s(t)$ [1]. A lower NMSE implies that the corrected signal is more similar to the true artifact free signal. The NMSE is calculated for each channel as

$$NMSE = \left(\frac{\sum_{t=1}^{T} (C^s(t) - B^s(t))^2}{\sum_{t=1}^{T} B^s(t)^2} \right) \tag{5}$$

3.2 Validation with acquired data

In an acquired signal $X(t)$, the true artifact free signal $B(t)$ and the ocular artifact $O(t)$ are unknown. The validation of correction methods on acquired data $X(t)$ depends on a number of factors, of which the most important is the availability of reference channels.

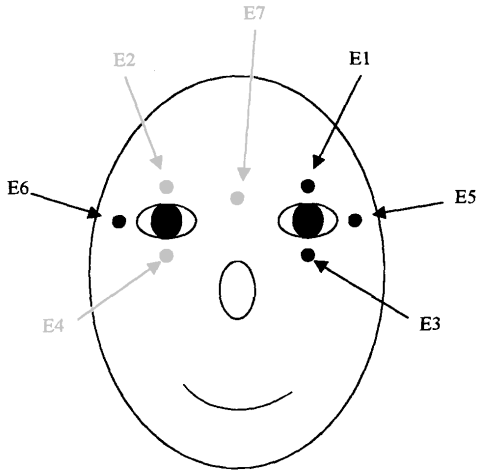

Figure 3: The electrode placement scheme used at the Tilburg symposium [3]. For VEOG the difference between E1 and E3 is computed (E1-E3). For HEOG the difference between E5 and E6 is computed (E5-E6). REOG is computed as (E1+E3)/2.

3.2.1 Regression validation

The electrooculogram (EOG) is measured from electrodes recording voltage changes close to the eyes, see figure 3. For vertical eye-movements, the vertical EOG (VEOG) is computed as the difference between voltages recorded above and below the eye. Horizontal eye-movements are measured as horizontal EOG (HEOG), the difference between voltages at the left and right outer canthi of the eyes. The radial component (REOG) of the eye-movement is measured by subtracting the average voltage at the eyes from the reference electrodes.

Regression validation is based on the assumption that the EOG and EEG channels are relatively uncorrelated [5]. The correlation between the corrected data and the reference EOG channels is determined by employing least-squares linear regressions where corrected EEG is the criterion variable and the H- and VEOG channels the predictor variables. For each electrode:

$$C(t) = X(t) - \beta_1 * VEOG - \beta_2 * HEOG \tag{6}$$

The resultant absolute unstandardized beta coefficients can be tested for relations with repeated measures ANOVA.

4 Optimization of calculations

The goal is to have an implementation of the model that requires as few computational resources as possible. Therefore, fast alternative algorithms to perform certain calculations have been researched and implemented.

4.1 Joint Diagonalization Algorithms

Simultaneous diagonalization, either approximate or exact, of a set of matrices has been proposed as a solution to statistical identification problems [4]. The approximate joint diagonalization of a set of real m-square symmetrical matrices

$$R = \{R_0, R_1, ..., R_J\}$$

where R_0 is positive definite, is an essential tool in BSS algorithms. The goal is to find the unmixing matrix W, also called the diagonalizer or separation matrix, such that the matrices WR_jW^T are as diagonal as possible for $j = 0, ..., J$.

SOBI includes a whitening step that ensures that $R_0 = I$. The problem consists of computing the W that minimizes some criterion. Let $\mathit{Off}(R) = \sum_{k \neq l} r_{l,k}^2$, where $r_{l,k}$ is the (l,k)th entry of matrix R. Then the solution for W is obtained by minimizing w.r.t. W the criterion

$$C_{jd}(W, R) = \sum_{j=0}^{J} \mathit{Off}(WR_jW^T) \tag{7}$$

Let $||R||_F^2 = tr(RR^T) = \sum_{l,k} r_{l,k}^2$ denote the squared Frobenius norm of R. Then minimizing (7) is equivalent to minimizing

$$C_{lsw}(W, D, R) = \sum_{j=0}^{J} ||R_j - W^T D_j W||_F^2 \tag{8}$$

w.r.t. W and the set of diagonal matrices

$$D = \{D_0, D_1, ...D_J\}$$

$||W||_F$ denotes the Frobenius norm of matrix W and is defined as

$$||W||_F = \sqrt{\sum_i \sum_j |w_{ij}|^2} \tag{9}$$

The diagonalization algorithms iterate over the solution space until convergence has been reached of the criterion drops below some given error threshold.

4.1.1 Jacobi Angles for Joint Diagonalization

Simultaneous diagonalization using Jacobi Angles is an extension of the Jacobi technique: a joint diagonality criterion is iteratively optimized under plane rotations [4].

Note again that simultaneous diagonalization may be obtained by minimizing equation (7) w.r.t. matrix W. The extended Jacobi technique constructs W as a product of plane rotations globally applied to all matrices in R.

Denote the operation $rotate(k, l, c, s)$ the complex rotation of a matrix R along axis (k, l) by complex angles c and s

$$\begin{aligned} R_k &= c * R_k + s * R_l \\ R_l &= c * R_l - s * R_k \end{aligned} \tag{10}$$

with $c, s \in C$ and $|c|^2 + |s|^2 = 1$. For each (k, l) the complex angles c and s should minimize the objective function

$$O(c, s) = \sum_j Off(rotate(R_j, k, l, c, s) * R * rotate(R_j, k, l, c, s)) \tag{11}$$

For a given pair (k, l) of indices, a $3x3$ real symmetric matrix G is defined as

$$G = Real\left(\sum_j h^H(R_j)h(R_j)\right) \tag{12}$$

$$h(R) = [r_{kk} - r_{ll}, r_{kl} + r_{lk}, i(r_{lk} - r_{kl})] \tag{13}$$

Then, under the constraint $|c|^2 + |s|^2 = 1$, the objective function $O(c, s)$ is minimized at

$$c = \sqrt{\frac{x + r}{2r}} \qquad s = \frac{y - iz}{\sqrt{2r(x + r)}} \qquad r = \sqrt{x^2 + y^2 + z^2} \tag{14}$$

where $(x, y, z)^T$ is the eigenvector associated to the largest eigenvalue of G. The algorithm considers for each pair of indices all matrices, and has complexity $O(Jm^2)$ for each iteration.

4.1.2 Fast Frobenius Algorithm

The Fast Frobenius algorithm is an iterative scheme to approximate the solution that minimizes (8). The basic idea is to use the invertibility of the matrix W as a constraint preventing convergence to the zero solution [15]. This invertibility can be enforced by carrying out the update of W in each iteration in multiplicative form as

$$W_{(n+1)} = (I + V_{(n)})W_{(n)} \tag{15}$$

where I denotes the identity matrix, the update matrix V is constrained to have zeros on the main diagonal, and n is the iteration number. The off-diagonal components of V are chosen to minimize the C_{lsw} criterion. To maintain invertibility of W is suffices to ensure invertibility of $(I + V_{(n)})$. This is done using the Levi-Desplanques theorem:

Theorem 1 *If an NxN matrix A is strictly diagonally dominant, then it is invertible.*

An NxN matrix A is strictly diagonally dominant if

$$|a_{ii}| > \sum_{j \neq i} |a_{ij} \qquad \forall i = 1, .., N \tag{16}$$

Note that the diagonal terms of $(I + V_{(n)})$ are equal to one. Therefore, is suffices to ensure that

$$\max_i \sum_{j \neq i} |v_{ij}| = ||V_{(n)}||_\infty < 1 \quad \text{or} \quad V_{(n)} = \frac{\theta}{||V_{(n)}||_F} V_{(n)} \tag{17}$$

For some θ ¡ 1. The entries for matrix $V_{(n)}$ are computed as

$$v_{ij} = \frac{z_{ij}y_{ji} - z_{ii}y_{ij}}{z_{jj}z_{ii} - z_{ij}^2} \quad \text{and} \quad v_{ji} = \frac{z_{ij}y_{ij} - z_{jj}y_{ji}}{z_{jj}z_{ii} - z_{ij}^2} \tag{18}$$

$$z_{ij} = \sum_k^J D_i^k D_j^k \tag{19}$$

$$y_{ij} = \sum_k^J D_j^k \frac{E_{ij}^k + E_{ji}^k}{2} \tag{20}$$

where D^k and E^k contain the diagonal and non-diagonal entries of matrix R_k respectively. The algorithm has complexity $O(Jm^2)$ per iteration.

4.1.3 ACDC Algorithm

Minimizing the criterion C_{lsw} in equation (8) w.r.t W is equivalent to minimizing the following criterion w.r.t mixing matrix $A = W^{-1}$ [13]:

$$C_{lsa}(A, D, R) = \sum_{j=0}^{J} ||R_j - AD_jA^T||_F^2 \tag{21}$$

The ACDC algorithm alternates between the following minimization schemes:

1. Minimization w.r.t D: the minimization of (21) w.r.t D for fixed A is given by:

$$\hat{D}_j = Diag\{\left[(A^TA) \odot (A^TA)\right]^{-1} diag(A^TR_jA)\} \tag{22}$$

   where $\odot$ denotes the Hadamard's product.

2. Minimization w.r.t A: the minimization of (21) w.r.t A for fixed D is realized by successive minimizations on the columns $a_1, .., a_m$ of A. For each column a_k of A while other columns are fixed, the minimization is comes from:

$$P_k = \sum_j D_j(k) \left[R_j - \sum_{i \neq k} D_j(i) a_i a_i^T\right] \tag{23}$$

   If v is the unit norm eigenvector associated to the largest eigenvalue λ of P_k, then

$$\hat{a}_k = \frac{\sqrt{\lambda} v}{\sqrt{\sum_j D_j^2(k)}} \tag{24}$$

The ACDC algorithm has complexity $O(Jm^3)$ per iteration.

4.1.4 LSB Algorithm

The least squares solution of (8) w.r.t diagonalizer W (renamed to B) can be rewritten as [6]:

$$C_{lsb}(B, D, R) = \sum_{j=0}^{J} ||BR_jB^T - D_j||_F^2 \tag{25}$$

Without constrains on B and D, the criterion is equal to zero for $B = 0$. Therefore, we impose constrictions such that $D_0 = I$ and $B_0 = I$. Minimization w.r.t. D is obtained by:

$$\hat{D}_j = Diag\{diag(BR_jB^T)\} \tag{26}$$

Then D in the criterion C_{lsb} can be eliminated:

$$C_{lsb} = \sum_{j=0}^{J} Off(BR^jB^T) \tag{27}$$

under constraint $Diag\{diag(BR_0B\} = I$. The minimization w.r.t. B is realized by successive minimizations on the rows of B, $b_1, .., b_m$:

$$C_{lsb}(B, R) = \sum_{j=0}^{J} \sum_{L=1}^{m} \sum_{i \neq L}^{m} (b_L^T R_j b_i)^2 \tag{28}$$

The minimum w.r.t. row b_l minimizes

$$Q_l = \sum_j R_j B(l)^T B(l) R_j \tag{29}$$

if R and B are whitened. $B(l)$ is obtained by removing the l-th row from B. The new solution $\hat{b}_l$ is the unit norm eigenvector associated to the smallest eigenvalue of Q_l. The LSB provides a sequence of matrices B_n by changing one row of the current matrix. For each full iteration the complexity is $O(Jm^3)$.

4.2 Joint Diagonalization Computations

Some of the computations in the joint diagonalization can be further optimized by either estimating the solution or solving the closed-form expression directly.

4.2.1 Power Iteration

The ACDC algorithm requires the computation of the largest eigenvalue and its associated eigenvector of some matrix A. Usually, computing the eigensystem is done by eigenvalue decomposition, which includes solving the characteristic polynomial and provides every eigenvalue and eigenvector of the matrix. This allows the largest eigenvalue and its eigenvector to be selected.

In stead of computing the complete eigenvalue decomposition, it can be faster (for large matrices) to perform the power iteration algorithm to approximate only the eigenvector corresponding to the largest eigenvalue. Starting from a random vector v_0, repeat until convergence:

$$v_{i+1} = \frac{Av_k}{||Av_k||} \tag{30}$$

If A has a unique largest eigenvalue and the starting vector has a non-zero component in the direction of the largest eigenvector, then a subsequence v_i converges to the eigenvector associated with the largest eigenvalue.

4.2.2 Rayleigh Quotient

For a given real symmetric matrix A and non-zero vector v, the Rayleigh quotient is defined as

$$RQ(A, v) = \frac{v^T Av}{v^T v} \tag{31}$$

The Rayleigh quotient reaches it maximum value $RQ(A, v) = \lambda_{max}$, the largest eigenvalue of A, when v is the corresponding eigenvector.

4.2.3 Einstein Summation Convention

The Einstein summation convention is a notational convention that implies the summation over indexed terms in a formula. Not only does this achieve notational brevity, but it can be used to combine multiple steps of computation into one.

Since the introduction of the function einsum in the Python package numpy up untill at least numpy version 1.14.0, einsum provides some advantages in memory accessing that allow some computations, like computing the outer product, to be faster than using numpy's built in function. In the joint diagonalization computations, the outer product of two vectors is computed using einsum.

4.3 Cross-correlation

The matrix of cross-correlations of the measured (whitened) signal $X(t)$ at time-lag τ is defined as

$$R(\tau) = \mathbb{E}[X(t)X(t-\tau)] = \int_{-\infty}^{\infty} X(t)\overline{X}(t-\tau)\,\mathrm{d}t \tag{32}$$

4.3.1 Data Whitening

Whitening of data X is performed by finding the transformation matrix U such that $X_{white} = UX$ results in X_{white} having unit diagonal co-variance. A commonly used choice for U is the eigen-system of the co-variance matrix associated with X. The eigenvectors in U rotate X along an axis that ensures the unit co-variance. Alternatively, by directly calculating the singular value decomposition, a factorization of X, we obtain

$$U\Sigma V = X \tag{33}$$

The columns of U and V are the left-singular values and right singular vectors of X respectively. The left singular vectors of X are a set of orthonormal eigenvectors of XX^T, and the right singular vectors are a set of orthonormal eigenvectors of X^TX. Since U and V are both orthonormal, the whitened matrix can be obtained by

$$UV = X_{white} \tag{34}$$

4.3.2 Fourier Transform Convolution

The cross-correlation over all lags (from -nsamples to +nsamples) can be computed using the Fourier transform convolution. For two signals x_1 and x_2 the cross-correlation is defined as

$$xcorr(\tau) = \sum_{t=0}^{|x1|-1} x_1(t)x_2^*(t - \tau + |x_1| - 1) \tag{35}$$

where $x_2(t) = 0$ when t is outside the range of x_2. This is equal to computing the convolution of the two signals as $(x_1 * x_2)(\tau - |x_1| + 1)$.

Fourier transform convolve is generally very efficient for computing cross-correlations. However, when the signal is very long and only a few lags are needed, a lot of unnecessary computations are done.

4.3.3 Calculating Cross-correlation Directly

The matrix of cross-correlations $R(\tau)$ of the whitened signal $X(t)$ can be computed directly for each τ. The signal is first shifted by the current τ, and then the expected value of the matrix product is computed. Instead of computing the last two steps separately, these steps are combined into one computation using Einstein notation. When only a few lags are needed, this proves to be the faster method for computing cross-correlation.

5 Optimization of parameters

The goal is to optimize the parameters required by the model such that there is a good trade-off between output-quality and calculation speed. Cross-validation on the (simulated) training data set is performed for all of the described parameters.

5.1 Diagonalization Parameters

5.1.1 Algorithm

The four different algorithms described in section 4.1 have been implemented in Python 3.4 using the numpy and scipy libraries, and with optimization described in 4.2 when applicable. To check whether different algorithms result in different results, the following set of algorithms have been used as parameter:

$$\text{Algorithms} = \{\text{Jacobi, Frobenius, ACDC, LSB}\}$$

5.1.2 Error Threshold

The diagonalization algorithms minimize a certain criterion, usually described in terms of the 'off-diagonal error' of the input matrices. Setting the threshold to lower values results in a more strict diagonalization. To test how the threshold affects the final result, the following set of error thresholds have been used as parameter:

$$\epsilon = \{0.1, 0.01, 0.001, 0.0001\}$$

5.2 SOBI Parameters

5.2.1 Set of lags

SOBI relies on second-order statistics based on the joint diagonalization of a set of correlation matrices that are calculated at different lags. Different applications require different sets of lags. Sutherland et. al. reviewed different sets [11] and suggest using a set of 42 lags $\tau = \{1, ..., 10, 12, 14, ..., 25, 30, ..., 100, 120, ..., 300\}$ for the separation of signals originating in the primary somatosensory cortex. Here, we review the original sets of lags again in their performance in EOG artifact removal. The following sets are reviewed:

$$\begin{aligned}
\tau_{std} &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, \\
&\quad 75, 80, 85, 90, 95, 100, 120, 140, 160, 180, 200, 220, 240, 260, 280, 300\} \\
\tau_1 &= \{1, 2\} \\
\tau_2 &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\} \\
\tau_3 &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, \\
&\quad 75, 80, 85, 90, 95, 100\}
\end{aligned}$$

5.2.2 Source-EOG Correlation Threshold

After the independent sources have been identified and unmixed with the unmixing matrix W, the sources originating from EOG can be located. Each source is compared to the EOG signals and those above a certain correlation level are flagged for elimination. Joyce et. al. [7] found a clear separation of ocular components in the geometric relations contained in the columns of W. The correlation values of sources identified as origination in the vicinity of the eyes were always significantly above 0.3. Hence the 0.3 threshold level was used to flag sources. However, as sampling rates can affect these relationships (Joyce et.al. sampled at 500 Hz), they recommend researchers to verify the corresponding correlation threshold for their own data. We compared the performance of the following correlation levels:

$$\rho = \{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6\}$$

5.2.3 Flagging sources

Besides flagging the sources to be eliminated based on the correlation with EOG channels as discussed in the previous section, Joyce et. al. used another method in identification: after the original sources are identified, SOBI is again performed on the same data - but with the EOG channels inverted. This can cause some of the sources to be inverted as well. The inverted sources are then flagged for elimination. We compared the performance of simply using a correlation threshold to the performance of using a correlation threshold and attempting to invert the EOG related sources, in what is further mentioned as the inverting step.

5.3 Results

5.3.1 Simulated data

For 54 simulated EEG signals, the normalized mean squared error (NMSE) is computed between the true signal and the observed signal, and the true signal and the corrected signal for each combination of parameter settings. The NMSE averaged over all subjects before correction is 20.992 μV. Since the simulated signal contained no lag in the EOG artifact, the lag parameter is omitted from analysis.

Algorithm	ϵ	NMSE (μV)	Runtime (s)
Jacobi	0.1	1.64339	2.24
	0.01	0.9544	3.40
	0.001	0.9494	5.48
	0.0001	0.9489	8.88
Frobenius	0.1	10.4860	0.59
	0.01	10.4860	0.59
	0.001	10.4860	0.59
	0.0001	10.4860	0.59
ACDC	0.1	13.1883	1.74
	0.01	13.1883	1.76
	0.001	13.1883	1.74
	0.0001	13.1883	1.74
LSB	0.1	10.3624	10.80
	0.01	10.3624	10.85
	0.001	10.3624	10.87
	0.0001	10.3624	10.83

Table 1: The sum of the NMSE across all electrodes after correction and the runtime, for each combination of diagonalizing algorithm and error threshold value (ϵ), averaged across subjects.

The Jacobi algorithm for diagonalization results in the lowest NMSE (see Table 1). Using this algorithm, lowering the epsilon value does not improve the NMSE by a lot beyond 0.01 (0.5% decrease from 0.01 to 0.001), while runtime increases fast (61% increase from 0.01 to 0.001). The highlighted row is used to set these parameters for the following analysis, shown in Table 2.

ρ	inversion	NMSE (μV)	Runtime (s)
0.1	True		
	False		
0.15	True		
	False		
0.2	True		
	False		
0.25	True		
	False		
0.3	True		
	False		
0.35	True		
	False		
0.4	True		
	False		
0.45	True		
	False		
0.5	True		
	False		
0.55	True		
	False		
0.6	True		
	False		

Table 2: The sum of the NMSE across all electrodes after correction and the runtime, for each combination of correlation threshold value (ρ) and inverting step setting, averaged across subjects.

5.3.2 Acquired data

For the acquired data of 10 subjects, regression validation is performed before and after correction, for each combination of parameter settings. The the resulting absolute summed β coefficients of the two EOG channels before correction are in Table 3.

The Jacobi algorithm results in the lowest absolute β values. Using this algorithm, the error threshold 0.001 results in the lowest β 6 of the 8 times (see highlighted row Table 4). Using that error threshold, the set τ_1 results in the lowest β for the HEOG, and the set τ_3 results in the lowest β for the VEOG, implying that shorter lags work better for HEOG artifact removal, and longer lags for VEOG removal.

The inverting step as described in 5.1.3 did not result in any difference in β coefficients compared to only using the correlation threshold to flag artifact related sources. However, only using the correlation threshold did result in a shorter runtime. In Table 6 and Table 7, only the results for this parameter setting is shown.

Subject	sum absolute β VEOG	sum absolute β HEOG
1	1.5579	1.8378
2	1.0332	1.6867
3	1.7404	1.1660
4	1.7631	1.5747
5	1.4757	1.0928
6	1.6957	1.2527
7	1.4458	1.1430
8	1.4446	1.3155
9	1.4628	1.1449
10	1.8243	1.2765
Mean	1.5444	1.3491

Table 3: The sum of the absolute β coefficients resulting from regression validation for each of the EOG channels for each subject, and the average across all 10 subjects.

		τ_{std}		τ_1		τ_2		τ_3	
Algorithm	ϵ	β V	β H	β V	β H	β V	β H	β V	β H
Jacobi	0.1	0.1070	0.2915	0.0782	0.2808	0.1195	0.2587	0.1215	0.3304
	0.01	0.0354	0.2558	0.0367	0.1843	0.0365	0.2062	0.0299	0.2987
	0.001	0.0330	0.2553	0.0349	0.1783	0.0369	0.2051	0.0284	0.2929
	0.0001	0.0331	0.2555	0.0349	0.1782	0.0368	0.2051	0.0285	0.2930
Frobenius	0.1	0.4058	0.4815	0.4058	0.4815	0.4058	0.4815	0.4058	0.4815
	0.01	0.4058	0.4815	0.4058	0.4815	0.4058	0.4815	0.4058	0.4815
	0.001	0.4058	0.4815	0.4058	0.4815	0.4058	0.4815	0.4058	0.4815
	0.0001	0.4058	0.4815	0.4058	0.4815	0.4058	0.4815	0.4058	0.4815
ACDC	0.1	0.5045	0.5711	0.4058	0.4815	0.4696	0.6054	0.4988	0.5399
	0.01	0.5045	0.5711	0.5471	0.6069	0.4696	0.6054	0.4988	0.5399
	0.001	0.5045	0.5711	0.5682	0.6239	0.4696	0.6054	0.4988	0.5399
	0.0001	0.5045	0.5711	0.5682	0.6239	0.4696	0.6054	0.4988	0.5399
LSB	0.1	0.4276	0.4535	0.5058	0.4403	0.4356	0.4770	0.5393	0.4974
	0.01	0.4276	0.4535	0.5058	0.4403	0.4356	0.4770	0.5393	0.4974
	0.001	0.4276	0.4535	0.5058	0.4403	0.4356	0.4770	0.5393	0.4974
	0.0001	0.4276	0.4535	0.5058	0.4403	0.4356	0.4770	0.5393	0.4974

Table 4: The sum of the absolute β coefficients resulting from regression validation for each of the EOG channels after correction, for each combination of diagonalizing algorithm, error threshold value (ϵ), and set of lags (τ), averaged across subjects.

		Runtime (s)			
Algorithm	ϵ	τ_{std}	τ_1	τ_2	τ_3
Jacobi	0.1	32.99	10.57	20.96	26.95
	0.01	34.03	10.40	21.00	28.02
	0.001	36.35	10.58	21.28	29.76
	0.0001	37.14	10.54	21.78	31.29
Frobenius	0.1	30.93	10.32	19.92	25.58
	0.01	30.99	10.13	19.87	25.57
	0.001	30.97	10.21	19.85	25.53
	0.0001	30.89	10.29	19.78	25.62
ACDC	0.1	33.24	10.18	22.03	27.96
	0.01	33.41	10.46	22.16	28.26
	0.001	33.40	10.34	22.03	27.94
	0.0001	33.20	10.22	21.95	27.85
LSB	0.1	39.66	16.66	29.26	30.22
	0.01	39.61	16.51	28.76	30.21
	0.001	39.27	16.55	28.81	30.05
	0.0001	39.11	16.81	29.13	30.49

Table 5: The runtime in seconds for each combination of diagonalizing algorithm, error threshold value (ϵ), and set of lags (τ), averaged across subjects.

	τ_{std}		τ_1		τ_2		τ_3	
ρ	β V	β H	β V	β H	β V	β H	β V	β H
0.1	0.0520	0.2038	0.0632	0.2016	0.0402	0.1941	0.0443	0.2064
0.15	0.0539	0.2548	0.0658	0.2351	0.0563	0.2629	0.0458	0.2759
0.2	0.0581	0.3339	0.0690	0.2605	0.0924	0.3095	0.0486	0.3279
0.25	0.0572	0.3612	0.0690	0.2605	0.0924	0.3095	0.0497	0.3517
0.3	0.0570	0.3785	0.0690	0.2605	0.0949	0.3715	0.0516	0.4249
0.35	0.0589	0.4123	0.0693	0.2924	0.0949	0.3715	0.0516	0.4249
0.4	0.0589	0.4123	0.0693	0.2924	0.0949	0.3715	0.0516	0.4249
0.45	0.0589	0.4123	0.0693	0.2924	0.0949	0.3715	0.0516	0.4249
0.5	0.0778	0.4431	0.0693	0.2924	0.0950	0.3702	0.0726	0.4477
0.55	0.0778	0.4431	0.0693	0.2924	0.0950	0.3702	0.0726	0.4477
0.6	0.0778	0.4431	0.0693	0.2924	0.0962	0.4134	0.0726	0.4477

Table 6: The sum of the absolute β coefficients resulting from regression validation for each of the EOG channels after correction without applying the inverting step, for each combination of correlation threshold value (ρ) and set of lags (τ), averaged across subjects.

	Runtime (s)			
ρ	τ_{std}	τ_1	τ_2	τ_3
-	23.99	4.06	11.94	19.97

Table 7: The runtime in seconds without applying the inverting step, for each set of lags (τ), averaged across subjects. Since the ρ parameter does not influence runtime, these parameter values are omitted from this table.

6 Discussion

References

[1] L. Albera, A. Kachenoura, P. Comon, A. Karfoul, F. Wendling, L. Sehnadji, and I. Merlet. ICA-based EEG denoising: a comparative analysis of fifteen methods. *Bulletin of the Polish Academy of Sciences - Technical sciences, Polish Academy of Sciences*, 60(3 Special issue on Data Mining in Bioengineering):407–418, 2012.

[2] A. Belouchrani, K. Abed-meraim, J. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–445, 1997.

[3] C. H M Brunia, J. Mocks, M. C. Van den Berg-Lenssen, M. Coelho, M. G H Coles, T. Elbert, T. Gasser, G. Gratton, E. C. Ifeachor, B. W. Jervis, W. Lutzenberger, L. Sroka, A. W. Van Blokland-Vogelesang, G. Van Driel, J. C. Woestenburg, P. Berg, W. C. McCallum, P. Dinh Tuan, P. V. Pocock, and W. T. Roth. Correcting ocular artifacts in the eeg: A comparison of several methods. *Journal of Psychophysiology*, 3(1):1–50, 1989.

[4] Jean-François Cardoso and Antoine Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl.*, 17(1):161–164, January 1995.

[5] R.J. Croft, J.S. Chandler, R.J. Barry, N.R. Cooper, and A.R. Clarke. EOG correction: a comparison of 4 methods. *Psychophysiology*, 42(1):16–24, 2005.

[6] S. Degerine and E. Kane. A Comparative Study of Approximate Joint Diagonalization Algorithms for Blind Source Separation in Presence of Additive Noise. *IEEE Transactions on Signal Processing*, 55(6):3022–3031, 2007.

[7] C.A. Joyce, I. Gorodnitsky, and M. Kutas. Automatic removal of eye movement and blink artifacts from EEG data using blind component separation. *Psychophysiology*, 41:313–325, 2004.

[8] A. Moreaux and G.D. Iannetti. Across-trial averaging of event-related EEG responses and beyond. *Magnetic Resonance Imaging*, 26:1041–1054, 2008.

[9] M.T. Sutherland, J.Y. Liu, and A.C. Tang. Temporal delays in blind identification of primary sometosensory cortex. *Proceedings of the third international conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004.*, page 4222, 2004.

[10] K. Sweeney, H. Ayaz, T. Ward, M. Izzetogly, S. McLoone, and B. Onalar. A methodology for validating artifact removal techniques for physiological signals. *IEEE transactions on information technology in biomedicine*, 16(5):918–927, 2012.

[11] A.C. Tang, M.T. Sutherland, and C.J. McKinney. Validation of SOBI components from high-density EEG. *NeuroImage*, 25:539–553, 2004.

[12] J.A. Urigüen and B. Garcia-Zapirain. EEG artifact removal – state-of-the-art and guidelines. *Journal of Neural Engineering*, 12(3):23pp, 2015.

[13] A. Yeredor. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Transactions on Signal Processing*, 50(7):1545–1553, 2002.

[14] L. Zhang, J. Kwok, and B.L. Lu. Applications of second order blind identification to high-density EEG-based brain imaging: a review. *ISNN 2010: Advances in Neural Networks*, 2:368–377, 2010.

[15] A. Ziehe, P. Laskov, G. Nolte, and K.R. Muller. A Fast Algorithm for Joint Diagonalization with Non-orthogonal Transformations and its Applications to Blind Source Seperation. *Journal of Machine Learning Research*, (5):777–800, 2004.