Published in final edited form as:

Int J Wirel Inf Netw. 2017 June ; 24(2): 109–123. doi:10.1007/s10776-017-0346-3.

On-line EEG Denoising and Cleaning Using Correlated Sparse Recovery and Active Learning

Manish Gupta, Scott A. Beckett, and Elizabeth B. Klerman

Abstract

We have developed two new methods that use sparse recovery and active learning techniques for near real-time artifact identification and removal in EEG recordings. The first algorithm, called Correlated Sparse Signal Recovery (CSSR) addresses the problem of structured sparse signal recovery when statistical rather than exact properties describing the structure of the signal are appropriate, as in the elimination of eye movement artifacts; such tasks cannot be done efficiently using structured models that assume a common sparsity profile of fixed groups of components. Our algorithm learns structured sparse coefficients in a Bayesian paradigm. Using it, we have successfully identified and subtracted eye movement (structured) artifacts in real EEG recordings resulting in minimal data loss. Our method outperforms ICA and standard sparse recovery algorithms by preserving both spectral and complexity properties of the denoised EEG. Our second method uses a new active selection algorithm that we call Output-based Active Selection (OAS). When applied to the task of detection of EEG epochs containing other non-structured artifacts from an ensemble of detectors, OAS boosts accuracy of the ensemble from 91% to 97.5% with only 10% active labels. Our methods can also be applied to real-time artifact removal in magnetoencephalography (MEG) and blood pressure signals.

1 Introduction

With the advent of non-invasive technology and body area networks, real-time wireless streaming of electroencephalography (EEG) signals (recording electrical activity of the brain) is promising for monitoring physiological states such as alertness [2] and clinical conditions such as seizure onset. Automatic detection of times of decreased performance might be useful in safety sensitive environments, for drivers [3] and for workers during night shifts [4]. While there are several studies of EEG-based systems predicting attention lapses [5–7] or detecting/predicting seizures [8, 9], those analyses assumed that the EEG had been cleaned of artifacts. The dominating presence of artifacts in real EEG data make predictions from these algorithms unreliable [10]. The deleterious impact of artifacts, which are exacerbated in ambulatory environments [11, 12], therefore is a serious obstacle to continuous and real-time monitoring using mobile EEG [13].

State-of-the art techniques for removal of artifacts have several pitfalls [14]. Selection of artifact-free epochs is expensive when done semi-manually, unreliable when done using simplistic filtering or thresholding techniques and can result in up to 90% data loss [15]. Artifact subtraction methods using regression [16], Independent Component Analysis (ICA) [17] or Principal Component Analysis (PCA) [18] require lengthy and high density (32+ channel) recordings or require some manual intervention, thus limiting their applicability for

real-time and non-invasive non-cumbersome monitoring (e.g., using an ergonomic EEG headset). Furthermore, selective subtraction of components can result in corruption of the desired EEG signal.

We propose a methodology for artifact identification and removal that is conducive to automated real-time EEG processing. This methodology includes the application in tandem of *two* new techniques that we have developed. (1) A new sparse coding method, that we call **Correlated Sparse Signal Recovery (CSSR)**, [1] (previously called Correlated Sparse Recovery (CSR)) is applied to *denoising* of raw EEG, i.e. identifying and subtracting *structured* artifacts such as eye movements and ECG artifact, which comprise about 70% of artifacts in wake EEG. (2) A new Active Learning (AL) method, that we call Output-based Active Selection (OAS) is applied to denoised EEG (obtained from the first step) and an ensemble of automated EEG detectors for *cleaning* of EEG, i.e. to detect and remove epochs containing any remaining *non-structured* artifacts.

This paper describes both these techniques and their applications to denoising and cleaning of real 6-channel EEG data from healthy awake individuals. The paper is organized into two parts. In the first part, we discuss the mathematical development of CSSR and its application to denoising of real and synthetic EEG data sets. In the second part, we discuss OAS based AL and its application to cleaning of real EEG data and separation of synthetic Gaussian mixtures.

PART A: CORRELATED SPARSE SIGNAL RECOVERY (CSSR) FOR DENOISING

CSSR was motivated by the observation that artifact signals in the EEG such as eye movements and ECG are the result of electric fields from localized dipole sources in the eye or heart, and are sparse in an appropriate dictionary whereas the more complex cortical EEG is not. Application of existing sparse coding algorithms [19, 20], however, such as orthogonal matching pursuit (OMP) or joint matching pursuit (JMP) and other algorithms that assume *fixed* structure, results in over- or under-fitting of these artifacts and spectral distortion in the denoised signal. In contrast, CSSR models *statistical* structure by incorporating a temporal and spatial correlation that defines artifact structure and extending the standard Sparse Bayesian Learning model (SBL) of [21].

2 CSSR - Mathematical Foundations

2.1 Bayesian Model

We use the paradigm of compressed sensing. Given the measurement vector $y \in \mathbb{R}^N$ we wish to recover a source vector $x \in \mathbb{R}^M$ that is sparse in a known dictionary matrix $H \in \mathbb{R}^{M \times N}$ where $\varepsilon \in \mathbb{R}^N$ is an unknown noise vector:

$$y = Hx + \varepsilon \quad (1)$$

For our purposes, x represents a structured artifact such as an eye blink that we want to recover and then eliminate from the measured EEG signal y . Typically H is overcomplete, in that $M \gg N$ and that the sparsity $K = \|x\|_0 \ll M$. The sparse recovery problem is:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \left\{ \|y - Hx\|_2^2 + \lambda \|x\|_p \right\} \quad (2)$$

where λ is a scalar that controls the relative importance applied to the Euclidean error and sparseness terms, and choices of $P = 0, 1$ are used. From a Bayesian perspective, we have a prior belief that x is sparse in the basis H and the objective is to provide a posterior belief for the coefficients x given the measured data y . A widely used "sparsity promoting" prior on x is the Laplacian prior, $x \sim \exp(-\|x\|_1)$ so that with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_N)$ (the notation $\mathcal{N}(\mu, \Sigma)$ denotes a Normal distribution with mean μ and variance Σ , and $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix), the l_1 problem (2) for $P = 1$ is exactly equivalent to the MAP estimation of x from the posterior density $p(x/y)$ with $\lambda = 2\sigma^2$. To carry out Bayesian inference in closed form a parameterized form of the prior is used instead as in the Sparse Bayesian Learning (SBL) algorithm [21][22].

Motivated by the problem of artifacts we use the following model of *structured sparsity*. Unlike *block sparsity* or *group sparsity* model, as in the Block Sparse Bayesian Learning (BSBL) algorithm ([23, 24]) where the assumption that signals share a fixed common coefficient support set is too restrictive, we assume a prior on the coefficients $x \sim \mathcal{N}(0, C)$ with the symmetric covariance matrix $C \in \mathbb{R}^{M \times M}$ (with $C \succ 0$) having the decomposition

$$C = \Lambda R \Lambda \quad (3)$$

Where $R \in \mathbb{R}^{M \times M}$ is the *correlation matrix* and $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_M)$, λ_k being the standard deviation of component k . This so-called *separation strategy* [25], allows us to express our prior beliefs about the temporal dependence between the sparse components of x via the correlation matrix R and we let Λ be estimated from the data. SBL and BSBL are special cases with additional assumptions of diagonal and block diagonal C , respectively. As in other sparse Bayesian models, if x is sparse then most λ_i are close to zero. In the algorithm below we assume that R is known *a priori*. The problem of sparse recovery then becomes a problem of estimating the parameters σ^2, Λ from given data y for which we use Expectation-Maximization (E-M) as described below.

2.2 E-M Based CSSR Algorithm

By treating x as a hidden variable, we apply the E-M algorithm to estimate σ^2, Λ as follows. From (1) above, the conditional density of y/x is $p(y/x) = \mathcal{N}(y; Hx, \sigma^2 I)$. From the conjugate prior density $x \sim \mathcal{N}(x; 0, \Lambda R \Lambda)$, the posterior density of x can be shown to be $p(x/y) = \mathcal{N}(x; \mu_{x/y}, \Sigma_{x/y})$ where, (Note: here notation $\mathcal{N}(z; \mu, \Sigma)$ refers to a Normal distribution in variable z with mean μ and variance Σ .)

$$\Sigma_{x/y} = \left(\Lambda^{-1} R^{-1} \Lambda^{-1} + \frac{1}{\sigma^2} H^T H \right)^{-1} \quad (4)$$

$$\mu_{x|y} = \frac{1}{\sigma^2} \sum_{x|y} H^T y \quad (5)$$

and the marginal density of y is $p(y) = \mathcal{N}(y; 0, \Sigma_y)$ with

$$\Sigma_y = H \Lambda R \Lambda^T H^T + \sigma^2 I \quad (6)$$

so that the likelihood function $L(y, \sigma^2, \Lambda)$ of y is

$$L(y; \sigma^2, \Lambda) = \log p(y) = \frac{-1}{2} \left(\log |\Sigma_y| + y^T \Sigma_y^{-1} y \right) \quad (7)$$

To estimate Λ , the Q function at the t^{th} iteration for the E-step is

$$\begin{aligned} Q(\Lambda, \Lambda^{(t)}) &= E_{x|y, \Lambda^{(t)}} \log(p(y, x)) \quad (8) \\ &= E_{x|y, \Lambda^{(t)}} \log N(y; Hx, \sigma^2 I) \end{aligned}$$

$$+ E_{x|y, \Lambda^{(t)}} \log N(x; 0, \Lambda R \Lambda) \quad (9)$$

and the M-step is

$$\begin{aligned} \Lambda^{(t+1)} &= \underset{\Lambda}{\operatorname{argmax}} Q(\Lambda, \Lambda^{(t)}) \quad (10) \\ &= \underset{\Lambda}{\operatorname{argmax}} E_{x|y, \Lambda^{(t)}} \log N(x; 0, \Lambda R \Lambda) \\ &= \underset{\Lambda}{\operatorname{argmin}} \left(2 \log |\Lambda| + \operatorname{tr}(\Lambda^{-1} R^{-1} \Lambda^{-1} S^{(t)}) \right) \end{aligned}$$

where

$$S^{(t)} = \sum_{x|y}^{(t)} + \mu_{x|y}^{(t)} \quad (11)$$

and the values $\sum_{x|y}^{(t)}, \mu_{x|y}^{(t)}$ are obtained using the value in $\Lambda^{(t)}$ in (4). For notational convenience, we write $L = \Lambda^{-1}$ and we will write the diagonal matrix L in terms of the vector $I = \operatorname{diag}(L) = [I_1 \ I_2 \ \dots \ I_M]^T$. Using the following identity, where A, B are arbitrary matrices and L is a diagonal matrix

$$\text{tr}(LALB) = l(A \circ B^T)l^T \quad (12)$$

where $\circ$ denotes the Shur matrix product, we can equivalently write the M-step of (10) as

$$l^{(t+1)} = \underset{l}{\text{argmin}} \left(2 \sum_{i=1}^M (\log l_i) + l(R^{-1} \circ S^{(t)})l^T \right) \quad (13)$$

The minimization step in (13) above can be done using gradient descent, since the gradient of the function f above to be minimized can be computed as

$$\nabla f^{(t)} = \text{diag}(-2L^{(t)-1} + 2S^{(t)}L^{(t)}R^{-1}) \quad (14)$$

The E-M steps for estimating σ^2 can be derived similarly. We simplify notation by writing $\lambda = 1/\sigma^2$, and write the E-step for the Q -function at the t^{th} iteration as

$$\begin{aligned} Q(\lambda, \lambda^{(t)}) &= E_{x|y, \lambda^{(t)}} \log p(y, x) \quad (15) \\ &= E_{x|y, \lambda^{(t)}} \log N\left(y; Hx, \frac{1}{\lambda}I\right) \\ &\quad + E_{x|y, \lambda^{(t)}} \log N(x; 0, \lambda R \Lambda) \end{aligned}$$

and the M-step is

$$\begin{aligned} \lambda^{(t+1)} &= \underset{\lambda}{\text{argmax}} Q(\lambda, \lambda^{(t)}) \quad (16) \\ &= \frac{1}{N} \|y - H\mu_{x|y}^{(t)}\|^2 \\ &\quad + \frac{1}{N} \lambda^{(t)} (M - \text{tr}(\Sigma_{x|y}^{(t)} C^{-1})) \end{aligned}$$

where the fact that $H^T H = \lambda(\Sigma_{x|y}^{-1} - C^{-1})$.

The above analysis motivates an iterative algorithm using steps (4), (13) and (16) until convergence is reached. The minimization in (13) is attained using the gradient (14). The expensive computation of $\Sigma_{x|y}^{-1}$ in (4) is made more efficient using the matrix inversion lemma, thereby reducing the complexity of the algorithm from that of inverting an $M \times M$ matrix to that of an $N \times N$ matrix (note $M \gg N$).

In our analysis we assume the sparsity level K is known, and thus the K largest Λ entries are retained. Most of the entries in the estimated Λ (a diagonal matrix) approach zero, and a

thresholding technique could be used instead of assuming a fixed sparsity K . Once iteration is complete, and final estimates $\hat{\lambda}$, $\hat{\Lambda}$ are obtained, the estimate $\hat{x}$ is given by $\hat{x} = \hat{\lambda} \hat{\Sigma}_{x|y} H^T y$ where

$$\hat{\Sigma}_{x|y} = \left(\hat{\Lambda}^{-1} R^{-1} \hat{\Lambda}^{-1} + \frac{1}{\hat{\lambda}} H^T H \right)^{-1} \quad (17)$$

and the corresponding estimate $\hat{y}$ of the recovered signal is then given by

$$\hat{y} = H \hat{x} \quad (18)$$

Since the full posterior of $\hat{x}$ is estimated, the above method allows computation of confidence intervals (error bars) of the estimated of $\hat{y}$. We call the above algorithm *Correlated Sparse Signal Recovery* (CSSR). We have also implemented a greedy (non-Bayesian) version of the algorithm which gives us similar estimates as the Bayesian version described above (details not shown).

The dictionary H above is chosen to be one comprising of *skew Gaussian* elements and their first and second derivatives. This particular dictionary was chosen because an empirically constructed dictionary from sample blink shapes was not adequate to model the variability in blink shapes (see **Results**), and eye blink shapes are not sparse in standard bases (e.g. Gabor, Daubechies). Elements in this dictionary are defined using the skew Gaussian kernel $g(x, \kappa)$ defined as

$$g(x, \kappa) = 2\phi(x)\Phi(\kappa x) \quad (19)$$

where $\phi(x)$ to denote a standard normal $\mathcal{N}(x; 0, 1)$ and $\Phi(x)$ to denote the CDF of a standard normal i.e. $\Phi(x) = \int_{-\infty}^x \phi(\xi) d\xi$. A real-valued *skew Gaussian* (SG) function $f_{\theta}: \mathbb{R} \rightarrow \mathbb{R}$ parameterized by $\theta = (\tau, \alpha, \kappa)$ is then

$$f_{\theta}(x) = C(\theta) g\left(\frac{x - \tau}{\alpha}, \kappa\right) \quad (20)$$

where $C(\theta)$ is a normalizing constant such that $\|f_{\theta}\| = 1$. The first and second derivatives, f'_{θ} and f''_{θ} respectively, are computed using the facts

$$\phi'(x) = -x\phi(x) \quad (21)$$

$$\phi''(x) = (x^2 - 1)\phi(x) \quad (22)$$

The parameters τ , α , κ are the *translation*, *scale* and *skew* parameters, respectively. A chosen set of parameter values then determines elements f_θ , f'_θ and f''_θ in a dictionary H .

3 CSSR - Methods

3.1 Data Sets

3.1.1 Real EEG Recordings Containing Artifacts—Four-channel (Fz, Cz, Pz, Oz) EEG and two-channel EOG (vertical and horizontal electro-oculogram channels, abbreviated (V)EOG and (H)EOG, respectively) recordings of nine healthy awake individuals during Karolin-ska Drowsiness Test (KDT) performance testing [43] of an inpatient protocol were used to validate and test our algorithm for identification and removal of artifacts. 2 sec epochs containing artifacts during these KDT portions of these recordings were identified and marked manually by a Registered Polysomnographic Technician (RPSGT). To test denoising, three of the above recordings were used with the KDT portions from which all epochs containing artifacts other than eye blinks were discarded (e.g., gross movement or loose electrode identified by the RPSGT). A total of approximately 14 minutes of these recordings were used to manually extract eye blink shapes and validate the chosen dictionary, and a total of 1.45 hours were used to test EEG denoising algorithms.

3.1.2 Synthetic EEG Containing Eye Blink Artifacts—A blink artifact signal $y_0(t)$ is simulated by fitting a real 2 s EEG epoch containing artifacts using OMP and three components from an overcomplete dictionary (as described in the next section). An epoch was chosen for this purpose in which spectral distortion (SD, see definition below) after $y_0(t)$ is subtracted is minimal, so that $y_0(t)$ is as close as possible to a real blink artifact signal. A 2 s epoch of manual RPSGT-identified artifact-free EEG $y_n(t)$ is then added to create a measured signal $y(t) = y_0(t) + \lambda y_n(t)$ where λ is adjusted for the desired noise level (Fig 5). By synthesizing an EEG signal this way, we are able to quantify the performance of our algorithm for recovering $y_0(t)$.

3.2 Application of CSSR

14 minutes of epochs containing eye blink artifacts were manually analyzed and the precise beginning and end of each blink artifact were marked, from which a set of parameters for analytic construction of the SG dictionary H were determined. The sparsity level K was also fixed heuristically by running algorithms on this data and comparing spectral distortion (SD, see definition below) for various values of K . For synthetic EEG data, a subdictionary $H_0 \subset H$ consisting of 29 elements was used for comparing recovery using sparse Bayesian learning (SBL) with our algorithm. The elements in the dictionary H correspond to Gaussian elements at a fixed scale with translations at intervals of 8 units, or 8/256 seconds for a sampling rate of 256Hz. Note that to make the example more realistic we constructed y_0 with a much larger dictionary than H (i.e., not all elements used to construct y_0 are contained in our dictionary H). For real EEG data, CSSR (both Bayesian and greedy versions) as well

as SBL, BSBL, OMP and JMP algorithms were applied to recordings from the KDT sections of the data sets described in Section 3.1.1 and using dictionary H . The algorithm is applied to each epoch independently. Epochs containing artifacts were identified using a k-means clustering algorithm on the sparse coefficients of the extracted blinks. The extracted artifacts were then subtracted from the original signals to obtain artifact-free EEG in these epochs. We additionally used ICA on these recordings and subtracted the manually identified eye blink component (per recording) to compare it with the CSSR of EEG denoising.

Two performance measures, *signal denoising ratio* (SDNR) and *spectral distortion* (SD), as defined below, were used to compare the performance of various denoising techniques. SD is a measure of how the denoised portion of signal spectrally compares with the non-noisy portion on a given frequency range. We also evaluated CSSR as a technique for *artifact detection* (i.e. identifying epochs that contain artifacts) by comparing artifactual epochs identified by CSSR with those manually marked by an RPSGT.

$$SDNR = \frac{\|y - \hat{y}\|_2}{\|y\|_2} \quad (23)$$

$$SD(\Omega) = \frac{\int_{\omega \in \Omega} [P_{ra}(\omega) - P_{na}(\omega)]^2 d\omega}{\int_{\omega \in \Omega} [P_{na}(\omega)]^2 d\omega} \quad (24)$$

where y is the multi-channel data for a single epoch, $\hat{y}$ is the extracting the artifact signal (18), $P_{ra}(\omega)$ is the power spectral density of the denoised signal over artifactual epochs and $P_{na}(\omega)$ is the power spectral density of the original signal over non-artifactual epochs. For our purposes, *smaller* values of SDNR and SD indicate better performance. In addition, we compared the multiscale entropy (MSE) [34] over denoised and noiseless portions as an indicator of how much the original complexity of the signal is maintained by the denoiser.

For synthetic data, since we know the actual artifact signal y_0 we directly compared it to the inferred artifact $\hat{y}$ using the percentage mean square error (PMSE) metric:

$$PMSE = \frac{\|y_0 - \hat{y}\|_2}{\|y_0\|_2} \quad (25)$$

4 CSSR - Results

4.1 Dictionary Construction and Validation

Blinks extracted from real EEG recordings show markedly different shapes across epochs and channels (Fig 1). This justifies use of an analytically constructed dictionary, instead of one empirically constructed using real blink waveforms. Atoms of this analytic dictionary ((Fig 3) can be used to sparsely represent the gamut of blinks. This dictionary was validated on all extracted blinks (Fig 2). Variation of SDNR of fitted epochs containing blinks with sparsity level K shows us that $K = 2$ is a reasonable choice (Fig. 4).

4.2 Denoising in Synthetic Data Using CSSR

For the simulated signal, $N = 461$, $M = 29$ and we chose sparsity $K = 3$. The value of K is chosen so that it results in the smallest $PMSE$. Comparison of the recovered signal $\hat{y}$

using SBL and CSSR (with $R = R_1$ as specified below) for two levels of $\lambda = 0.75$ and $\lambda = 0.375$ (Fig 5) shows that SBL fits extraneous components resulting in a much poorer fit. The values of $PMSE$ (percentage mean square error), as defined below, are shown in Table 1.

$$PMSE = \frac{\|y_0 - \hat{y}\|_2}{\|y_0\|_2} \times 100 \quad (26)$$

The chosen smaller dictionary H is only for simplification: in reality we would use a much larger dictionary to get a better fit with both SBL and CSSR. The choice for R_1 in this test was made as follows: without any correlations, SBL matches the noise components to the right of the blink (i.e. later in time) because they are also sparse in our dictionary H . For CSSR, we set R_1 with a positive correlation of 0.9 amongst elements of H corresponding to translation levels shorter than 1.25 seconds. While this choice was made heuristically, it illustrates our hypothesis that specifying *a priori* correlations can favor selection of actual signal components over noisy components that are also sparse in the chosen dictionary.

4.3 Denoising in EEG Recordings Using CSSR

The CSSR algorithm identifies and subtracts blink artifacts from EEG signals correctly as confirmed by an RPGST: a visual snapshot is shown in Fig 6. Our algorithm does not suffer from problems of overfitting as in OMP/SBL, and that of underfitting, as in JMP/BSBL (Fig 9(a)). Compared to ICA, it selectively subtracts the artifact noise without removing useful EEG components (Fig 9(b)). Comparison of power spectrum of the denoised signal epochs with that of artifact-free epochs shows that the CSSR algorithm exhibits the least SD (Fig 9(b)), especially in the 1–3Hz range, which is the spectral band (Delta EEG band) most impacted by the presence of eye blink artifacts. Various denoising metrics are compared in Table 2. A comparison of MSE of the denoised signal with that of the original signal indicates that the complexity of the actual EEG signal is preserved after denoising (Fig 7).

For *detecting* epochs containing blink artifacts, a k-means clustering using only the maximum signal amplitude and largest sparse coefficient shows a clear separation of epochs

with and without blinks (Fig 8). Compared with manual RPSGT-identified artifacts, our had a specificity of 96% and sensitivity of 97%.

PART B: ACTIVE LEARNING USING OUTPUT-BASED ACTIVE SELECTION

(OAS-AL)—Development and application of output-based active selection (OAS) on these data to remove epochs with other (i.e., non-structured) sources of artifacts was motivated by the following considerations. While several algorithms to detect EEG epochs containing artifacts exist, none of them have high accuracy by themselves. An *ensemble* of such detectors can possibly do much better when provided with examples (labels) of epochs containing artifacts. The number of examples needed to train the ensemble can be vastly reduced using Active Learning (AL), a paradigm of machine learning based on the idea that learning can be greatly improved if a learner is allowed to choose the data from which it learns, as opposed to the traditional passive learning paradigm in which examples are presented to the learner (Fig 10). Intuitively, the advantage comes from the assumption that labeling of data is often the most expensive operation in the learning process and thus overall cost can be reduced if the learner is able to request only those labels that improve its learning ability. Current AL methods [27,28] use uncertainty of unclassified inputs to determine which labels should be acquired to re-train the classifier in the hope of improved prediction accuracy. However, uncertain samples can often be noisy, so acquiring labels for them and retraining the classifier can actually make prediction worse. Instead, our novel OAS combines uncertain sample inputs with predicted output to give an informative active measure and improves accuracy in noisy cases. We have applied OAS to detection of EEG epochs containing artifacts from an ensemble of automated detectors.

Algorithm 1

Active Learning

Given: Labeled set $\mathcal{L} = \{x_j, y_j\}_{j=1}^L$, Unlabeled set $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$, Batch size B , Total number of active labels K

```

1 repeat
2    $\theta^* = \text{Learn}(\mathcal{L})$  ; // learn model using current  $\mathcal{L}$ 
3    $(x_{i_1} \dots x_{i_B}) = \text{Select}(\theta^*, \mathcal{U})$  ; // select  $B$ 
    informative samples
4    $(y_{i_1} \dots y_{i_B}) = \text{Oracle}(x_{i_1} \dots x_{i_B})$  ; // query oracle
    for labels
5    $\mathcal{L} = \mathcal{L} \cup \{(x_{i_1}, y_{i_1}) \dots (x_{i_K}, y_{i_K})\}$  ; // update
    labeled set
6    $\mathcal{U} = \mathcal{U} \setminus \{x_{i_1} \dots x_{i_B}\}$  ; // update unlabeled set
7 until  $K$  unique active labels obtained;
8 return  $\theta^*, \mathcal{L}, \mathcal{U}$ ;
```

5 OAS - Theory and Foundations

5.1 Active Learning (AL)

In AL, the learner is allowed to choose the data from which it learns, in contrast with a traditional passive learning paradigm in which examples are presented to the learner (Fig 10). An essential component of an active learner is the process of *active selection*, which determines what samples to actively query. AL is closely related to semi-supervised learning (SSL), in which learner is able to learn from a vast pool of unlabeled data in addition to a small number of labeled examples. AL goes one step further in that it is able to determine which labels need to be acquired. While both SSL and AL address the issue of expensive labeling, SSL makes use of information it can derive from unlabeled instances by selecting instances with least ambiguity or most confidence, and AL explores the unknown aspects of the unlabeled data and selects instances with most ambiguity or least confidence [27, 28]. In other words, while SSL attempts to select instances that are most representative, AL selects instances that are most informative.

A generic active learning algorithm is described in Algorithm 1, and the schematic of a single iteration shown in Fig 11. Here a base learner *Learn* (which can be supervised or semi-supervised) returns a trained model θ^* that is then used by *Select* to present samples to an *Oracle* that in turn provides labels for these selected samples. Most active selection algorithms [27] use *uncertainty* based selection: e.g. if the trained model θ produces a posterior probability $p(y/x)$ of output y given sample x then samples x_i with the posterior probability (or likelihood) are selected for active labeling. Other selection criteria, e.g. margin [29, 30], information density [31] or importance weighting [32] are also possible, but in most cases it is some measure of a classifier's own confidence alone that is used to determine which samples to select for active labeling.

5.2 Bernoulli-Gaussian Mixture Model (BGMM)

A basic building block for OAS is a Bernoulli-Gaussian Mixture Model (BGMM). In the general case, we are given a feature space $\mathbf{X} \subseteq \mathbb{R}^d$ and a binary output space $\mathbf{Y} = \{0, 1\}$. Features in a dataset are represented by the random variable $X \in \mathbf{X}$ and the output (class) by the random variable $Y \in \mathbf{Y}$. The goal is to estimate the joint density p_{XY} from a set of input observations $\{x_j\}_{j=1}^N$ and output values (labels) $y = \{y_j\}_{j=1}^L$. We assume $L \leq N$, and let $U = N - L$. The estimation problem is *unsupervised* if $L = 0$, *supervised* if $L = N$ and *semi-supervised* if $0 < L < N$. Our model also includes augmented input observations, $\{u_i\}_{i=1}^N$ and $\{v_i\}_{i=1}^N$ where $u_j \in \mathbb{R}^K$ and $v_j \in \{0, 1\}^K$. These augmented inputs into the BGMM represent the soft and crisp output of a base classifier that are necessary for our output-based active selection model. We assume that (x_j, y_j, u_j, v_j) are jointly iid, (u_j, v_j) are independent of x_j and that u_j, v_j are only component-wise dependent. A parametric joint density model is assumed as follows. x_j is assumed to have a marginal 2-component Gaussian mixture distribution, i.e. $(x_j, y_j = c) = \mathcal{N}(x_j; \mu_c, \Sigma_c)$ for $c = 0, 1$. The distribution of (u_j, v_j) is determined using the following rules. Here $B(z, \pi)$ denotes a Bernoulli distribution for variable z with success probability π , and u_{jk} represents the k^{th} component of u_j .

$$(u_{jk}|y_j = c) \sim \mathcal{B}(u_{jk}; \pi_{ck}) \quad (27)$$

$$(v_{jk}|y_j = c, u_{jk} = m) \sim \mathcal{N}(v_{jk}; \lambda_{cmk}, \sigma_{cmk}^2) \quad (28)$$

for $c = 0, 1$ and $m = 0, 1$. The above assumptions fully specify the joint distribution $p(x, y, u, v)$ with parameters, for $c = 0, 1$ as α_c (mixture proportions), $\Sigma_c \in \mathbb{R}^{d \times d}$, $\mu_c \in \mathbb{R}^d$, $\pi_c \in [0, 1]$, $\lambda \in \mathbb{R}^{K \times 2}$, $\sigma_c \in \mathbb{R}^{K \times 2}$ where we have written (with slight notation abuse) λ_c as the matrix of elements λ_{cmk} and σ_c as the matrix of elements σ_{cmk} . These parameters are estimated using the Expectation-Maximization (E-M) algorithm, by considering the variables $y = (y_{L+1} \dots y_N)$ as hidden variables. The steps for this estimation are fairly standard [41], and enumerated elsewhere [26].

5.3 OAS Algorithm

In our new OAS based AL method, instead of using a classifier's uncertainty $p_{\theta}(y/x)$ alone for selection, we propose to (i) use the base classifier's uncertainty $p_{\theta}(y/x)$ along with the classifier's outputs y and original input features x as inputs to a meta-classifier (a BGMM model), and then (ii) use the uncertainty from this meta-classifier for selection. OAS and an active learning algorithm using OAS (that we call OAS-AL) are shown in Function Output-based Active Selection and Algorithm 2, and a single iteration shown schematically in Fig 12. Both supervised and semi-supervised version of a base classifier can be used with OAS. The implementation shown is that of an *adaptive* version of OAS, that outputs both confident samples (of size C) and active samples (of size B). The idea is that when there is no significant improvement in confidence, active selection can be switched to random (passive) selection by the iterative algorithm. The corresponding active learning algorithm OAS-AL adaptively change the batch size B and confident output size C via the sigmoidal functions

$$B(t) = N \left(b_0 + b_1 \frac{e^{\beta(t-t_0)}}{1 + e^{\beta(t-t_0)}} \right) \quad (29)$$

$$C(t) = N \left(1 - c_0 - c_1 \frac{e^{-\alpha(t-t_0)}}{1 + e^{-\alpha(t-t_0)}} \right) \quad (30)$$

where t is the iteration number, N is the total number of samples, and $b_0 = 0.008$, $b_1 = 0.012$, $c_0 = 0.12$, $c_1 = 0.43$, and α, β, t_0 are parameters chosen by the algorithm. The constants control rate of convergence, and hence the total number of iterations desired, and were chosen heuristically based upon algorithm accuracy against simulated data sets. The

parameters α, β are provided by the user, and the effect of them on the overall performance of the algorithms is discussed in the results.

Function Output-based Active Selection(OAS)

```

1 Function OAS-A( $\theta^*, \mathcal{L}, \mathcal{U}, B$ )
   Input: Trained classifier  $\theta^*$ , labeled set
            $\mathcal{L} = \{x_j, y_j\}_{j=1}^L$ , unlabeled set
            $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$ , batch size  $B$ , confident
           output size  $C$ 
   Output: Active sample set  $\mathcal{A}$  of size  $B$ ,
           Confident output set  $\mathcal{C}$  of size  $C$ 
2 for each sample  $j$  do
3   |  $(\hat{y}_j, v_j) = \theta^*(x_j);$  // get crisp & soft
   | classifier output
4 end
5 for each labeled sample  $j$  do
6   |  $\hat{y}_j = y_j;$ 
7 end
8  $\theta_{BGM}^* = \text{BGMM}(\{\hat{y}_j\}_{j=1}^N, \{v_j\}_{j=1}^N, \{x_j\}_{j=1}^N);$ 
   // train BGMM
9 for each samples  $j$  do
   | // get output & posterior probability of
   | output
10  |  $(z_j, p(z_j)) = \theta_{BGM}^*(\hat{y}_j, v_j, x_j,)$ 
11 end
12 Pick  $\{i_1 \dots i_B\}$  with smallest  $|p(z_i) - \frac{1}{2}|$ ;
13 Pick  $\{l_1 \dots l_C\}$  with largest  $|p(z_l) - \frac{1}{2}|$ ;
14 return  $\mathcal{A} = \{x_{i_1} \dots x_{i_B}\}$  and
        $\mathcal{C} = \{(x_{l_1}, \hat{y}_{l_1}) \dots (x_{l_C}, \hat{y}_{l_C})\};$ 

```

5.4 Ensemble Classification (EC) Using OAS-AL

The OAS-AL algorithm is now applied to the task of *ensemble classification* (EC)[33], that is, producing a single classification output from the output of multiple binary classifiers applied to the same input data. These binary classifiers are assumed to be non-retrainable, i.e. they always return the same output given an input. Existing methods for ensemble classification such as combination or transformation based [33] apply a rule or meta-classifier to the output of the binary classifiers to produce a single output. Instead, we propose an AL method as follows (Fig 13). The outputs from the ensemble are used as additional inputs to the BGMM. Additionally, our method internally uses an ensemble of re-trainable classifiers that are inputs to the OAS. The choice of the number and types of classifiers comprising this internal ensemble is made by the specific application.

Algorithm 2

Active Learning using OAS (OASAL)

Given: Labeled set $\mathcal{L} = \{x_j, y_j\}_{j=1}^L$, Unlabeled set $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$, Total number of active labels K , Parameters α, β

- 1 Initialize Confident output $\mathcal{C} = \emptyset$, Active Set $\mathcal{A} = \emptyset$;
- 2 $t \leftarrow 0$;
- 3 **repeat**
- 4 $\theta^* = \text{Learn}(\mathcal{L})$; // learn model using current $\mathcal{L}$
- 5 Compute B_t, C_t using α, β and (29) ;
- 6 $(\{x_{i_1} \dots x_{i_B}\}, \{(x_{l_1} \hat{y}_{l_1}) \dots (x_{l_C} \hat{y}_{l_1})\}) = \text{OAS-A}(\theta^*, \mathcal{U}, \mathcal{L}, B_t, C_t)$;
- 7 $\{y_{i_1} \dots y_{i_B}\} = \text{Oracle}(x_{i_1} \dots x_{i_B})$;
- 8 $\mathcal{A} = \mathcal{A} \cup (x_{i_1} \dots x_{i_B})$;
- 9 $\mathcal{C} = \mathcal{C} \cup \{(x_{l_1} \hat{y}_{l_1}) \dots (x_{l_C} \hat{y}_{l_1})\}$;
- 10 $\mathcal{L} = \mathcal{L} \cup \{(x_{i_1} y_{i_1}) \dots (x_{i_K} y_{i_K})\}$;
- 11 $\mathcal{U} = \mathcal{U} \setminus \{x_{i_1} \dots x_{i_B}\}$;
- 12 $t \leftarrow t + 1$;
- 13 **until** $K = |\mathcal{A}|$ or no change in $|\mathcal{C}|$;
- 14 **if** $|\mathcal{A}| < K$ **then**
- 15 $\mathcal{U} = \mathcal{U} \setminus \mathcal{C}$;
- 16 $K = K - |\mathcal{A}|$;
- 17 Continue with Random Selection AL for $\mathcal{L}, \mathcal{U}, K, B$;
- 18 **end**
- 19 **return** $\theta^*, \mathcal{L}, \mathcal{U}, \mathcal{C}$;

6 OAS - Methods

6.1 Data Sets

6.1.1 Real EEG Recordings Containing Artifacts—The same EEG recordings described in Section 3.1.1 were used for testing OAS-AL. A set comprising of 8700 epochs across all subjects (a total of 4.83 hours that contains the 1.45 hours described in Section 3.1.1) was used to test OAS-AL and other EC algorithms for EEG cleaning.

6.1.2 Synthetic Gaussian Mixture based on EEG Feature Sets—A 31-dimensional feature vector was derived for each of the 8700 epochs of the 6-channel EEG data set described in Section 3.1.1. CSSR algorithm was applied with sparsity level $K = 2$ and dictionary H (described in the next section) to a 2 sec epoch $z(t)$ of EEG data, to give coefficients $v_1^{(c)}, v_2^{(c)}$ of a sparse vector $v^{(c)}$ for channel c along with a scalar $q^{(c)}$ representing error in the sparse representation. The 31-dimensional feature vector was then comprised of the following per-channel (c) features: $|v_1^{(c)}|, |v_2^{(c)}|$, mean of $|z^{(c)}(t)|$, mean of $|Hv^{(c)}|$ and $q^{(c)}$;

and the correlation coefficient r between Fz and (V)EOG channels. The gathered feature vectors were normalized per KDT episode. Two multivariate (31-dimensional) normal distributions were then fit separately to artifactual and to non-artifactual epochs. $N_0 = 5800$ samples were randomly sampled from the first distribution and $N_1 = 2900$ from the second. Noise uniformly distributed over the hypercube $[-\eta, \eta]^{31}$, where η is a parameter controlling noise rate, was then added to the sample vectors. A scatter plot of the first two principal components of generated data is shown for $\eta = 0$ and $\eta = 0.3$ in Fig. 14.

6.2 Application of OAS-AL

OAS-AL was applied to both synthetic and real data sets and compared to existing algorithms. For the binary classification task on the synthetic Gaussian mixture (Section 6.1.2), we applied some commonly used AL algorithms and OAS-AL. In all tests, a Gaussian mixture based semi-supervised learner was used as the base classifier (see [42] and discussion in 5.2). The AL algorithms tested include: (i) *Uncertainty based AL (Active-UNC)* [27], where the posterior probability is used as uncertainty measure to determine active samples; (ii) *Importance-weight based AL (Active-IW)* [32], where *importance weights* per sample are used to determine active samples (iii) *Query-by-Bagging (Active-QBB)* [28,35], where a disagreement measure between several parallel classifiers derived from the base classifier is used to determine active samples. Several other existing AL algorithms were also tested but eliminated due to poorer performance (a full discussion is in [26]). For reference, we also compared just using the base classifier where a random fraction of samples are labeled (i.e., *Passive-SSL*).

On the real EEG data, OAS-AL was applied to an EC task for detection of epochs containing artifacts. The six base artifact detectors used for this task were: (i) k-means clustering based upon a feature set comprising values of dominant sparse components computed by CSSR, quality of the sparse fit, and raw signal amplitude; (ii) k-means clustering on first two principal components of above feature set; (iii) k-means clustering on features used by (ii) along with instantaneous correlation value r between Fz and (V)EOG channel; (iv) semi-supervised clustering using manifold learning [36] on features used by (i); (v) thresholding of values and match quality of dominant sparse components computed by CSSR; (vi) thresholding of z-scored differential (Fz-Cz and Pz-Oz) raw signal values. No single artifact detector works best on all epochs or all artifact types; these individual classifiers had accuracy ranging from 80–91%, with fair correlation diversity [37]. We compared performance of OAS-AL for artifact detection with other ensemble classification and active learning algorithms (see Fig. 17 for description). Several other algorithms were also tested but their results are not presented here as they had poorer performance (a full discussion is in [26]).

Accuracy, false positive rate (FPR) and false negative rate (FNR) of classification are metrics used to evaluate performance of a binary classifier. In semi-supervised, supervised or AL tasks, accuracy (or its complement, error rate), as a function of the number of labels, or *label complexity* is important. A passive (supervised or semi-supervised) learner with need $O\left(\frac{1}{\epsilon}\right)$ labels for error rate ϵ . An ideal active learner, on the other hand, has $O\left(\ln \frac{1}{\epsilon}\right)$ label complexity [38]. Another metric used to evaluate a learning algorithm is *noise sensitivity*,

that is, the error rate as a function of noise rate η for a fixed number of labels. It is known that when the noise distribution is not known then accuracy decreases at best linearly with noise [39].

7 OAS - Results

7.1 Separation of Noisy Synthetic Data Using OAS-AL

OAS-AL is able to separate the synthetic Gaussian mixture (Section 6.1.2) with high accuracy - the error rate is less than 2% when using only 10% active labels. Comparison of the label complexity of OAS-AL on this dataset (Fig 15) with other standard active learning algorithms (Active-UNC, Active-IW and Active-QBB) and passive learning shows that OAS-AL has the least error rate for any number of labels, and is closest in attaining the ideal $O\left(\ln\frac{1}{\epsilon}\right)$ complexity for active learning. When agnostic noise η is added, this difference is less pronounced, but overall OAS-AL has the best noise complexity (Fig. 16).

7.2 Removal of Epochs Containing Artifacts in EEG Recordings Using OAS-AL

Using 10% active labels, OAS-AL is able to achieve an accuracy of 97.5% in detection of epochs containing artifacts, with a FPR of 0.5% and FNR of 1%. This is the best amongst several state-of-the-art active and nonactive ensemble classification algorithms (Fig 17). Note that standard combination algorithms such as *behavior knowledge space* (BKS) [40] boost accuracy of the ensemble only up to 92.5% from the individual range of 80–91%. The label complexity of OAS-AL is also best when compared with other active ensemble classification algorithms (Fig 18).

7.3 Optimal Parameters for OAS-AL Algorithm

The parameters α , β , t_0 in (29) were determined empirically using performance of OAS-AL on the Gaussian mixture dataset with a fixed (10%) active label fraction. These parameters make a difference in the algorithm convergence and performance. t_0 determines the total number of iterations and hence may be fixed upfront for practicality. We observed that with $t_0 = 5$ the total number of iterations was between 10 and 25. Using grid search on the accuracy output, the optimal α , β values were determined to be $\alpha = 3.2$, $\beta = 1.2$. With these optimal values, the total number of active samples requested and confident samples output saturate to a steady value. OAS-AL switches to random sampling after there is no change in the number of confident samples outputted. The empirical analysis shows that accuracy does not improve beyond this point (Fig 19), thereby justifying our idea behind the algorithm. The best possible accuracy is achieved when this saturation point is reached exactly when the total number of active samples requested equals the maximum number of active requests allowed.

8 Conclusions and Discussion

We have presented two new methods for EEG artifact removal: (i) for denoising, using CSSR to identify and subtract structured artifacts (and retaining the underlying information in the epochs); and (ii) for cleaning, using OAS-AL to remove epochs containing non-structured artifacts. These methods are computationally simple, require only a few number

of simultaneous EEG channels, and work in real-time and thus are ideal for use in body area wireless network applications.

CSSR, making use of temporal and spatial correlations eliminates structured artifacts and reconstructs cortical EEG with high fidelity in both linear (spectral) and non-linear (entropy) measures. Compared to existing methods of artifact detection our method (i) preserves EEG instead of discarding data in epochs containing artifacts; (ii) permits fully on-line implementation since it is epoch-based, unlike ICA which requires long recordings; (iii) requires only few (4–6) channels of EEG facilitating practicality and non-invasiveness; (iv) is fully automated requiring no manual intervention. We have shown that ICA, even when applied to full length recordings, is outperformed by our epoch-based algorithm for faithfully reconstructing blink artifact-free EEG. Our real-time artifact elimination technique may enable using EEG in ambulatory medical applications for monitoring cognitive states during wake and even detecting sporadic events such as seizure onset.

Our alternative to standard group sparsity models of structured compressed sensing has applicability beyond EEG artifacts. We have shown that the proper use of a correlation in the prior joint probability density of coefficients can improve Bayesian sparse recovery of a noisy signal when both the signal and noise have sparse components in a dictionary; for example, both eye blink and cortical EEG components can be sparse in a dictionary but the temporal and spatial relationship amongst these components can help differentiate them. However, while on the average CSSR does better denoising compared to other structured sparsity algorithms, there are some individual epochs in which JMP/BSBL can perform better than CSSR, particularly when the main deflection from the eye blink is very weak in the frontal channels. A practical implementation may use a combination of JMP/BSBL and CSSR. Further extensions to our current work include application to other artifact types, run-time optimization of the algorithm to be operable in mobile environments, strategies on learning the correlation matrix R from data and applications to learning other (non-artifactual) structural information in EEG such as measures of sleepiness.

We have also introduced a new method (OAS) for selection of active samples that provides a computationally simple alternative to existing active selection algorithms and addresses pitfalls of uncertainty based sampling in noisy (non-separable) data. We implemented our method using a Bernoulli-Gaussian model based EM algorithm and tested active learning using OAS (i.e. OAS-AL) on simulated Gaussian mixtures and ensembles with non-retrainable members. When applied to real EEG artifact data, it boosts the accuracy of epoch classification using an ensemble of automated classifiers from 92% to 97.5% which is not possible with standard AL or EC methods. OAS obviates the problem of sampling bias inherent in uncertainty sampling by incorporating the predicted output of a base learner, and provides the following advantages over other AL algorithms: (i) computational simplicity, which permits real-time implementations (ii) consistently better performance in presence of both agnostic and non-agnostic noise, (iii) ease of adaptation to ensemble learning since it internally uses an ensemble based classifier, and (iv) permitting adaptive implementations that can detect when AL is no longer effective.

Our algorithm was tested on specific data sets, and it is plausible that OAS-AL may not always outperform other active learning algorithms on all data sets. For instance, a simple data set that is a mixture of two Gaussian distribution and no added agnostic noise, uncertainty based AL can do as well as OAS-AL, if not better. At the other extreme, with very high noise levels, OAS-AL may result in poorer classification boundaries compared to uncertainty based AL. In our current testing and implementation, some parameters (such as α , β) were chosen by empirical testing. A future implementation will be able to learn these parameter adaptively perhaps resulting in even higher accuracy. OAS could possibly be also used in conjunction with existing active sampling techniques such as density weighting. In addition, alternate implementations would allow it to be used with non EM based base learners. Future work should also include theoretical analysis and guarantees of our algorithm.

References

1. Gupta, M., Beckett, S., Klernan, E. Proceedings of 2016 10th International Symposium on Medical Information and Communication (ISMICT 2016). IEEE; On-line EEG Denoising Using Correlated Sparse Recovery; p. 123-127.
2. Erwin C. AL.(1973). Psychophysiologic indices of drowsiness Detroit, Mich: International Automotive Engineering Congress.
3. Lin, C-t, Wu, R-c, Liang, S-f, Chao, W-h, Chen, Y-j, Jung, T-p. EEG-based drowsiness estimation for safety driving using independent component analysis. IEEE Transactions on Circuits and Systems I: Regular Papers. Dec; 2005 52(12):2726–2738.
4. Lockley SW, Landrigan CP, Barger LK, Czeisler CA. When policy meets physiology: the challenge of reducing resident work hours. Clinical orthopaedics and related research. 2006; 449:116–127. [PubMed: 16770285]
5. Lal SKL, Craig A, Boord P, Kirkup L, Nguyen H. Development of an algorithm for an EEG-based driver fatigue countermeasure. Journal of Safety Research. 2003; 34(3):321–328. [PubMed: 12963079]
6. Vuckovic A, Radivojevic V, Chen ACN, Popovic D. Automatic recognition of alertness and drowsiness from EEG by an artificial neural network. Medical engineering & physics. Jun; 2002 24(5):349–60. [PubMed: 12052362]
7. Chen, L-l, Zhao, Y., Zhang, J., Zou, J-z. Automatic detection of alertness/drowsiness from physiological signals using wavelet-based nonlinear features and machine learning. Expert Systems with Applications. 2015; 42(21):7344–7355.
8. Sareen S, Sood SK, Kumar S. An Automatic Prediction of Epileptic Seizures Using Cloud Computing and Wireless Sensor Networks. Journal of Medical Systems. 2016
9. Zeng K, Yan J, Wang Y, Sik A, Ouyang G, Li X. Automatic detection of absence seizures with compressive sensing EEG. Neurocomputing. Oct.2016 171:497–502. 2016.
10. Cassani R, Falk T, Fraga F. The effects of automated artifact removal algorithms on electroencephalography-based Alzheimer's disease diagnosis. Frontiers in aging Jan.2014 7(10):e44439.
11. Erkens I, Molina GG. Artifact detection and correction in Neurofeedback and BCI applications. Electronics. 2008
12. Farney RJ, Walker JM, Cloward TV, Shilling KC, Boyle KM, Simons RG. Polysomnography in hospitalized patients using a wireless wide area network. J Clin Sleep Med. 2006; 2(1):28–34. [PubMed: 17557434]
13. Mihajlovic V, Grundlehner B, Vullers R, Penders J. Wearable, wireless EEG solutions in daily life applications: What are we missing? IEEE Journal of Biomedical and Health Informatics. 2015; 19(1):6–21. [PubMed: 25486653]

14. Khatwani P, Tiwari A. A survey on different noise removal techniques of EEG signals. *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*. Jan; 2013 2(2):1091–1095.
15. Jung T-PP, Makeig S, Humphries C, Lee T-WW, McKeown MJ, Iragui V, Sejnowski TJ. Removing electroencephalographic artifacts by blind source separation. *Tech Rep*. Jan 2.2000
16. Semlitsch HV, Anderer P, Schuster P, Presslich O. A solution for reliable and valid reduction of ocular artifacts, applied to the P300 ERP. *Psychophysiology*. 1986; 23(6):695–703. [PubMed: 3823345]
17. Delorme A, Makeig S, Sejnowski T. Automatic artifact rejection for EEG data using high-order statistics and independent component analysis. *International workshop on ICA*. Jan; 2001 7(10): 457–462.
18. Schachinger D, Schindler K, Kluge T. Automatic reduction of artifacts in EEG-signals. 2007 15th International Conference on Digital Signal Processing, DSP 2007. Jan; 2007 7(10):143–146.
19. Baraniuk R. Compressive sensing. 2008 42nd Annual Conference on Information Sciences and Systems. 2008
20. Yu L, Sun H, Barbot JP, Zheng G. Bayesian compressive sensing for cluster structured sparse signals. *Signal Processing*. 2012; 92(1):259–269.
21. Wipf D, Wipf D, Rao B, Rao B. Sparse bayesian learning for basis selection. *IEEE Trans on Signal Processing, Special Issue on Machine Learning Methods in Signal Processing*. 2004; 52(8):2153–2164.
22. Ji S, Xue Y, Carin L. Bayesian compressive sensing. *IEEE Transactions on Signal Processing*. 2008; 56(6):2346–2356.
23. Zhang Z, Rao BD. Sparse Signal Recovery With Temporally Correlated Source Vectors Using Sparse Bayesian Learning. *IEEE Journal of Selected Topics in Signal Processing*. 2011; 5(5):912–926.
24. Zhang Z, Rao BD. Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation. *IEEE Transactions on Signal Processing*. 2013; 61(8):2009–2015.
25. Barnard J, McCulloch R, Meng X-L. Modeling Covariance Matrices in Terms of Standard Deviations and Correlations, With Application To Shrinkage. *Statistica Sinica*. 2000; 10:1281–1311.
26. Gupta, M. Ph.D. dissertation. Harvard University; May, 2016 Complexity Reduction for Near Real-time High Dimensional Filtering and Estimation Applied to Biological Signals.
27. Settles B. Active Learning Literature Survey. *Machine Learning*. 2010; 15(2):201–221.
28. Freund Y, Seung HS, Shamir E, Tishby N. Selective Sampling Using the Query by Committee Algorithm. *Machine Learning*. 1997; 168(1997):133–168.
29. Huang, S-j, Jin, R., Zhou, Z-h. Active Learning by Querying Informative and Representative Examples. *Advances in Neural Information Processing Systems* 23. 2010; 36(10):892–900.
30. Hoi SCH, Jin R, Zhu J, Lyu MR. Semi-supervised SVM batch mode active learning for image retrieval. *Computer Vision and Pattern Recognition, 2008 CVPR 2008 IEEE Conference on*. 2008:1–7.
31. Settles B, Craven M. An analysis of active learning strategies for sequence labeling tasks. *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP 08*, no October. 2008:1070.
32. Beygelzimer A, Dasgupta S, Langford J. Importance Weighted Active Learning. *Proceedings of the 26th Annual International Conference on Machine Learning ICML 09*. 2008; (ii):1–8. abs/0812.4.
33. Zhou ZH. *Ensemble Methods: Foundations and Algorithms*. 2012
34. Costa M, Goldberger AL, Peng C-K. Multiscale entropy analysis of complex physiologic time series. *Physical review letters*. 2002; 89(6):068102. [PubMed: 12190613]
35. Settles B. *Active Learning Morgan and Claypool*. 2012; (1)
36. Cai D, He X. Manifold adaptive experimental design for text categorization. *IEEE Transactions on Knowledge and Data Engineering*. 2012; 24(4):707–719.
37. Polikar R. Ensemble based systems in decision making. *Circuits and Systems Magazine*. 2006:21–45.

38. Hanneke S. Rates of Convergence in Active Learning. *The Annals of Statistics*. 2011; 39(1):333–361.
39. Kääriäinen M. Active Learning in the Non-realizable Case. *Alt*. 2006:63–77.
40. Dietterich T. Ensemble Learning. *Apr*. 2002 2007:1–16. 2007.
41. Bishop, CM. *Pattern Recognition and Machine Learning* (Information Science and Statistics. Springer-Verlag New York, Inc; 2006.
42. Zhu, X., Goldberg, AB., Brachman, R., Dietterich, T. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers; 2009.
43. Kaida K, Takahashi M, Akerstedt T, Nataka A, Otsuka Y, Haratani T, Fukasawa K. Validation of the Karolinska sleepiness scale against performance and EEG variables. *Clin Neurophysiol*. 2006; 117(7):1574–81. [PubMed: 16679057]

Fig. 1. Sample blink shapes showing variability across (a) epochs and (b) two channels (Fz and (V)EOG). Amplitudes are normalized and each panel is 2 sec in duration.

Fig. 2. Validation of dictionary atoms by fitting atom combinations to real blink EEG signals. Each panel is 2 sec in duration.

Fig. 3. Sample atoms in constructed blink dictionary. Our dictionary comprises of 212649 atoms, and, on average three atoms are required to match each blink. Each panel represents a signal epoch 2 sec long.

Fig. 4. Spectral Distortion (SD) vs. Sparsity Level (K) for OMP/SBL and CSSR (a) over 0–4Hz (b) over 0–20Hz, indicating that $K = 2$ is a reasonable choice.

Fig. 5.

Comparison of artifact signal recovery in simulated EEG signal using SBL and CSSR with $R = R_1$ and $\lambda = 0.75$.

Fig. 6. Snapshot of multi-channel EEG recording with original (black) and reconstructed (red) signals after blink artifact elimination using CSSR.

Fig. 7. Multi-scale entropy (MSE) after elimination of artifacts and reconstruction of EEG across all epochs in a single recording using CSSR compared with values from artifact-free epochs.

Fig. 8. Clustering of epochs into ones with and without blinks using estimated sparse coefficients.

Fig. 9.

Comparison of CSSR, OMP, JMP and ICA algorithms for artifact elimination (a) Single epoch showing varying over-fitting of extracted artifact and loss of useful EEG after reconstruction. Some fits are omitted for visual clarity. (b) Spectral distortion across all epochs in single recording after reconstruction.

Fig. 10.

The paradigms of *supervised*, *semi-supervised*, and *active* learning. In (*supervised learning* (solid black arrows)), examples are pre-selected at random to be labeled by an expert that are then used as examples to train a learner. In (*semi-supervised learning* (black dashed arrow)) the unlabeled samples are also used to train a learner. In *active learning* (grey arrows) the learner determines a subset of unlabeled examples to be queried the expert to be used as its training examples.

Fig. 11.

Schematic of a single iteration of a generic selection-based active learning algorithm. $\mathcal{U}$ is the unlabeled set, $\mathcal{L}$ is the labeled set, θ^* is the trained model. The dashed arrow from $\mathcal{U}$ to the Learn module is for a semi-supervised base learner. See algorithm 1

Fig. 12.

Schematic of a single iteration of adaptive active learning algorithms using output-based active selection (OAS) (OAS is shown in the light blue box). $\mathcal{U}$ is the unlabeled set, $\mathcal{L}$ is the labeled set, θ^* is the trained model. *Select* implemented by OAS also returns most confident samples that are saved in $\mathcal{O}$ across iterations to determine when to stop the iteration. BGMM is the E-M based model by OAS. The dashed arrow from $\mathcal{U}$ to the *Learn* module is present in the semi-supervised case only. See algorithm 2.

Fig. 13.

Schematic of a single iteration of ensemble clustering (EC) using OAS-AL. The algorithm is identical to that in Fig 12 with modifications shown in purple. BGMM takes additional ensemble inputs, both the ensemble for which we want to do clustering, as well an internal ensemble of base learners.

Fig. 14.

Scatter plot of the first two principal components of the 31 dimensional synthetic data (two Gaussian mixture) (a) Noise rate $\eta = 0$ (b) Noise rate $\eta = 0.3$

Fig. 15.

Comparison of label complexities (error rate vs. label fraction) for some active algorithms that use the Gaussian mixture data set (Section 6.1.2 with $\eta = 0$. Passive SSL is included as reference.

Fig. 16. Comparison of noise sensitivity for OAS-AL with uncertainty based AL (Active-UNC) and passive learning. Other active algorithms with performance worse than Active-UNC are omitted.

Fig. 17.

Comparison of the some active learning (AL) and ensemble classification (EC) algorithms for the EEG artifact data, when using 10% labels. Algorithms, from bottom up: (i) BKS, behavior knowledge space; (ii) BGMM-EC, unsupervised transformation method based upon BGMM model; (iii) SVM-EC, supervised transformation method based upon support vector machine (SVM); (iv) ML-EC, semi-supervised transformation method based upon manifold learning; (v) SVM-AL, active method based upon disagreement measure from SVM base classifiers; (vi) BGMSS-AL, active method based upon disagreement measure from BGMM semi-supervised base classifiers; (vii) Active-SSL, active method based upon uncertainty measure from semi-supervised classifier; and (viii) OAS-AL, active learning using output-based selection. See [26] for details on these algorithms.

Fig. 18.

Label Complexity of OAS-AL and Active-SSL (top two ensemble classification algorithms, see Fig 17) for task of ensemble clustering (EC) on EEG artifact data. Since the ensemble has 35% disagreement rate, labels beyond 35% are not necessary.

Fig. 19.

(a) Accuracies, and (b) number of manual (active), most confident, least confident and all samples, for iterations of the OAS-AL algorithm on the synthetic Gaussian mixture dataset with $\alpha = 3.2$, $\beta = 1.2$, $t_0 = 5$. At a certain point (iteration 9) there is no increase in the number of confident samples that can be learnt and hence no more active samples are requested. For the optimal set of parameters, this limit of active samples equals the maximum number of active samples that can be requested.

Table 1

PMSE values for Arifact Fitting in Simulated EEG Signal

	$\lambda = 0.75$	$\lambda = 0.375$
SBL	91.60	49.23
CSSR	21.60	14.65

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 2

SDNR and SD Values for Artifact Elimination Algorithms

	SDNR	SD₁₋₃₂ Hz	SD₁₋₄ Hz
CSSR	0.067	0.1489	0.0357
OMP/SBL	0.127	0.1503	0.3160
JMP/BSBL	0.091	0.1181	0.0825
ICA	0.430	0.2954	0.1429