UNIVERSITY OF CALIFORNIA SAN DIEGO

Furthering the Automation of Electroencephalographic Source Analysis

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Luca Pion-Tonachini

Committee in charge:

Ken Kreutz-Delgado, Chair
Vikash Gilja
Tzyy-Ping Jung
Scott Makeig
Piya Pal
Virginia de Sa

2019

The dissertation of Luca Pion-Tonachini is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

DEDICATION

To my mother Tullia Tonachini, my father Paul Pion,
my step-father Chuck Mohr, my step-mother Carla Pion,
my sister Robyn Shelly-Mohr, my sister Gaia Mohr-Tonachini,
my brother Joel Pion, my sister Silke Pion,
my friends Tim Harding and Jason Clegg,
my girlfriend Fara Khaleeli, and all my cats.

EPIGRAPH

For we are like tree trunks in the snow.

In appearance they lie sleekly and a little push should be enough to set them rolling.

No, it can't be done, for they are firmly wedded to the ground.

But see, even that is only appearance.

The Trees

by Franz Kafka

Translated by Willa and Edwin Muir

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my departmental advisor Prof. Ken Kreutz-Delgado and my other advisor Scott Makeig. Ken introduced me to the field of machine learning and, by nurturing an intuitive and geometric understanding of the math involved, shared his love of the field with me. He also connected me with the Swartz Center for Computational Neuroscience (SCCN) and introduced me to its director, Scott Makeig. As my advisor, Scott taught me a great deal about electroencephalographic data processing and analysis. He has been ever patient with my writing and has taught me to look deeper and to remember to ask simple questions. I would like to thank both Ken and Scott for their support through my years of work towards the Ph.D degree.

I would also like to thank many of my friends and colleagues at the SCCN who have provided me with guidance, support, and friendship. Makoto Miyakoshi was my first point of contact for learning how to use EEGLAB and has been immensely helpful whenever I had questions regarding EEG analysis and experimental design. Sheng-Hsiou Hsu, Chi-Yuan Chang, Ramón Martinez-Cancino, and Clement Lee have also been friends and collaborators over many projects and years of work. Thanks also to Arno Delorme and Ramón Martinez-Cancino who continue to maintain EEGLAB, a software package for EEG processing and analysis which played a central role in many of my projects. Robert Buffington also deserves many thanks as he has been endlessly accommodating, specifically in helping to set up all of the necessary infrastructure for the ICLabel project, and also more generally with all matters of technical issues I came across. Thanks to Nima Bigdely-Shamlo and Özgür Yiğit Balkan who guided me and helped on some earlier projects. I would also like to acknowledge the slew of Italian officemates that my advisor Scott Makeig has consistently placed with me. In chronological order: Fiorenzo Artoni, Marco Petilli, and Francesco Marini. We had many fun conversations, both productive and in avoidance of productiveness.

I would like to extend my sincere gratitude to the National Science Foundation and to the Swartz Foundation for their substantial support. I also want to thank my Ph.D. committee for

their time and support. Finally, my sincere thanks to my family and friends for keeping me happy, sane, and motivated through this process.

Chapter 2, in part, is a reprint of the material as it appears in Luca Pion-Tonachini, Scott Makeig, and Ken Kreutz-Delgado. Crowd labeling latent Dirichlet allocation. *Knowledge and Information Systems*, 53(3):749–765, 2017. The dissertation author was the primary investigator of this paper. The authors would like to thank Paul D. Pion, DVM, DipACVIM (Cardiology), for his input on which EEG components contain signals originating from heart activity. This work was supported in part by a gift by the Swartz Foundation (Old Field, NY) and by the National Science Foundation under grant number GRFP DGE-1144086.

Chapter 3, in part, has been submitted for publication of the material as it may appear in Luca Pion-Tonachini, Ken Kreutz-Delgado, and Scott Makeig. ICLabel: An automated electroencephalographic independent component classifier, dataset, and website. *NeuroImage (under review)*, 2019a. The dissertation author was the primary investigator of this paper. The expert-labeled test set was annotated with help from James Desjardins, Agatha Lenartowicz, Thea Radüntz, Lawrence Ward and Elizabeth Blundon, and Matthew Wisniewski. Their contributions are greatly appreciated. Thanks also to Francesco Marini for editorial comments. This work was supported in part by a gift from the Swartz Foundation (Old Field NY), by grants from the National Science Foundation under grant number GRFP DGE-1144086 and the National Institutes of Health under grant number 2R01-NS047293-14A1, and by a contract with Oculus VR, LLC. Nvidia Corporation donated a Tesla K40 GPU through its GPU Grant Program which was used to efficiently train all artificial neural network models.

Chapter 4, in part, has been submitted for publication of the material as it may appear in Chi-Yuan Chang, Sheng-Hsiou Hsu, Luca Pion-Tonachini, and Tzyy-Ping Jung. Evaluation of artifact subspace reconstruction for automatic EEG artifact component removal. *Transactions on Biomedical Engineering (under review)*, 2019. The dissertation author was the tertiary investigator of this paper. The authors gratefully acknowledge the valuable suggestions and comments from

Makoto Miyakoshi and the author of artifact subspace reconstruction, Christian Kothe.

Chapter 5, in part, is a reprint of the material as it appears in Luca Pion-Tonachini, Sheng-Hsiou Hsu, Scott Makeig, Tzyy-Ping Jung, and Gert Cauwenberghs. Real-time EEG source-mapping toolbox (REST): Online ICA and source localization. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 4114–4117. IEEE, 2015. The dissertation author was the primary investigator of this paper. Luca Pion-Tonachini and Sheng-Hsiou Hsu contributes equally to this work. This work was supported in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-10-2-0022 and by the National Science Foundation under grant numbers EFRI-M3C 1137279 and GRFP DGE-1144086.

Chapter 6, in part, is a reprint of the material as it appears in Luca Pion-Tonachini, Sheng-Hsiou Hsu, Chi-Yuan Chang, Tzyy-Ping Jung, and Scott Makeig. Online automatic artifact rejection using the real-time EEG source-mapping toolbox (REST). In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 106–109. IEEE, 2018. The dissertation author was the primary investigator of this paper.

Chapter 7, in part, is a reprint of the material as it appears in Sheng-Hsiou Hsu, Luca Pion-Tonachini, Jason Palmer, Makoto Miyakoshi, Scott Makeig, and Tzyy-Ping Jung. Modeling brain dynamic state changes with adaptive mixture independent component analysis. *NeuroImage*, 183:47–61, 2018. The dissertation author was the secondary investigator of this paper. This work was supported in part by a gift fund (KreutzKamp TMS RES F-2467), the National Science Foundation under grant number IIP-1719130), and the Army Research Laboratory under grant number W911NF-10-2-0022. Dr. Makeig's participation was funded by a grant from the National Institutes of Health under grant number R01 NS047293-13A1 and by a gift from The Swartz Foundation (Old Field, NY).

VITA

2012	Bachelor of Science in Electrical Engineering *magna cum laude*, University of California San Diego
2014	Master of Science in Electrical Engineering, University of California San Diego
2016–2018	Graduate Teaching Assistant, University of California San Diego
2019	Doctor of Philosophy in Electrical Engineering, University of California San Diego

PUBLICATIONS

Luca Pion-Tonachini, Ken Kreutz-Delgado, and Scott Makeig. ICLabel: An automated electroencephalographic independent component classifier, dataset, and website. *NeuroImage (under review)*, 2019a.

Luca Pion-Tonachini, Ken Kreutz-Delgado, and Scott Makeig. The ICLabel Dataset of electroencephalographic (EEG) independent component (IC) features. *Data in Brief (under review)*, 2019b.

Luca Pion-Tonachini, Scott Makeig, and Ken Kreutz-Delgado. Crowd labeling latent Dirichlet allocation. *Knowledge and Information Systems*, 53(3):749–765, 2017.

Sheng-Hsiou Hsu, Luca Pion-Tonachini, Jason Palmer, Makoto Miyakoshi, Scott Makeig, and Tzyy-Ping Jung. Modeling brain dynamic state changes with adaptive mixture independent component analysis. *NeuroImage*, 183:47–61, 2018.

Luca Pion-Tonachini, Sheng-Hsiou Hsu, Scott Makeig, Tzyy-Ping Jung, and Gert Cauwenberghs. Real-time EEG source-mapping toolbox (REST): Online ICA and source localization. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 4114–4117. IEEE, 2015.

Luca Pion-Tonachini, Sheng-Hsiou Hsu, Chi-Yuan Chang, Tzyy-Ping Jung, and Scott Makeig. Online automatic artifact rejection using the real-time EEG source-mapping toolbox (REST). In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 106–109. IEEE, 2018.

Chi-Yuan Chang, Sheng-Hsiou Hsu, Luca Pion-Tonachini, and Tzyy-Ping Jung. Evaluation of artifact subspace reconstruction for automatic EEG artifact component removal. *Transactions on Biomedical Engineering (under review)*, 2019.

Chi-Yuan Chang, Sheng-Hsiou Hsu, Luca Pion-Tonachini, and Tzyy-Ping Jung. Evaluation of artifact subspace reconstruction for automatic EEG artifact removal. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1242–1245. IEEE, 2018.

Sheng-Hsiou Hsu, Luca Pion-Tonachini, Tzyy-Ping Jung, and Gert Cauwenberghs. Tracking non-stationary EEG sources using adaptive online recursive independent component analysis. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 4106–4109. IEEE, 2015.

Becky Su, Shouhei Shirafuji, Tomomichi Oya, Yousuke Ogata, Tetsuro Funato, Natsue Yoshimura, Luca Pion-Tonachini, Scott Makeig, Kazuhiko Seki, and Jun Ota. Source separation and localization of individual superficial forearm extensor muscles using high-density surface electromyography. In *Micro-NanoMechatronics and Human Science (MHS), 2016 International Symposium on*, pages 1–7. IEEE, 2016.

ABSTRACT OF THE DISSERTATION

Furthering the Automation of Electroencephalographic Source Analysis

by

Luca Pion-Tonachini

Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics, and Control)

University of California San Diego, 2019

Ken Kreutz-Delgado, Chair

The electroencephalogram (EEG) provides a non-invasive, minimally restrictive, and relatively low-cost measure of mesoscale brain dynamics with high temporal resolution. Although signals recorded in parallel by multiple, near-adjacent EEG scalp electrode channels are highly correlated and combine signals from many different sources, biological and non-biological, independent component analysis (ICA) has been shown to isolate the various source generator processes underlying those recordings. While ICA-based methods have been seeing more and more use, EEG researchers are hampered by the additional manual intervention necessary for source-resolved analyses. These issues can be largely mitigated through the automation of several stages of EEG source analysis. To this end, we developed and evaluated the ICLabel classifier, an automated independent component classifier trained on a large dataset with crowdsourced labels. The crowdsourced labels were estimated using the novel crowd labeling (CL) algorithm, crowd labeling latent Dirichlet allocation (CL-LDA), developed here. The ICLabel dataset that was used to train the ICLabel classifier was also made public to aid in future development of IC classifiers. We also evaluated artifact subspace reconstruction (ASR), an algorithm for artifact removal which is applicable both offline and in real-time, and aids both channel-level and source-level analyses. These tools are combined in the Real-time EEG Source-mapping toolbox (REST) to showcase the utility and ease of real-time, source-level analyses once the individual components of an EEG analysis pipeline are automated. Finally we evaluate adaptive mixture ICA (AMICA) and explore its utility for automatic EEG segmentation and nonstationary analysis. All of these tools and methods are open-source and freely available online.

Chapter 1

Introduction

1.1 Background

Electroencephalography (EEG) is a non-invasive, functional brain-activity recording modality with high temporal resolution and relatively low cost that has been widely used in the fields of neuroscience [Grummett et al., 2014, Artoni et al., 2017], clinical assessment [Marzbani et al., 2016], and brain-computer interfaces[Fabiani et al., 2004]. Despite these benefits, an unavoidable and potentially confounding issue is that EEG recordings mix activities of more sources than just the participant's brain activity. Each EEG electrode channel collects a linear mixture of all suitably projecting electrical signals, some of them not originating from the cortex or even from other biological sources. The relative proportions of those mixtures depend on the positions and orientations of the signal generators and the electric fields they produce relative to each recording channel, which always records the difference between activity at two or more scalp electrodes. This mixing process applies to brain activity as well. Far-field electrical potentials from regions of locally-coherent cortical field activity will not only reach the closest EEG electrodes, but nearly the whole electrode montage to varying degrees [Delorme et al., 2012, Brazier, 1966]. Independent component analysis (ICA) [Jutten and Herault, 1991, Bell and Sejnowski, 1995, Lee et al., 1999, Palmer et al., 2008], a form of blind source separation (BSS), has been shown to unmix and segregate recorded EEG activity into maximally independent generated signals [Makeig et al., 1996, Jung et al., 1998, 2000b, Delorme et al., 2012]. By assuming that the original, unmixed source signals are spatially stationary and statistically independent of each other, and that the mixing occurs linearly and instantaneously, ICA simultaneously estimates both a set of linear spatial filters that unmix the recorded signals and the source signals that are the products of that linear unmixing. Analyzing EEG data at the level of cortical source dynamics is a complicated problem, but allows for much more biologically plausible, physiologically meaningful, and functionally significant results than treating scalp data channels as if they indexed single brain sources. A physiological interpretation of ICA applied to scalp EEG recordings can be found in Onton et al. [2006] and Delorme et al. [2012]. In short, this research has clarified (1) that functional independence across brain regions should be accompanied by temporal independence of the source EEG activities and (2) linear and instantaneous mixing of source EEG activities is produced by volume conduction and scalp mixing.

A typical multichannel EEG recording contains electrical far-field signals emanating from different regions of the participant's brain where cortical tissue generates synchronous electrical potentials [Malmivuo and Plonsey, 1995]. Further potentials that project onto the scalp arise in the subject's eye; and as the subject rotates their eyes, the spatial patterns of these projection change. Electromyographic (EMG) activity associated with any muscle contractions strong and near enough to the electrodes are also summed into the recorded EEG signals. Even electrocardiographic (ECG) signals originating from the participant's heart can appear in scalp EEG recordings. Entirely non-biological signals such as 50-Hz or 60-Hz oscillations induced by alternating current electrical fixtures such as fluorescent lights may also contribute to the recorded EEG. The electrodes themselves can introduce artifacts into the recorded signals when the electrode-skin interface impedance is large or unstable. All of these electrical fields and signal artifacts are combined to form the instantaneous, linear mixture of signals recorded in each electrode channel. However, the various source signals themselves are largely generated independently and should not have any consistent instantaneous effect upon one another, justifying the use of ICA decomposition.

Though useful, the application of ICA to EEG data introduces two problems: (1) sensitivity to noise and artifacts, which is also a concern when not performing source analysis (i.e. channel-based analyses), and (2) ambiguity of the ICA results. If too many artifacts are present in an EEG recording, or even just a few with extreme amplitudes, the ICA solution found may be unusable or noisy, being then comprised of crudely defined independent components (IC), each summing poorly unmixed source signals. Once the data are decomposed, determining whether the decomposition is satisfactory and, if so, which ICs to analyze, requires more work. These nontrivial issues can be enough to dissuade researchers from performing source analysis on their data and can also slow the speed at which researchers can analyze data.

A solution to these problems is automation. By offloading the work from researchers to algorithms, less time, effort, and attention need to be dedicated to data preparation and can instead be allocated to the analysis and interpretation of results. Furthermore, automation makes the application of source analysis to real-time applications possible, opening the possibilities of source-resolved brain-computer interfaces (BCI) and real-time monitoring. As will be described in the next section, certain stages of common to most EEG source analysis workflows are not adequately automated. Through the development of new tools and the investigation of little-used existing method, we aim to further the automation of EEG source analysis.

1.2 Problem Statement

Although specific EEG source analysis pipelines differ, the steps involved can be summarized as follows:

1. Recording and importing data

2. Preprocessing

3. Transient artifact removal

4. ICA decomposition

5. IC selection

6. Source analysis

Varying levels of automation already exists for some of these steps. For example, (Step 2) preprocessing is already largely automated as broadly-applicable digital filters are easy to design and use and (Step 4) ICA decomposition algorithms are easy to apply to EEG recordings through widely-available computer programs (though it is not necessarily easy to get a good result). While (Step 1) the recording of EEG data still takes manual intervention to setup, the actual duration of data collection can be seen as automatic in that no intervention is necessary unless something goes wrong or the recording long enough for the conductive gel to dry; an issue that is being addressed through improvements to dry electrode technology.

Here, we address the processes listed above which are not yet adequately automated: (Step 3) transient artifact removal, (Step 5) IC selection, and (Step 6) analysis. While we make no claim of unequivocally automating these processes, we do improve upon the state-of-the-art for some and evaluate the effectiveness of promising, but little-studied algorithms. Furthermore, with automated EEG processing methods, applications in real-time become feasible so long as the computational cost is well managed. We consequently developed a number of these methods into a tool that enables their combined application on EEG data in near-real-time. Specifics are given in Section 1.3.

1.3 Contributions

In Chapter 2 we developed a new crowd labeling algorithm which was used to estimate reference labels for a large dataset of EEG IC features. These labels and the dataset were used in the work described in Chapter 3, where we developed the ICLabel project comprised of an EEG IC classifier, a dataset of EEG IC features and crowdsourced reference labels, and a website for collecting more crowdsourced labels. We compared the ICLabel classifier's performance against existing IC classifiers. The ICLabel classifier abates the need for manual intervention by providing EEG researchers with consistent, efficient, and state-of-the-art automated IC classification.

In Chapter 4 we assessed the performance of artifact subspace reconstruction (ASR) as an automated, real-time-capable artifact rejection method for EEG data. Although ASR has existed since 2014, it has not been systematically evaluated prior to this work. In doing so, we expect to promote trust in the method, leading to wider adoption through better understanding of the method.

In Chapter 5 we developed the real-time EEG source-mapping toolbox (REST) which implements a pipeline for real-time EEG source analysis with visualizations of intermediate calculations and the ability to estimate IC source locations in near-real-time. In Chapter 6 we extended REST by incorporating ASR and IC classifiers. We showed that REST could effectively negate many common EEG artifacts in near-real-time. This toolbox furthers the automation of EEG both by demonstrating the effectiveness of pipelines which are currently possible and by making such pipelines easier to implement and understand.

In Chapter 7 we assessed the utility of adaptive mixture ICA (AMICA) as an unsupervised brain-state monitoring technique on both simulated EEG data and actual EEG recordings during sleep. AMICA has existed since 2006 but, similar to ASR, it has seen limited adoption due to methodological complexity and computational cost. By demonstrating the utility and of the method and exploring the effects of its parameters, we hope to encourage wider usage when applicable.

In Chapter 8 we conclude with a summary of the work completed and a deeper discussion of how each of the previous chapters further the automation of EEG source analysis.

Chapter 2

Dataset Creation

2.1 Introduction

One of the central requirements to automating electroencephalographic (EEG) source analysis is the ability to parse EEG datasets decomposed using independent component analysis (ICA). We approach that problem in Chapter 3 by developing an automated independent component (IC) classifier. The classifier is trained using the ICLabel dataset: a collection of features extracted from over 200,000 ICs, a subset of which also have crowdsourced independent component labels. In this chapter we develop and assess a novel crowd labeling (CL) algorithm with which to process the crowdsourced IC label suggestions. As a result of applying this CL algorithm, we are able to estimate reference labels for those ICs with crowdsourced labels; without which we could not train an IC classifier.

2.2 Background

Crowd labeling (CL), also referred to as crowd-consensus, is a form of crowdsourcing with the purpose of labeling or categorizing the elements of a provided set of items [Muhammadi et al., 2013]. Examples of such problems include identifying types of food from pictures and rating the emotion most representative of a sentence. Generating a label for a single example is typically easy and takes anywhere from a second to a minute depending on the task. However, with the large, unlabeled datasets that are so common nowadays, the number of labels needed is often far larger than any person has the time or inclination to produce. In such cases, crowdsourcing can be an effective solution as it greatly reduces the time required through sharing and parallelization of labor among volunteers or paid workers. Unfortunately, skill levels within the pool of workers are typically unknown beforehand. When monetary incentives are provided on a per-task basis, there may even be malicious workers who assign labels at random. This results in a collection of labels that are not entirely reliable, meaning that some labels would likely not match the opinion of a domain expert. CL algorithms exist specifically to estimate a label for each question, object, or feature (henceforth called an instance) that is more reliable than the worker inputs from which that label is generated.

The simplest crowdlabeling strategy, selecting the instance category by majority vote, assigns the most commonly submitted label for each instance. In many situations, this method is good enough. Given sufficient votes, low error rates, and lack of consistent bias among workers, the law of large numbers guarantees the average will be reliable. More complex algorithms learn a set of parameters to describe the skill or biases of each worker [Muhammadi et al., 2013]. In the simplest case, the problem can be recast as learning a weighted average over worker submissions. From this perspective, the majority vote can be described as an averaging method assigning equal weight to all workers. Even more complex models can also learn parameters to describe the difficulty of each instance to account for disagreement between otherwise reliable workers. These algorithms can then be applied to binary classification, multi-class classification, multiple-choice classification (where classes vary for each question or task), or even to more free-form paradigms in which workers are allowed to respond with unique, self-generated responses. Though many such algorithms already exist, they largely share the assumptions that each instance pertains to a single class, that responses relate to a single class, and that workers provide at most one response per instance.

After first formally posing the problem of CL, we introduce crowd labeling latent Dirichlet allocation (CL-LDA), an algorithm that generalizes the family of multi-class classification CL algorithms in four important ways:

1. Instance classes are viewed as compositional rather than categorical.
2. Workers may respond with any number of guessed possibilities if they cannot distinguish an obvious correct answer.
3. Responses that do not directly correspond to a class, and may have a different assumed meaning for each worker, are allowed as response options.
4. Prior information on workers can be incorporated in a structured and Bayesian manner.

While latent Dirichlet allocation (LDA) [Blei et al., 2003] is a well known algorithm that has been applied to a wide variety of problems, including something approaching crowd labeling [Agarwal and Chen, 2010], as far as we are aware it has not been generalized in a way that renders it applicable to general CL problems. We provide a generalization that allows CL-LDA to be used in all the above applications of CL excluding multiple choice classification. For simplicity, we restrain our analysis to binary and multi-class datasets. We then provide a comparison to prior CL methods. The notation used in the chapter is presented in Table 2.1. Vector variables are distinguished by boldface and matrix variables by underlined boldface. An additional subscript on vector variables indicates indexing over the scalar elements of the vector.

2.3 Problem Description

The CL problem begins with the assumption that for each instance in a dataset, there is a true class label. CL algorithms then attempt to estimate the class label for each instance in a dataset. Estimates are made using the responses (henceforth referred to as votes) to those instances provided by a set of workers. A CL algorithm accomplishes this by comparing the votes provided by all workers, considering which workers agree or disagree with other workers and on which instances, thereby determining which label is most probable for each instance. More rigorously, CL requires a set of workers $\mathscr{U}$ indexed $u \in \{1,\ldots,U\}$ who consider a set of instances $\mathscr{D}$ indexed $d \in \{1,\ldots,D\}$ producing a set of votes $\mathscr{V} = \{v_{di} \in \{1,\ldots,R\}| \quad d \in \mathscr{D}, \quad i \in \{1,\ldots,N_d\}\}$ where U is the number of workers, D is the number of instances, R is the number of possible responses, and N_d is the number of votes on instance d as can be seen in Table 2.1. For each instance there is assumed to exist an unknown, true class vector $\mathbf{y}_d \in \mathscr{Y}$ that relates the instance to C possible distinct classes. The goal of CL, provided this information, is to generate an estimate $\vec{\theta}_d \in \mathbf{\Theta}$ as close to $\vec{y}_d$ as possible for each instance.

In previous methods, $\vec{y}_d$ was assumed to be a discrete value indicating the true class. To accommodate generalization 1 for compositional data, it is assumed that $\vec{y}_d \in \mathbb{S}^{C-1}$ where $\mathbb{S}^n$ is the n-dimensional probability simplex in $\mathbb{R}^{n+1}$. This means that $\sum_{l=1}^{C} y_{dl} = 1$; i.e., the elements of $\vec{y}_d$ sum to one. The assumption of previous methods that $\vec{y}_d$ is discrete is a special case solution under this generalization, wherein $\vec{y}_d$ is a binary indicator vector with only the element related to the true class being equal to one and all others zero.

2.4 Latent Dirichlet Allocation

LDA is a generative hierarchical Bayesian mixture model for unsupervised data clustering based on unobserved similarities or themes common throughout a dataset [Blei et al., 2003]. It is most well known for topic modeling in documents [Blei et al., 2003], but has many other applications such as recommendation systems [Blei et al., 2003, Krestel et al., 2009], object detection in images [Wang and Grimson, 2008], and image annotation [Lienou et al., 2010]. In this section we provide a brief review of LDA for context before presenting the generalization to

Table 2.1: Notation

Symbol	Meaning
$\underline{\boldsymbol{I}}^{N\times N}$	Identity matrix of size N by N
$\underline{\mathbf{1}}^{N\times M}$	Matrix of all ones of size N by M
D	Number of instances in the dataset
C	Number of classes
R	Number of possible responses
N_d	Number of votes submitted on sample d
$\mathcal{V}$	All votes
U	Number of workers contributing votes
$\mathcal{U}$	All workers contributing votes
$\vec{\alpha}$	Prior class distribution in the dataset
$\vec{\beta}_k^u$	Prior vote distribution given class k and worker u
$\underline{\beta}^u$	Prior vote distribution matrix on all classes for worker u
$\vec{\theta}_d$	Class distribution of instance d
$\vec{\phi}_k^u$	Vote distribution given class k and worker u
v_{di}	Value of vote i in document d
z_{di}	Class of vote i in document d
$\mathcal{Z}^{-di}$	All vote-classes excluding that of vote i in document d
m_{du}	Weight of a vote on sample d by worker u
n_{jklu}	Combined weight of votes with value l by worker u that are assigned class k on instance j
n_{jklu}^{-di}	Same as above excluding the weight of the ith vote on instance d

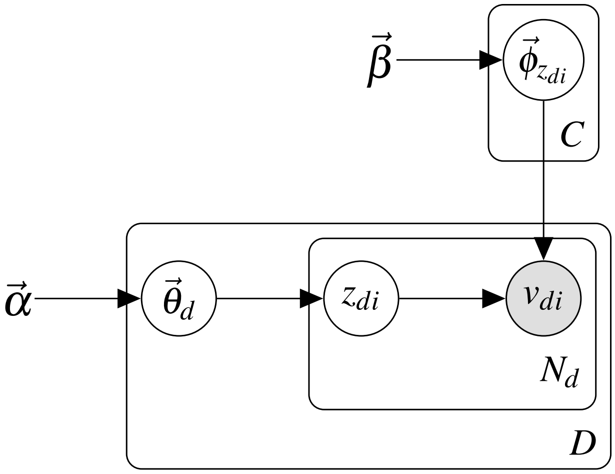


Figure 2.1: Graphical model for LDA.

CL-LDA. While the model described here is referred to in the original paper [Blei et al., 2003] as smoothed-LDA, the smoothed-LDA model is also commonly called LDA as we do here. For simplicity, we describe LDA from the perspective of document topic modeling to maintain a consistent analogy between the intuitions behind LDA and CL-LDA.

LDA applied to document topic modeling learns a probabilistic generative model for a corpus, $\mathscr{D}$, comprised of D documents. Each document, d, contains N_d words. The probabilistic generation of these documents and words begins with a Dirichlet prior over topics in the corpus with parameter vector $\vec{\alpha}$, from which each document's topic distribution, $\vec{\theta}_d$, is drawn. For each document, N_d samples are taken from a multinomial distribution with parameter vector $\vec{\theta}_d$. These samples are word-topics within the document, denoted as z_{di}, which make explicit the topic of the context in which a word is used. For example, the word "rash" could be a symptom in the context of medical literature, but might also describe a decision in the context of a political commentary. A second Dirichlet prior has parameter vector $\vec{\beta}$ over word distributions given topics from which each topic dependent word distribution, $\vec{\phi}_k$, is sampled. For each word-topic z_{di}, a word is drawn from a multinomial distribution with parameter vector $\vec{\phi}_{z_{di}}$. Effectively, $\vec{\phi}_k$ parametrizes the vocabulary used by topic k. In summary:

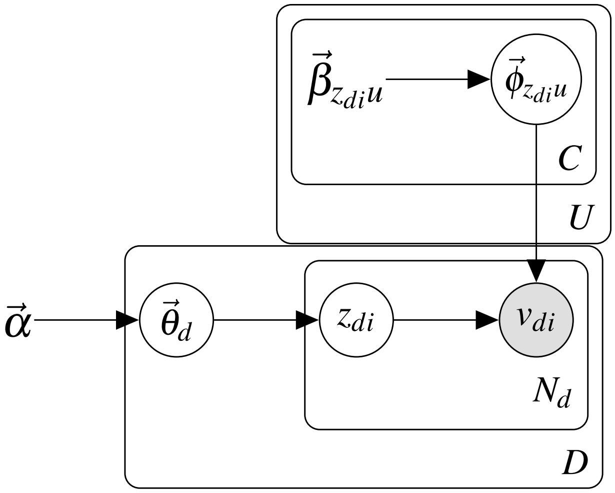


Figure 2.2: Graphical model for crowd labeling latent Dirichlet allocation (CL-LDA).

$$\vec{\theta}_d \sim \text{Dirichlet}(\vec{\alpha}) \quad \forall d \in \mathscr{D} \tag{2.1}$$

$$\vec{\phi}_k \sim \text{Dirichlet}(\vec{\beta}) \quad \forall k \in \{1,\ldots,C\} \tag{2.2}$$

$$z_{di} \sim \text{Multinomial}(\vec{\theta}_d) \quad \forall d \in \mathscr{D}, \forall i \in \{1,\ldots,N_d\} \tag{2.3}$$

$$v_{di} \sim \text{Multinomial}(\vec{\phi}_{z_{di}}) \quad \forall d \in \mathscr{D}, \forall i \in \{1,\ldots,N_d\} \tag{2.4}$$

The only information provided to the model are the words, $\mathscr{V}$, in the corpus and the priors, $\vec{\alpha}$ and $\vec{\beta}$. In the original derivation N_d is treated as a random variable drawn from a Poisson distribution, but as it is independent from the other data generating variables, $\vec{\theta}_d$ and z_{di}, it may be treated as deterministic.

2.5 Crowd Labeling Latent Dirichlet Allocation

Though LDA may not immediately appear applicable to the problem of CL, it can be shown to be analogous given two generalizations. Where before the data were a corpus of documents containing words, for CL we analyze a set of instances on which workers have voted.

There is a clear relation between documents in a corpus and instances from CL in which the words in documents are analogous to votes on those instances. The missing component in the topic modeling paradigm is some complement to the relation between workers and their votes. Such a relationship can be easily added by generalizing the topic-dependent word distribution in classical LDA, $\vec{\phi}_k$, to a class and worker dependent vote distribution, $\vec{\phi}_k^u$. Likewise, the prior over vote distributions, $\vec{\beta}$, is generalized on a per-worker and per-class basis as $\vec{\beta}_k^u$. This generalized prior can be thought of as a per worker prior on confusion matrices. While the possibility exists for a unique prior assigned to each worker, a more apt model would use the worker-dependent priors to describe known populations of workers. Worker-dependent priors need not be different, as the worker and class-dependent vote distributions will still be learned from the data. Therefore that flexibility should only be employed when prior information supports its use.

As a third generalization, vote-classes are given weights, m_{du}, such that each voting worker has equal weight on an instance, independent of how many votes they submit on that instance. This is an adaptation of term weighting schemes [Wilson and Chew, 2010] using a different formula for the value of weights and serves to allow multiple responses within a single vote, CL generalization 2 from Section 2.2, without biasing the result in favor of users who do so more often.

2.5.1 Meaning of vote-classes

When compared to word-topics in LDA, vote-classes in CL-LDA are less intuitive because when a worker submits a vote, that vote often appears obvious in its intention. A distinction can nevertheless be made between equivalent votes, one of which is made explicit though the vote-class latent variables. Suppose there is an instance from a dataset with two possible classes. One worker submits a vote for the first class and a second worker submits a vote for the second class. Assuming the first worker is estimated as trustworthy, the vote-class matches the vote by that user. If the second worker is estimated as inaccurate, as they often misidentify the first class as being the second, then the vote-class will likely still be for the first class and the instance will be estimated as strongly first class. If, instead, the second worker is estimated to be accurate, the second vote-class will follow the vote as being of the second class and the instance will be estimated to be evenly split. In the case that the second worker votes both classes, each vote has a unique vote-class and so can vary the interpretation of this instance from either fully class one due to a misunderstanding of class two on the workers behalf, a mixture of both classes as the worker correctly identified similarities to both, or fully class two due to a misunderstanding of class one. Vote-classes estimate the best interpretation of votes, as they might differ from the obvious intention due to a misunderstanding of response options or misinterpretation of instances by the worker.

2.5.2 Inference on CL-LDA

We use collapsed Gibbs sampling [Griffiths and Steyvers, 2004] to perform inference on the CL-LDA model. For reference, the word-topic probabilities in LDA are calculated as:

$$P(z_{di} = k | \mathscr{Z}^{-di}, \mathscr{V}, \vec{\alpha}, \vec{\beta}) \propto (n^{-di}_{jk*} + \alpha_k) \frac{n^{-di}_{*kv_{di}} + \beta_{v_{di}}}{n^{-di}_{*k*} + \beta_*} \tag{2.5}$$

and the marginalized distribution parameters reconstructed as:

$$\theta_{dk} = \frac{n_{dk*} + \alpha_k}{n_{d**} + \alpha_*} \qquad \phi_{kl} = \frac{n_{*kl} + \beta_l}{n_{*k*} + \beta_*} \tag{2.6}$$

where $\mathscr{Z}^{-di}$ are all the word-topics excluding z_{di} and n_{jkl} is the number of times word l appears in document j with topic k. A $*$ indicates a summation over the index it occupies; e.g. n_{jk*} is the number of words in document j that have topic k.

Accounting for workers and weighting votes to equalize the influence of each worker, the instance-class probabilities in CL-LDA are calculated as:

$$P(z_{di}=k|\mathscr{Z}^{-di},\mathscr{V},\vec{\alpha},\underline{\beta}^{\mathscr{U}}) \propto (n_{d*k*}^{-di}+\alpha_k)\frac{n_{*kv_{di}u}^{-di}+\beta_{kv_{di}}^{u}}{n_{*k*u}^{-di}+\beta_{k*}^{u}} \tag{2.7}$$

and the marginalized distribution parameters reconstructed as:

$$\theta_{dk}=\frac{n_{dk**}+\alpha_k}{n_{d***}+\alpha_*} \qquad \phi_{kl}^{u}=\frac{n_{*klu}+\beta_{kl}^{u}}{n_{*k*u}+\beta_{k*}^{u}} \tag{2.8}$$

Everything is the same as before except that z, w, and n are additionally indexed by workers and $\vec{\beta}$ is additionally indexed by workers and classes. Although not evident from the equation, the counts are also changed to summations of weights, m, such that $n_{dz_{di}v_{di}u}^{-di}=n_{dz_{di}v_{di}u}-m_{du}$ rather than simply subtracting one as before. The result is that computational complexity remains unchanged from LDA.

2.5.3 Effect of priors in CL-LDA

Just as in the original derivation of LDA, priors can be interpreted as pseudo-votes that act to smooth the solution towards prior beliefs. Selection of adequate parameters for the prior distributions is essential for CL-LDA to correctly infer instance classes. This is especially true for each $\vec{\beta}_k^u$. If $\vec{\beta}_k^u$ is set to be uniform, then the class-dependent distributions that CL-LDA finds will be associated to an unknown class and the resulting solution would have to be analyzed to determine the meaning of each “class”, thereby negating the utility and autonomy of the algorithm. By choosing $\vec{\beta}_k^u$ such that each possible vote is favored by the most associated class, the results are guided to a known distribution of classes. If it is assumed a priori that the workers are highly skilled, and given that each vote has a one to one correspondence to a single, unique class, then $\underline{\beta}^u$ is a scaled identity matrix with scaling equivalent to the strength of the assumption of worker competence. Workers are usually not perfect, so it instead makes sense to set $\underline{\beta}^u$ to a positive, linear combination of an identity matrix and a matrix of uniform values. If a possible vote does not have a one-to-one correspondence to a class, then that value can be set however best fits prior assumptions.

The class distribution prior does not require such premeditated structure but is also important. This distribution is entirely analogous to that of LDA and should incorporate any prior knowledge on the distribution of instance classes within the dataset. The smoothing effect of the class distribution prior leads to a concern when choosing the scaling for $\vec{\alpha}$ as there are typically far fewer responses per instance in CL than there are words in a document. Care should therefore be taken that α_* is significantly less than the typical number of worker responses on an instance of the dataset or else the smoothing effect of the α will overpower most votes.

2.5.4 Bayesian prior estimation

Imposing a prior can benefit inference by favoring estimates that are closer to solutions that are believed to be more likely, especially when there is minimal data available. Conversely, if there is a large discrepancy between the belief guiding an imposed prior and the true distribution, that prior can be equally detrimental when there is not enough data to overcome its influence. Bayesian prior estimation (BPE) optimizes the prior distribution so as to maximize the data evidence. Following the analysis by Wallach et. al. of estimating non-symmetric Dirichlet priors in LDA [Wallach et al., 2009], an extension is possible to CL-LDA with Bayesian prior estimation (CL-LDA-BPE). While Wallach suggests the use of her derived estimator [Wallach, 2008], this is not possible with CL-LDA as Wallach's derivation assumes that counts are integer valued, which is not applicable here as a result of vote weighting. Therefore, CL-LDA-BPE uses Minka's fixed-point iteration [Minka, 2000]. It is adapted to CL-LDA-BPE as:

$$\hat{\alpha}_k = \alpha_k \frac{\sum_{d=1}^{D} \Psi(n_{dk**} + \alpha_k) - \Psi(\alpha_k)}{\sum_{d=1}^{D} \Psi(n_{d***} + \alpha_*) - \Psi(\alpha_*)} \tag{2.9}$$

and

$$\hat{\beta}^u_{kl} = \beta^u_{kl} \frac{\Psi(n_{*klu} + \beta^u_{kl}) - \Psi(\beta^u_{kl})}{\Psi(n_{*k*u} + \beta^u_{k*}) - \Psi(\beta^u_{k*})} \tag{2.10}$$

where $\hat{\alpha}_k$ is the updated kth element of $\vec{\alpha}$, $\hat{\beta}^u_{kl}$ is the updated lth element of β^u_k, and $\Psi(\cdot)$ is the Digamma function. BPE is a single example of the extensive literature of LDA modifications that can possibly be adapted to CL-LDA. Other advantageous modifications include parallelizing inference across processors [Wang et al., 2009], parallelizing inference on a graphical processing unit [Yan et al., 2009], and adaptation for online inference when receiving streaming data [Hoffman et al., 2010, Canini et al., 2009].

2.5.5 Similar prior methods

CL-LDA can be interpreted as part of a larger family of CL algorithms based upon the confusion matrix approach of Dawin and Skene (DS) [Dawid and Skene, 1979]. DS models each worker using a confusion matrix across classes and votes with the assumption that each instance is of a particular discrete class. This approach can be, and has been, extended many times. Many extensions make the model Bayesian by incorporating hierarchical prior distributions over the classes, workers, or both. For example, the independent Bayesian Classifier Combination [Kim and Ghahramani, 2012] and its variations [Moreno et al., 2014] extend DS by imposing a Dirichlet prior over the class distribution, Dirichlet priors over the rows of the confusion matrices, and exponential priors on the Dirichlet parameters. Latent Confusion Analysis (LCA) [Sato et al., 2014] also extends this method using confusions matrices by imposing normalized gamma priors on the confusion matrices as well as assuming that voting patterns are structured and shared throughout populations of workers. LCA can also be framed from the perspective of LDA but is changed substantially to incorporate shared voting patterns and latent variables for instance difficulty. As previously described, preserving more similarities to LDA, as CL-LDA does, provides many additional benefits, such as access to a rich literature and the use of efficient collapsed Gibbs sampling for inference.

2.6 Experimental Evaluation

To quantify the performance of CL-LDA when applied to CL problems, we use SQUARE [Sheshadri and Lease, 2013, Sheshadri, 2014]: a toolbox that applies CL algorithms to publicly available datasets as well as simulated datasets under various conditions ranging from unsupervised to fully supervised. As CL-LDA is an unsupervised method, only the unsupervised methods of SQUARE are utilized.

2.6.1 Datasets

To compare CL-LDA against other CL methods, we are required to use data that are compatible with those other methods. As a result, datasets that require CL-LDA's added capabilities cannot be used in these comparisons. An exception is made for multi-class datasets, in which case algorithms that only accept binary classes are excluded. The data in this experiment are therefore in the form of workers voting for a single discrete label per instance.

Found data

All non-simulated CL datasets used in these tests are publicly available on the internet. Each varies in response types, overall number of responses, and distributions of responses across workers and instances. Details of these datasets are shown in Table 2.2. AC2 [Ipeirotis et al., 2010] consists of collected worker ratings on websites ranging from child-friendly to pornographic on a four-point scale. In BM [Mozafari et al., 2012], workers rate the sentiment of tweets as positive or negative. CSv3B is a series of judgments on whether statements are true or false and is a binarized version of CSv3 [Orr, 2013] in which there was a very rare "skip" response accounting for less than 0.15% of all responses. In HC [Buckley et al., 2010, Tang and Lease, 2011], workers rate search results as either not-, somewhat-, or very-relevant while HCB combines the somewhat-relevant and very-relevant ratings into a single response. WVSCM [Whitehill et al., 2009] has workers discriminate images of genuine smiles from images of forced smiles. All datasets are either binary or multi-class except for AC2 which is ordinal.

Simulated data

To compare the resilience of CL-LDA to spammers as compared to other methods, we generate simulated data. The simulated datasets are again made to conform with the requirements of the other algorithms. Every simulated dataset consists of 16,000 instances which pertain to one of seven classes. The prior probabilities of the *i*th class is $i/28$. To explore the effects low-accuracy workers, we model those workers as random spammers who vote uniformly at random over all classes. Smart spammers are modeled as trying to avoid detection while minimizing effort by always voting with the class having the highest prior probability. Such workers might appear when monetary rewards are offered for every instance completed. Here, smart spammers always vote for the seventh class. Each worker votes on 300 separate instances at random within the dataset. Non-spamming workers are modeled purely as an accuracy so as to not explicitly favor any family of methods over any other. Each dataset contains 16 workers with 98% accuracy and 160 with 70% accuracy. For each type of spammer, five additional datasets are created adding 32 to 160 spammers in increments of 32. For each condition, ten datasets are generated with different random seeds to provide a measure the stability on each solution.

2.6.2 Other CL algorithms

Sheshadri et. al. [Sheshadri and Lease, 2013] provide a suitable review of each method compared, which is summarized here. The majority vote (MV) takes the most commonly voted class for instance as the correct answer without any regard for which workers produced those votes. ZenCrowd (ZC) [Demartini et al., 2012] generalizes MV by adding a scalar parameter for

Table 2.2: Benchmark data metrics.

Dataset	Classes	Instances	Evaluation Labels	Workers	Votes
AC2	4	11,040	333	825	89,948
BM	2	1,000	1,000	83	5,000
CSv3B	2	42,624	550	57	214,665
HC	4	20,232	4,459	766	97,164
HCB	2	20,026	3,277	762	90,564
WVSCM	3	2,134	159	64	19,287

each worker's ability which can be either positive for helpful workers or negative for adversarial. Dawin and Skene (DS) [Dawid and Skene, 1979] estimates a confusion matrix for each worker to provide a more detailed model of worker ability. Naive Bayes (NB) [Snow et al., 2008] also employs a confusion matrix, but with Laplace smoothing. The Generative model of Labels, Abilities, and Difficulties (GLAD) [Whitehill et al., 2009] models workers with a scalar parameter and additionally models each instance with a parameter estimating its difficulty. The algorithm for GLAD used in SQUARE is only compatible with datasets containing binary valued votes. Caltech UCSD Binary Annotation Model (CUBAM) [Welinder et al., 2010] estimates workers with parameters for skill and bias while also estimating instance difficulty. The implementation of CUBAM used here is also only applicable to binary data.

2.6.3 Implementation details

For all the experiments in this chapter, CL-LDA uses four Gibbs sampling chains with a burn-in of 200 samples and then takes the average over vote-classes associated with each instance for the next 300 samples of each chain combined. CL-LDA-BPE has a longer burn-in of 4,000 samples to allow the BPE to converge and then averages the next 1,000 samples. Equations 2.9 and 2.10 are applied until convergence after every 20 complete Gibbs sampling iterations during the burn-in period, beginning after the first 100 samples.

For both methods, $\underline{\beta}^{\mathcal{U}}$ is set to $3 \cdot (0.9 \cdot \underline{\boldsymbol{I}}^{W \times W} + 0.1 \cdot \underline{\mathbf{1}}^{W \times W}/W)$, which weakly assumes that workers are proficient. $\vec{\alpha}$ is set to a uniform vector with total sum of 0.5 which effectively adds a pseudovote with weight of 0.5 to each instance. Each vote-class is initialized to the class with the highest probability to produce the relevant vote given the worker's prior, i.e., $\arg\max_{l} \beta^{u}_{kl}$.

2.6.4 Results

Both implementations of CL-LDA generally perform as well as other CL methods. Each class of algorithms perform better or worse on any given dataset based on the characteristics specific to that dataset and those variations can be seen in the following results.

Found datasets

The performance of both versions of CL-LDA on the SQUARE datasets are shown in Table 2.3 and Table 2.4. All performance metrics are calculated using the mean across class-specific performance. Accuracies among all algorithms are similar on AC2, BM, and CSv3B. Only on HC, HCB, and WVSCM is there a wider range of accuracies and for two of those three, CL-LDA is the top performer. For precision, both CL-LDA methods display a wider range in performance. When compared to DS, which is heavily cited and uses a very similar model, CL-LDA and CL-LDA-BPE do better on almost all datasets in both accuracy and precision.

Simulated datasets

The simulated data experiments indicate that CL-LDA performs better than MV under the influence of many spammers, as shown in Figure 2.3. CL-LDA-BPE and DS are also affected much less than MV and also performs better than CL-LDA in this case. ZC strangely appears to perform better with the influence of spammers though suffers from high variability with random spammers. These results should be interpreted with some skepticism as DS and CL-LDA-BPE

Table 2.3: Unsupervised accuracy on found data. Bold values indicate best performance for each dataset.

Algorithms	Datasets					
	AC2	BM	CSv3B	HC	HCB	WVSCM
CL-LDA	0.876	0.813	0.960	**0.947**	**0.664**	0.757
CL-LDA-BPE	**0.885**	0.812	0.972	0.938	0.631	0.786
CUBAM	-	0.804	0.942	-	0.646	0.671
DS	0.850	0.812	0.958	0.926	0.643	0.772
GLAD	-	0.811	0.972	-	0.325	**0.826**
MV	0.867	0.812	0.973	0.884	0.506	0.710
RY	-	**0.818**	0.972	-	0.493	0.809
ZC	0.833	0.815	**0.975**	0.684	0.271	0.818

Table 2.4: Unsupervised precision on benchmark data. Bold values indicate best performance for each dataset.

Algorithms	Datasets					
	AC2	BM	CSv3B	HC	HCB	WVSCM
CL-LDA	0.507	**0.636**	0.919	0.647	0.722	0.716
CL-LDA-BPE	**0.540**	0.578	0.943	0.654	0.758	0.717
CUBAM	-	0.613	0.882	-	0.736	**0.746**
DS	0.491	0.613	0.915	0.664	0.734	0.690
GLAD	-	0.573	0.943	-	0.773	0.700
MV	0.481	0.576	0.944	0.595	0.746	0.729
RY	-	0.586	0.943	-	**0.778**	0.694
ZC	0.405	0.530	**0.950**	**0.666**	0.771	0.318

did not perform in such a superior manner on the found CL datasets as compared to CL-LDA. Still, these experiments provide some insight into each algorithm's robustness to poor workers.

Effects of Bayesian prior estimation

CL-LDA-BPE does perform better than CL-LDA in certain cases, but it also does worse in other as demonstrated by some of the non-simulated datasets. In effect, BPE becomes beneficial when the provided priors greatly misalign with the true class distributions. This claim is supported by the increased efficacy of CL-LDA-BPE when applied to the simulated data as the initial priors impose the beliefs that all classes are equally likely and that all workers are somewhat competent. Once an initial estimate has been formed, BPE can relearn priors to support the data. It follows that in cases when some idea of the data distribution and worker capability is already known, CL-LDA is sufficient. Applying BPE to only the class distribution prior or only to the workers, even to just a subset of workers, is not only possible, but very easy as well. A disadvantage to CL-LDA-BPE is that the BPE requires significantly more Gibbs sampling iterations, and therefore more time, to converge than CL-LDA.

2.7 Application to the ICLabel dataset

We demonstrate our method on the challenging problem of labeling unmixed, independent components (ICs) of multidimensional electroencephalography (EEG) data [Makeig et al., 1996], making full use of all four generalizations listed in Section 2.2. We apply CL-LDA to crowdsourced label suggestions to provide training labels for a subset of the ICLabel dataset, a collection of over 200,000 EEG ICs that previously has no such labels, which is described in detail in Chapter 3. Generating labels for a subset of the ICLabel dataset allows the use of semi-supervised learning algorithms on the entire dataset, enabling the creation of an automated EEG IC classifier to aid neuroscientists in analyzing large collections of datasets, to help and

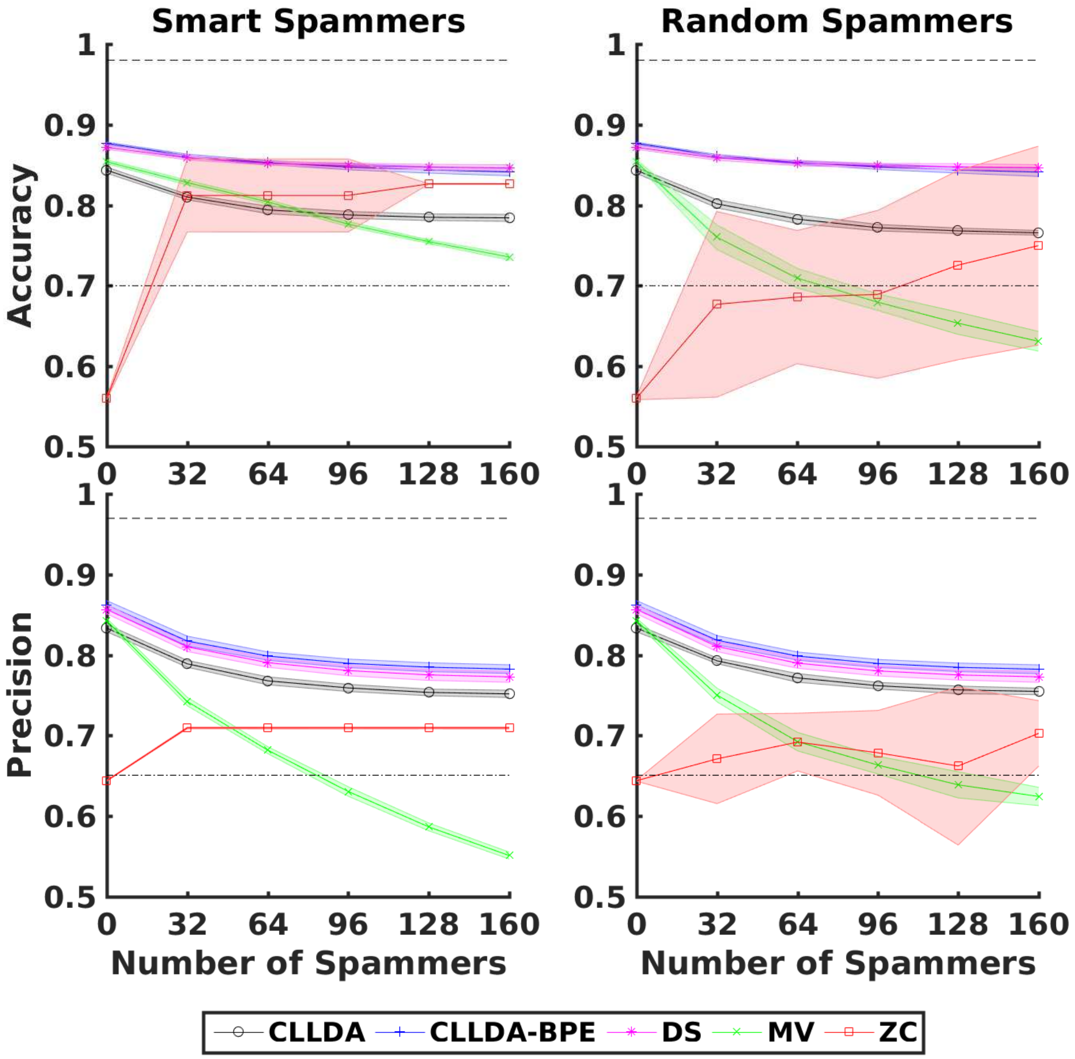


Figure 2.3: Effects of varying numbers and types of spammers on CL-LDA, CL-LDA-BPE and other CL algorithms. The black dashed line is the expected performance for the proficient workers while the black dot-dashed line is the expected performance for the medium performance workers. Central lines indicate mean performance over 10 datasets. Colored area around central lines indicate the standard deviation of performances over 10 datasets.

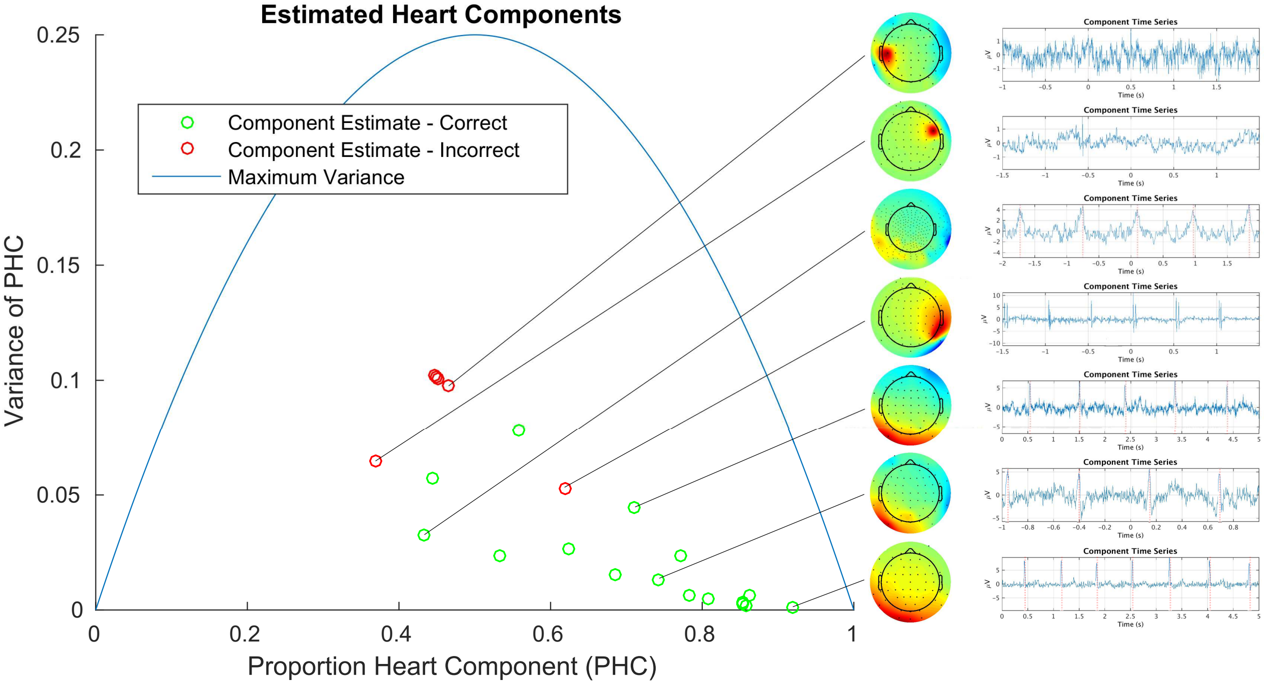


Figure 2.4: EEG components from the ICLabel dataset that CL-LDA estimated as mostly capturing signals generated from a subject's heart. The horizontal axis shows how strong the estimate is while the vertical axis shows how stable that estimate is. The blue parabola represents the maximum estimate variance which would only occur in the case of estimates drawn from multinomial distributions. Green dots represent actual Heart components while red dots are false positives. A clear trend can be seen with false positives having lower strength and higher variance estimates. On the right, exemplar EEG components are shown. The cartoon head visualizes the resulting pattern of electrical potentials at the scalp resulting from activity of that EEG component. Red, green, and blue represent positive, neutral, and negative polarity respectively with recording electrode positions shown as black dots. The graph to the right of each head show a segment of time series activity from that component. The only high strength false-positive has time series activity that closely resembles the QRS complex that is highly characteristic of heart activity.

Table 2.5: ICLabel dataset metrics as of 2017.

Classes	ICs	Experts	Votes	Non-Experts	Votes
7	4,375	3	2,596	19	8,754

teach those who do not know how to distinguish ICs manually, and for applications which require automation such as certain real-time brain-computer interfaces.

ICs are separated into seven classes based on the estimated source of the EEG component signal. The classes are “Brain” for signals originating from a subject’s cerebral cortex, “Muscle” for signals generated by muscle activity, “Eye” for electrical potentials produced by the retina, “Heart” for components that account for the electrical activity from the heart, “Line Noise” for components following external electrical fields produced by nearby power fixtures or electronics, “Channel Noise” for artifacts resulting from poor electrode quality or loose electrode contacts, and “Other” as an amalgamation of additional rare classes and poorly unmixed or otherwise uninterpretable signals which provide little or no usable information to neuroscientists. Information regarding the ICLabel dataset as it was at the time of this analysis can be seen in Table 2.5. For more detailed descriptions of the aforementioned IC categories and for further information regarding the ICLabel dataset and the ICLabel classifier, see Chapter 3.

The task of labeling ICs is difficult enough that when asking experts to evaluate the same components, it is not uncommon for them to disagree on the correct labels for a significant number of those components. Because of this difficulty and the occasionally imperfect unmixing that may result from the algorithms used, compositional labels on ICs provide a more informative model when ascribing meaning or origin to an IC. By describing ICs as a composition, as CL-LDA allows with generalization 1, labels can express similarity to multiple classes while maintaining the capacity to define an IC as primarily from a single class. Pursuant to the compositional label model, votes are also not limited to a single class. Instead, using generalization 2, workers may select any number of classes they find to be applicable to a given IC. In cases when a worker feels significant doubt in his or her assessment of an IC, the worker can indicate that uncertainty through an additional “?” response which may be used in addition to uncertain guesses or alone as a way to abstain from voting, as provided for by generalization 3. Finally, there is a subset of workers whom we, a priori, deem to be experts. This information is incorporated into the CL model to counteract any biases in the general population of workers [Della Penna and Reid, 2012]. As “expert” does not mean infallible in this context, their individual votes cannot be treated as ground truth. Information regarding expert skill is instead incorporated into the model using generalization 4, i.e., worker matrix priors $\vec{\beta}_k^u$.

2.7.1 Implementation details

When applying CL-LDA to the ICLabel dataset, the matrix prior for experts and non-experts are set to $30 \cdot \left[\underline{\mathbf{I}}^{7\times 7} \quad \mathbf{1}^{7\times 1}/7\right]$ and $30 \cdot \left[(0.4 \cdot \underline{\boldsymbol{I}}^{W\times W} + 0.6 \cdot \underline{\mathbf{1}}^{W\times W}/W) \quad \mathbf{1}^{7\times 1}/2\right]$ respectively. These priors effectively add 30 pseudo-votes to each possible vote value for a total of 240 pseudo-votes per worker which strongly assumes a skill gap between experts and non-experts, but not so strong that those assumptions cannot be overcome by workers who cast many votes. While such a strong prior may appear excessive, it is helpful in overcoming many common biases among non-expert workers for this task. The class prior vector is set to $\left[0.15 \quad 0.1 \quad 0.05 \quad 0.025 \quad 0.025 \quad 0.025 \quad 0.15\right]$ for ICs of type “Brain”, “Muscle”, “Eye”, “Heart”, “Line Noise”, “Channel Noise”, and “Other” respectively. As in Section 2.6.3, CL-LDA is run with a 200 sample burning and an average is taken over the following 300 samples.

2.7.2 Results

Without ground-truth labels, evaluation is difficult on the ICLabel dataset. The occasional ambiguity between EEG component class types adds to the challenge as well. To overcome these difficulties, we inspect Heart components as they are very clearly defined while still necessitating a level of skill to identify. Heart components are rare, allowing for a full manual analysis of all components with any “Heart” votes. For this evaluation, classes are separated by taking the maximum class contribution to each compositional label, solely to simplify manual analysis. The manual classification of Heart components was done with the help of a cardiologist. Figure 2.4 shows all components that CL-LDA estimates as primarily Heart component. Of the 23 instances selected, 16 are actually “Heart” while the other five are not. Of all 91 ICs that have any “Heart” votes, 21 are actually “Heart”. Combining this information provides an accuracy of 89%. For comparison, MV only accurately labels 12 heart components and achieves an overall accuracy of 82%. More importantly, the results in Figure 2.4 are almost linearly separable. As the proportion Heart component (PHC) decreases and the variance of PHC increases, the less likely a component is to be an actual Heart component.

2.7.3 Label variance

Label variances can potentially provide an important benefit to the broader goal of the ICLabel dataset, and generally to other CL datasets as well. CL-LDA separates the estimate into a composition over classes and generates variance measures on fractions of the composition as an analog to confidence. By having an explicit measure of the estimate credibility, heavy voting overlap (which implicitly ensures a level of label confidence) is no longer a requirement and therefore should allow workers to label more unique components, thereby increasing the expected number of low probability components in the labeled dataset. To fully utilize this approach in CL, the classifier has to incorporate confidence values during training by assuming label heteroscedasticity. An exemplary method that makes this assumption is generalized least squares (GLS) which scales errors according to the noise covariance using Mahalanobis distance. The effect is to diminish the penalty of misclassifying an instance along dimensions with high label variance. Therefore, if a class estimate is the product of a single unreliable worker, the resulting penalty for misclassification will be minuscule and the classifier will not be heavily skewed as a result of the inaccurate label. In fact, CL-LDA can just as easily provide an estimate of the full label covariance which matches the GLS example more closely. This benefit is clearly applicable to the ICLabel dataset results shown in Figure 2.4 as all but one of the incorrectly labeled heart components have high label variance in addition to lower PHC. Figure 2.4 is indicative of how label variances provides an additional layer of information that can aid the generation of the EEG IC classifier and other CL dependent applications and comprehensive validation of the claim will be presented in future work.

2.8 Conclusion

In this chapter we have presented a new CL algorithm, CL-LDA: a generalization of the well known latent Dirichlet allocation (LDA). Using SQUARE, we show that CL-LDA performs comparable to or better than many other CL algorithms, depending upon the dataset, while allowing for four useful generalizations to the CL problem. These generalizations allow CL-LDA to be used with datasets that require or would benefit from compositional labels, multi-response votes, class-agnostic responses, and structured Bayesian incorporation of prior knowledge regarding worker abilities. Furthermore, CL-LDA provides variance estimates on each class proportion assigned to an instance as a measure of confidence. We discuss the convenience of using a method based upon LDA as it provides access to an extensive literature with which to easily extend CL-LDA; a fact which we exploit by incorporating Bayesian prior learning of all priors in CL-LDA-BPE. We show CL-LDA-BPE to be better in cases when true class distributions and worker abilities vary strongly from uninformed guesses.

We then apply CL-LDA to the ICLabel dataset which uses all four stated generalizations. EEG components that capture heart signals demonstrate the utility of variance on class proportions to separate poor class labels estimates from those that are more likely to be true. These class label variances can be incorporated into the error function during classifier training making the classifier robust to unreliable labels rather than discarding or suffering from those labels. In future work, such a classifier can be used to further validate the efficacy of CL-LDA by comparing the performance of a classifier trained on labels generating according to the majority vote against that of a classifier trained with labels from CL-LDA.

2.9 Acknowledgements

Chapter 2, in part, is a reprint of the material as it appears in Luca Pion-Tonachini, Scott Makeig, and Ken Kreutz-Delgado. Crowd labeling latent Dirichlet allocation. *Knowledge and Information Systems*, 53(3):749–765, 2017. The dissertation author was the primary investigator of this paper. The authors would like to thank Paul D. Pion, DVM, DipACVIM (Cardiology), for his input on which EEG components contain signals originating from heart activity. This work was supported in part by a gift by the Swartz Foundation (Old Field, NY) and by the National Science Foundation under grant number GRFP DGE-1144086.

Chapter 3

IC Classification

3.1 Introduction and Overview

As independent component analysis (ICA) does not consider any signal or event annotations in conjunction with the electroencephalographic (EEG) data, any structure present in the ICA solution thereby lacks explicit labels. Consequently, the raw ICA output is an unordered and unlabeled set of independent components (IC). One common step towards organizing the results is to standardize the IC scalp projection norms and order ICs by descending time series activity power. Even so, the provenance of each IC signal is difficult to determine without sufficient training and time dedicated to manual inspection; it is for this reason that manual intervention is typically required before EEG sources may be analyzed after ICA-decomposing EEG datasets. An automated solution to determining IC signal categories, referred to as IC classification or IC labeling, would aid the study and use of EEG data in four ways:

1. Provide consistency in the categorization of ICs.

2. Expedite IC selection in large-scale studies.

3. Automate IC selection for real-time applications including brain-computer interfaces (BCI).

4. Guide IC selection for people lacking the necessary training and help them to learn through examples.

In this chapter, we present a new IC classifier, along with the dataset used to train and validate that classifier and the website used to collect crowdsourced IC labels for the dataset. The classifier is referred to as the ICLabel classifier while the dataset and website are referred to as the ICLabel dataset and ICLabel website, respectively. The process for creating and validating the ICLabel classifier began with the creation of the ICLabel dataset and website, as the website was used to annotate the dataset needed to make the classifier.

The first step was to create the ICLabel training set by collecting examples of EEG ICs and pairing them with classifications of those ICs. The ICLabel website (https://iclabel.ucsd.edu/tutorial)was designed with the express purpose of generating these IC labels for ICs that had no prior annotations. The website also functions as an educational tool as well as a crowdsourcing platform for accumulating redundant IC labels from website users. These redundant labels are then combined, using a crowd labeling (CL) algorithm, to generate probabilistic labels for the training set. In addition to the ICLabel training set, we also constructed a second ICLabel expert-labeled test set containing additional ICs not present in the training set, used for classifier validation.

With this foundation in place, the next step was to create and validate the ICLabel classifier. To do so, multiple candidate classifiers were trained using the ICLabel training set and the final ICLabel classifier was modeled after the candidate classifier that best performed on the cross-validated training set. Once trained on the ICLabel training set, the ICLabel classifier was validated against other publicly available IC classifiers on the ICLabel expert-labeled test set. The final products of this process are the ICLabel classifier, dataset, and website, all of which are freely available online. The classifier may be downloaded through the EEGLAB extensions manager under the name ICLabel or may be downloaded directly from https://github.com/sccn/ICLabel. The ICLabel dataset may be downloaded from https://github.com/lucapton/ICLabel-Dataset and the educational ICLabel website is accessible at https://iclabel.ucsd.edu/tutorial.

3.2 Background

3.2.1 EEG component interpretation

When a signal generator produces electric fields with a stable spatial projection pattern across the recording electrodes, ICA decomposition may capture that activity in one IC. Perfect separation of source signals is not always possible and, often, is difficult to verify without concurrent invasive recordings. Suboptimal signal unmixing can happen because of poor ICA convergence due to an insufficient amount of clean data or excessive artifacts and noise in the data. Some source signals cannot be fully described in one IC, as when signal source projections are not spatially stationary. However, due to the iterative nature of the convergence of ICA algorithms, most ICs primarily account for one specific source signal, even when some sources are not perfectly separated [Hsu et al., 2014]. To simplify further discussion, rather than referring to, for example, “primarily brain-related” or “non-brain-related” ICs, ICs accounting predominantly for activity originating within the brain will be referred to as “Brain ICs”. This verbal denotation can be generalized to any number of IC categories, the definitions of which are provided in Section 3.2.1. While this denotation is simpler to read and write, it also hides the possibility of complexities and imperfections in the ICs and in the signals they describe. It is therefore important that the reader not forget the possible intricacies masked by this simple nomenclature.

3.2.2 Prior methods

Several other attempts to automatically solve the IC classification problem have been made publicly available. A recent and largely comprehensive summary of those methods can be found in the introduction of Tamburro et al. [2018]. For our purposes, we only consider and compare methods and their supporting algorithms that are (1) publicly available, (2) do not require any information beyond the ICA-decomposed EEG recordings and generally available meta-data such as electrode locations, and (3) have at minimum a category for Brain ICs as defined in Section 3.2.1. This excludes IC classification methods that have not released the trained classifiers, classifiers that only classify certain non-brain artifacts, and methods that require additional recordings such data from an electrooculogram (EOG), ECG, electromyogram (EMG), or accelerometer.

Provided the first two constraints hold, a direct comparison of all accessible methods on a common collection of datasets becomes possible and is presented in Section 3.4.1. EEG IC classifiers that matched the above criteria are summarized here:

- **MARA** [Winkler et al., 2011, 2014] is an IC classifier that estimates the probability of ICs being either (non-brain) artifactual or Brain ICs. It uses a regularized LDA model trained on 43 10-minute EEG recordings from eight subjects consisting of 1290 ICs. All ICs were labeled by two experts. All recordings used the same experimental paradigm.

- **ADJUST** [Mognon et al., 2011] classifies ICs into five discrete categories, three of which are related to eye activity. Its feature-specific thresholds were learned from 20 EEG recordings for a single experimental paradigm.

- **FASTER** [Nolan et al., 2010] was intended as a full processing pipeline that cleans unprocessed, raw EEG data. Only the portion that classifies ICs is considered here. FASTER labels an IC as "artifactual" if any of the features it calculates deviates from the dataset average by more than three standard deviations.

- **SASICA** [Chaumon et al., 2015] performs semi-automatic classification based on features from MARA, FASTER, and ADJUST plus additional features. SASICA was primarily intended as an educational tool to help users learn how to manually label ICs. It uses feature-specific thresholds to determine which ICs should be rejected, presumably keeping only Brain ICs for further analysis. When operating automatically, SASICA uses thresholds between two to four standard deviations from the dataset average. Alternatively, thresholds may be manually chosen.

- **IC_MARC** [Frølich et al., 2015] uses a multinomial logistic regression model trained on 46 EEG recordings comprising 8023 ICs and two experimental paradigms. The associated publication describes two versions. In the first, the features were selected using two-level cross-validation over a larger initial set of features, referred to as the established feature set (IC_MARC$_{\text{EF}}$). The second version uses selected spatial features and, while originally intended for short recordings, appears to work better in practice, and is referred to below as the spatial feature set (IC_MARC$_{\text{SF}}$). Both versions compute probabilistic labels over six classes, two of which are related to eye activity.

Despite the existence of these IC classification methods and others, there remains room for improvement by increasing output *descriptiveness*, *accuracy*, and *efficiency*, terms which are defined as follows. An IC classifier can be said to be more *descriptive* if it can differentiate between a larger number of useful IC categories and if the classifications provided are probabilistic across all relevant categories rather than discrete, single-category determinations. In the case of an ambiguous EEG component with hard labels, there is no recourse to convey that ambiguity. If a discrete classifier produces an incorrect component label, there is also no way to find the next best category from the discrete classification. FASTER, ADJUST, and SASICA are examples of classifiers that produce discrete classifications. This is discussed further in Section 3.5.1.

Accuracy refers not only to classifier performance on the same type of data it was trained on, but how well that classifier's performance generalizes across all EEG data, independent of experiment, recording environment, amplifier, electrode montage, preprocessing pipeline, etc. Though measuring performance across all possible datasets is infeasible, computing performance across multiple experiments and recording conditions should be a minimum requirement. The previous methods listed above used one or two experiment types with the exception of SASICA and MARA which used more. Furthermore, because even expert human IC classifiers often disagree [Chaumon et al., 2015, Frølich et al., 2015] it is important to find a consensus among multiple labelers. This is a matter that many of the prior projects handled well, although some did not explicitly report how many labelers, expert or otherwise, were used.

Efficiency refers to the computational load and speed of extracting the required IC features and computing IC classifications. While generally beneficial, efficiency is only situationally important. Specifically, efficiency is paramount when IC classification is desired for online streaming data. Without a computationally efficient classifier, the delay incurred when classifying ICs may negate any utility gained through obtaining the classifications. In offline cases, efficiency is merely a matter of convenience and, possibly, of cost.

3.2.3 The ICLabel project

The ICLabel project provides improved classifications based on the aforementioned desirable qualities of an EEG IC classifier. To be sufficiently *descriptive*, the ICLabel classifier computes IC class probabilities across seven classes as described below. To achieve *accuracy* across EEG recording conditions, the ICLabel dataset used to train and evaluate the ICLabel classifier encompasses a wide variety of EEG datasets from a multitude of paradigms. These example ICs are paired with component labels collected through the ICLabel website from hundreds of contributors. Finally, to maintain sufficient computational *efficiency*, relatively simple IC features are used as input to an artificial neural network architecture (ANN) that, while slow to train, computes IC labels quickly. The end result is made freely and easily available through the ICLabel plug-in for the EEGLAB software environment [Delorme and Makeig, 2004, Delorme et al., 2011].

The seven IC categories addressed in this work are:

- **Brain** ICs contain activity believed to originate from locally synchronously activity in one (or sometimes two well-connected) cortical patches. The cortical patches are typically small and produce smoothly varying dipolar projections onto the scalp. Brain ICs tend to have power spectral densities with inversely related frequency and power and, often, exhibit increased power in frequency bands between 5 and 30 Hz. See Figure 3.1 for an example of a Brain IC.

- **Muscle** ICs contain activity originating from groups of muscle motor units (MU) and contain strong high-frequency broadband activity aggregating many MU action potentials (MUAP) during muscle contractions and periods of static tension. These ICs are effectively surface EMG measures recorded using EEG electrodes. They are easily recognized by high broadband power at frequencies above 20–30 Hz. Often times they can appear dipolar like Brain ICs, but as their sources are located outside the skull, their dipolar pattern is much more localized than for Brain sources.

- **Eye** ICs describe activity originating from the eyes, induced by the high metabolic rate in the retina that produces an electrical dipole (positive pole at the cornea, negative at the retina) [Malmivuo and Plonsey, 1995]. Rotating the eyes shifts the projection of this standing dipole to the frontal scalp. Eye ICs can be further subdivided into ICs accounting for activity associated with horizontal eye movements and ICs accounting for blinks and vertical eye movements. Both have scalp projections centered on the eyes and show clear quick or sustained "square" DC-shifts depending on whether the IC is describing blinks or eye movements respectively.

- **Heart** ICs, though more rare, can be found in EEG recordings. They are effectively electrocardiographic (ECG) signals recorded using scalp EEG electrodes. They are recognizable by the clear QRS-complexes [Malmivuo and Plonsey, 1995] in their time series and often have scalp projections that closely approximate a diagonal linear gradient from left-posterior to right-anterior. Heart ICs can rarely have localized scalp projections if an electrode is placed directly above a superficial vein or artery.

- **Line Noise** ICs capture the effects of line current noise emanating from nearby electrical fixtures or poorly grounded EEG amplifiers. They are immediately recognizable by their high concentration of power at either 50 Hz or 60 Hz depending on the local standard. These effects can only be well separated if the line noise interference is spatially stationary across the EEG electrodes. Otherwise, it is unlikely that a single IC will be able to describe the line noise activity. Instead, several or even all components may be contaminated to varying degrees.

- **Channel Noise** ICs indicate that some portion of the signal recorded at an electrode channel is already nearly statistically independent of those from other channels. These components can be produced by high impedance at the scalp-electrode junction or physical electrode movement, and are typically an indication of poor signal quality or large artifacts affecting single channels. If an ICA decomposition is primarily comprised of this IC category, that is a strong indication that the data has received insufficient preprocessing. In this chapter, "Channel Noise" will sometime be shortened to "Chan Noise".

- **Other** ICs, rather than being an explicit category, act as a catch-all for ICs that fit none of the previous types. These primarily fall into two categories: ICs containing indeterminate noise or ICs containing multiple signals that ICA decomposition could not separate well. For ICA-decomposed high-density EEG recordings (64 channels and above), the majority of ICs typically fall into this category.

3.3 Materials and Methods

3.3.1 ICLabel dataset and website

The ICLabel training set used to train the ICLabel classifier currently has been drawn from 6,352 EEG recordings collected from storage drives at the Swartz Center for Computational

Neuroscience (SCCN) at UC San Diego (https://sccn.ucsd.edu). These datasets come from many studies which encompass a portion of the experiments recorded at the SCCN and those brought to the SCCN by visiting researchers since 2001. Numbers of electrodes used in these studies largely range from 32 to 256, many with 64 or 128. In many of the studies, participants sat facing a computer monitor and pressed buttons to deliver responses to presented visual stimuli. In some studies, subjects were standing, balancing on a force plate, throwing darts, exploring the room space, or making mirroring movements with a partner. There were no studies involving brain stimulation (e.g. transcranial magnetic stimulation (TMS)) and few studies involving children or aged adults. Importantly, the degree of accuracy that can be claimed for the recorded electrode scalp positions differs across studies. In some, the recorded positions were standard template positions only. In other studies, 3D position-measuring systems were used to record electrode positions (e.g. Polhemus or Zybris), but in nearly all cases the DipFit plug-in in EEGLAB adapted the recorded positions to a standard template head model after a by-eye fit to the recorded montage positions. As the EEG recordings were not expected to be accompanied by individual participant magnetic resonance head images, positions of head fiducials were usually not recorded. We believe these recordings represent data typical of psychophysiological experiment data recorded during the past 15 years or so. The considerable variety of methods, montages, and subject populations adds variability that may help the ICLabel classifier to generalize well.

In aggregate, these recordings include a total of 203,307 unique ICs; none of which had standardized IC classification metadata and were therefore effectively unlabeled for the purposes of this project. Prior to computing features, each dataset was converted to a common average reference [Dien, 1998]. For each IC, the ICLabel training set includes a set of standard measures: a scalp topography, median power spectral density (PSD) and autocorrelation function, and single and bilaterally symmetric equivalent current dipole (ECD) model fits, plus features used in previously published classifiers (ADJUST, FASTER, SASICA, described in Section 3.2.2). These features potentially provide an IC classifier with information contributory to computing accurate component labels.

IC features descriptions

Scalp topographies are a visual representation of how IC activity projects to the subject's scalp by interpolating and extrapolating IC projections to each electrode position into a standard projection image across the scalp. These square images, 32 pixels to a side, are calculated using a slightly modified version of the `topoplot` function in EEGLAB. Furthermore, the information required to generate the scalp topographies for each dataset (when available) is also included in the form of the estimated ICA mixing matrix, channel locations, and channel labels. Power spectral densities from 1 to 100 Hz are calculated using a variation of Welch's method [Welch, 1967] that takes the median value across time windows rather than the mean. This version was used because movement artifacts are a common occurrence in EEG datasets and the sample median is more robust to outliers than the sample mean [Hampel et al., 2011].

ECD model estimates are based on a three-layer boundary element method (BEM) forward-problem electrical head template (MNI) and assume that each IC scalp topography is the scalp projection of an infinitely small point-source current dipole inside the skull [Brazier, 1966, Henderson et al., 1975, Adde et al., 2003]. Some ICs require a dual-symmetric ECD model, likely representing the joint activation of cortical patches directly connected across the brain midline, e.g. by the corpus callosum. The ECD model is fit using the DipFit plug-in in EEGLAB which calculates dipole positions and moments that best match the IC scalp topography. The better the resulting fit, the more "dipolar" an IC can be said to be. Examples of some of these features are shown in Figure 3.1.

ICLabel website and label collection

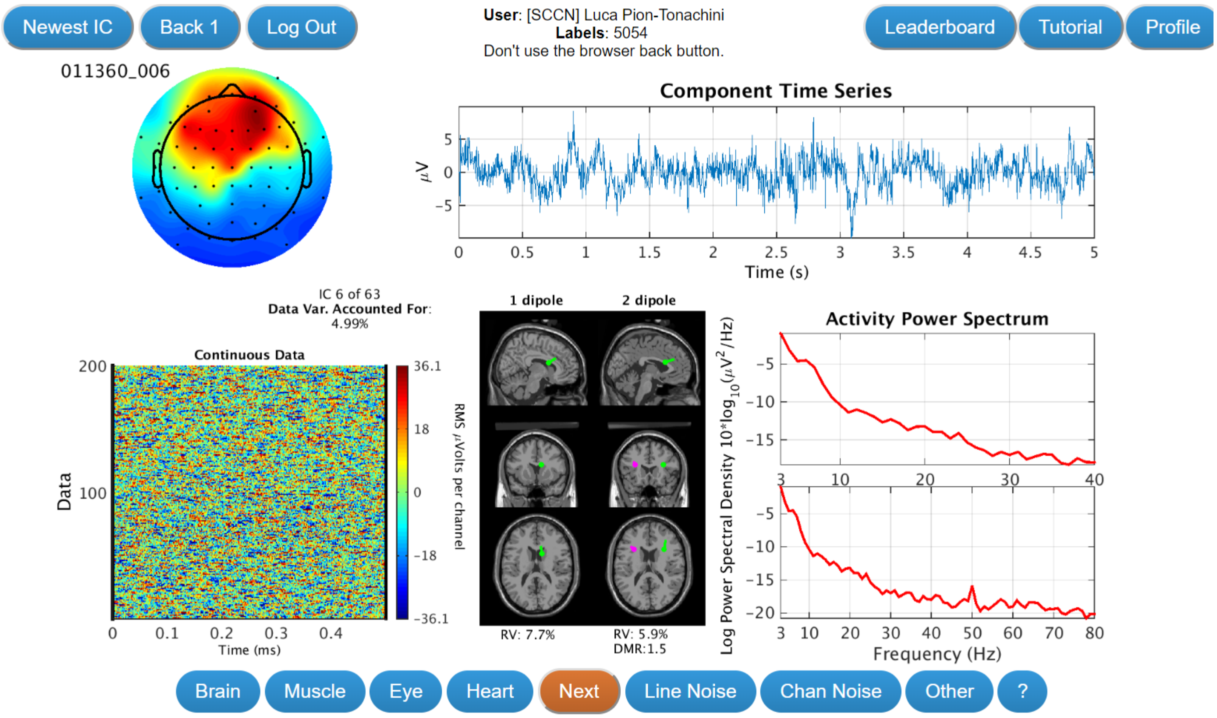
To gather labels for ICs in the ICLabel training set, the ICLabel website (https://iclabel.ucsd.edu/tutorial) was created in the PHP scripting language using the Laravel website framework.

With the help of over 250 contributors, henceforth referred to as "labelers", the ICLabel website collected over 34,000 suggested labels on over 8,000 ICs through the interface illustrated in Figure 3.1. Currently, each labeled IC has an average of 3.8 suggested labels associated with it. The website was advertised through the EEGLAB mailing list of EEGLAB users worldwide, and to the SCCN mailing list for lab members and visitors. The labeler pool is comprised of several IC labeling experts and many more labelers of unknown skill. To mitigate the effect of novices contributing incorrect labels to the database, the website also provides a thorough tutorial on how to recognize and label EEG ICs. In this way, the ICLabel website has become an educational tool. Many visitors to the website read the IC labeling tutorial and use the "practice labeling" tool (https://iclabel.ucsd.edu/labelfeedback) that offers feedback about the labels others have assigned to the provided sample ICs. The "practice labeling" tool currently has been used more than 49,000 times and some professors report using it to train students.

Crowd labeling

To create a coherent set of IC labels accompanying a subset of the ICs in the ICLabel training set, suggested labels collected through the ICLabel website were processed using the crowd labeling (CL) algorithm "crowd labeling latent Dirichlet allocation" (CL-LDA, see Chapter 2). This gave 5,937 usable labeled EEG ICs in the training set. CL algorithms estimate a single "true label" given redundant labels for that IC provided by various labelers. This can be done multiple ways, but every CL method must reconcile disagreeing labels. CL algorithms generally do so by noting which labelers tend to agree with others and which labelers do not, upweighting and downweighting votes from those users respectively. Some methods model only the estimated labels, while others in addition model the apparent skill of each labeler; some even estimate the difficulty of the individual items being labeled.

CL-LDA estimates "true labels" as a compositional vector (vector of non-negative elements that sum to one) for each IC using the redundant labels from different labelers. Com-



User: [SCCN] Luca Pion-Tonachini
Labels: 5054
Don't use the browser back button.

Figure 3.1: An IC labeling example from the ICLabel website (https://iclabel.ucsd.edu/tutorial), which also gives a detailed description of the features shown above. Label contributors are shown the illustrated IC measures and must decide which IC category or categories best apply. They mark their decision by clicking on the blue buttons below, and have the option of selecting multiple categories in the case that they cannot decide on one or believe the IC contains an additive mixture of sources. There is also a “?” button that they can use to indicate low confidence in the submitted label.

positional labels can be thought of as softened discrete labels. In the case of ICs, this is the difference between allowing an IC to be partly “Eye” and partly “Muscle”, or mostly “Brain” plus some “Line Noise”, as opposed to asserting that any particular IC must be surely “Brain” or “Muscle” or some other class. In effect, compositional labels acknowledge that ICs may be partially ambiguous, or might not contain perfectly unmixed signals. Compositional labels can also reveal how ICs of one category may be confused with another category. Further details on CL-LDA and the specific hyperparameters used in the ICLabel dataset are given in Appendix 3.D.

3.3.2 ICLabel expert-labeled test set

IC classification performance on the ICLabel training set is not an ideal indicator of general IC classification performance for two reasons: (1) the labels are crowdsourced, so that, even after applying CL-LDA, there are likely errors in some labels, and (2) the dataset is used many times over in the course of network and hyper-parameter optimization (described in Section 3.3.3) which may have caused some level of implicit overfitting despite measures taken to avoid this.

For these reasons, additional datasets not present in the training set were procured and six experts were asked to label 130 ICs from those datasets. These 130 ICs comprise the ICLabel test set we used to validate the ICLabel classifier and to compare its results against existing IC classifiers. The ten additional datasets came from five different studies, two datasets from each, that had used differing recording environments, experimental paradigms, EEG amplifiers, electrode montages, preprocessing pipelines, and even ICA algorithms. These variations were purposely sought as a surrogate test of the ICLabel classifier’s ability to generalize. As expert labeling is a scarce resource, only a subset of the ICs from the chosen datasets were shown to the experts for labeling. These ICs were selected by sorting the ICs within a dataset by decreasing power and taking the union among the first five ICs, five more ICs at equally spaced intervals in descending order of source power (always including the weakest IC), and the seven ICs with highest selected class probability as per the ICLabel$_{\text{Beta}}$ EEGLAB plug-in for each IC category, so as to more evenly include examples of rare classes such as Heart ICs. This usually produced 12 to 13 selected ICs per dataset, giving a total of 130 ICs in the expert-labeled test set from the ten additional datasets. The six redundant expert labels per IC were also collected through the ICLabel website, a section visible only to labelers manually marked as "experts", and were combined into a single label estimate for each IC using CL-LDA with settings detailed in Appendix 3.D.

3.3.3 ICLabel candidate classifiers

Multiple candidate classifiers were trained and compared to select the architecture and training paradigm best suited for creating the final ICLabel classifier. These candidate versions differed in the feature sets used as inputs, in training paradigm, and in model structure. In this way the ICLabel training set was used to train six candidate ICLabel classifiers. Three artificial neural network (ANN) architectures were tested; all had the same underlying convolutional neural network (CNN) structure used for inference. Figure 3.2 graphically summarizes the three ANN architectures of the ICLabel candidates. Two of those architectures were CNNs trained on only the labeled ICs. The first of those CNNs optimized an unweighted cross entropy loss while the second optimized a weighted cross entropy loss that doubly weighted Brain IC classification errors (wCNN). Cross entropy is a mathematical function that compares two class probability vectors (typically label vectors) and produces a scalar output related to how similar those two vector are. See Appendix 3.A for a more detailed explanation. The third classifier architecture was based on a variation of semi-supervised learning generative adversarial networks (SSGAN) [Odena, 2016, Salimans et al., 2016], an extension of generative adversarial networks (GAN) [Goodfellow et al., 2014]. Detailed descriptions of the ICLabel candidate classifier inputs, architectures, and training paradigms are given in Appendix 3.E for the two CNNs and Appendix 3.B for the GAN.

Each of the three network architectures described here were further differentiated by associating them with two possible groups of input feature sets. The first group used scalp

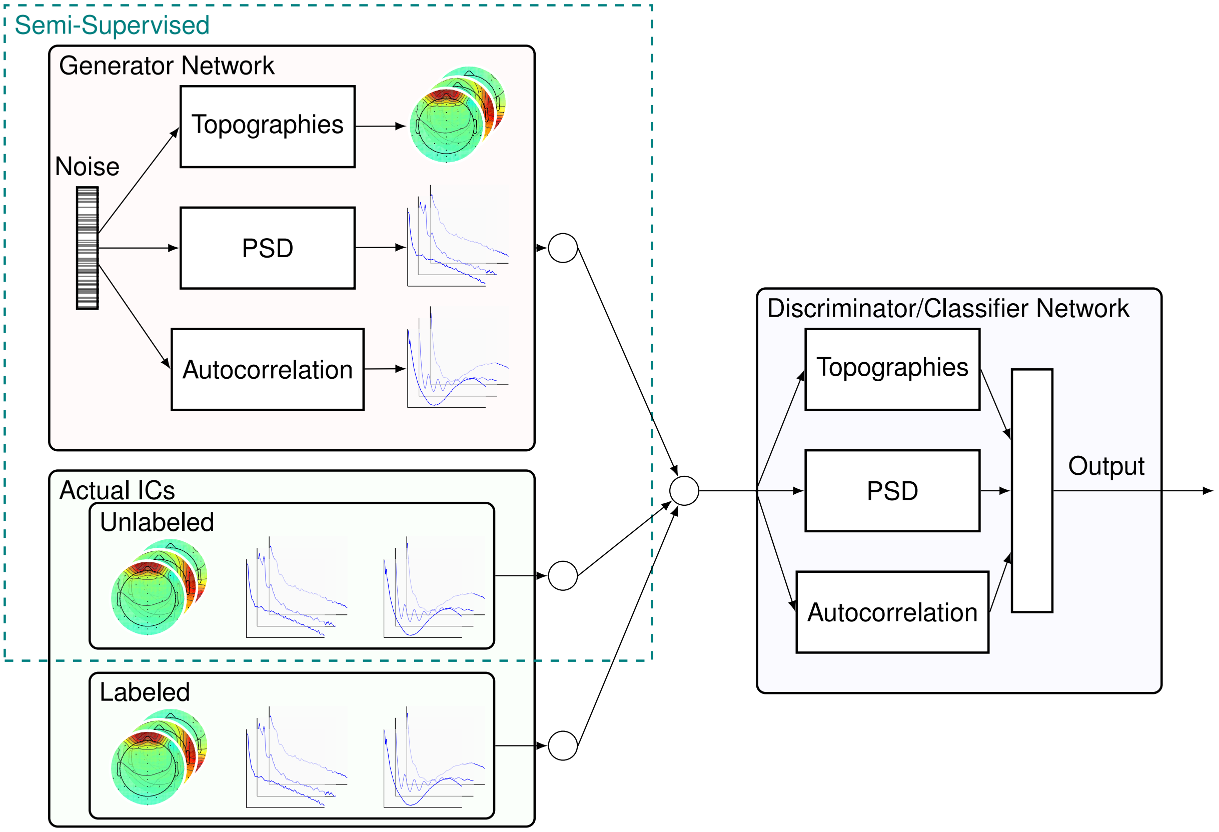


Figure 3.2: Candidate artificial neural network (ANN) architectures tested in developing the ICLabel classifier. White rectangles represent ANN blocks comprised of one or more convolutional layers; arrows indicate information flow. The section in the upper left labeled "Semi-Supervised" (teal dashed outline) was only present in the GAN paradigm during training and was used to generate simulated IC features to compare against unlabeled training examples from the ICLabel training set. The box to the right labeled "Discriminator" remained nearly identical in structure for all three training paradigms (although the parameters used in the final learned network differed). Convergence of arrows into the classifier network indicates the input sources for the classifier during training and does *not* imply data combination, e.g. through summation. After training is complete, classifiers were given *unlabeled* ICs to classify. See Appendix 3.E for a detailed description of the ANN implementations.

topographies and PSDs as inputs, while the second group also used autocorrelation functions. The other feature sets included in the full ICLabel training set were not used by the candidate classifiers as they were either too computationally expensive to compute or were found to not contribute new information in preliminary evaluations beyond the information provided by the scalp topographies, PSDs, and autocorrelation functions.

As described in Appendix 3.E, the ICLabel training set was augmented to four times its original size by exploiting left–right and positive–negative symmetries in scalp topographies. This augmentation was not repeated for the expert-labeled test set. Instead, the final ICLabel classifier internally duplicates each IC to exploit the two scalp topography symmetries and takes the average of the four resulting classifications.

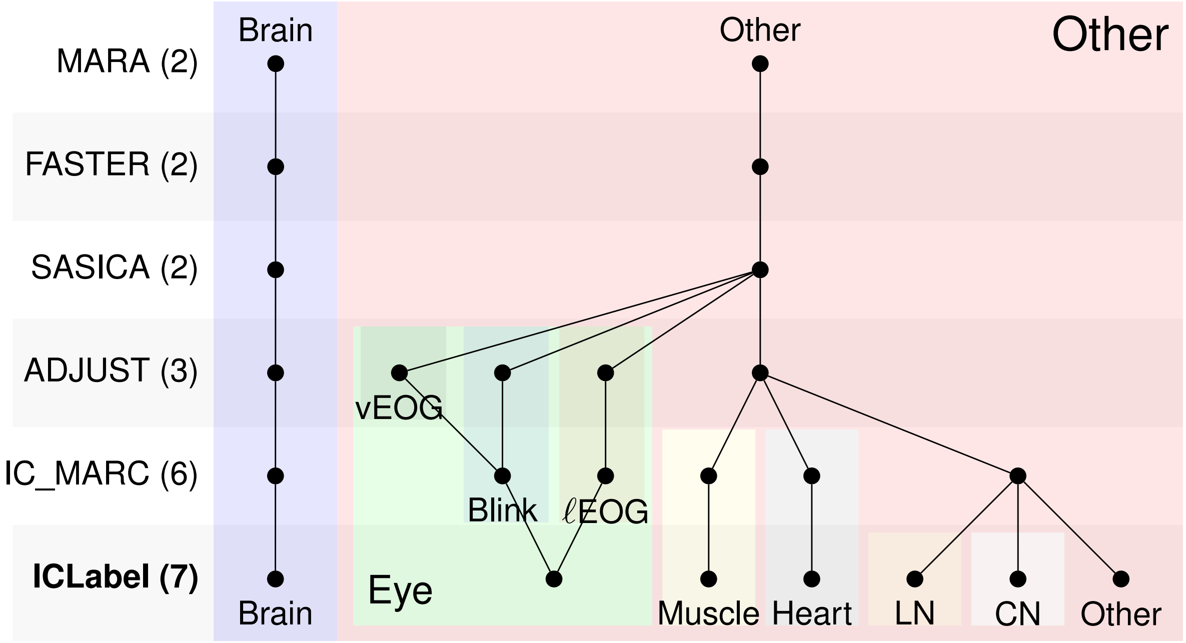
3.3.4 Evaluation

To select the candidate classifier that would become the released ICLabel classifier, six candidate versions of the ICLabel classifier were tested using a three-by-two factorial design with repeated measures on the ICLabel training set. The first factor, ANN architecture, had three levels (described in Section 3.3.3): (1) GAN, (2) CNN, and (3) wCNN. The second factor, feature sets provided to the classifiers, had two levels: (1) networks using only scalp topographies and PSDs and (2) networks also using autocorrelation functions. Below, use of the autocorrelation feature set is indicated by a subscript “AC” following the architecture, as in GAN_{AC}.

To compare the performance of candidate classifiers, the labeled portion of the ICLabel training set was split so as to follow a ten-fold stratified cross-validation scheme. Within each fold, the data were split into training, validation, and testing data (at a ratio of 8:1:1) in a way that attempted to maintain equal class proportions across the three subsets of the labeled data. The training data from each fold was used to train every candidate classifier version, and that fold’s validation data were used to determine when to stop training with early stopping [Prechelt, 2012]. Each fold’s test data were used to calculate the performance of all classifiers trained on that fold's training data. Overall performance for each candidate classifier was taken as the average performance measured across all ten folds. While not relevant to candidate classifier selection, performance of some published IC classification methods was also calculated on the same cross-validation folds. To not waste any training data, the training paradigm that produced the best performing ICLabel candidate was then used to train a new classifier using the best performing candidate architecture with the *entire* ICLabel training set, minus 400 labeled examples now held out as a validation set for early stopping. The resulting classifier became the official ICLabel classifier and was compared to existing methods on the expert-labeled test set.

Performance comparisons between the candidate IC classifiers required a fixed set of IC classes over which to compare scores. As most IC classifiers discriminate between differing sets of IC categories, both in number and interpretation, it was necessary to merge label categories to allow direct classifier comparisons. At one extreme, IC labels and predictions can be reduced to either "Brain" or "Other" to allow comparison of nearly all the IC classifiers. Further subsets could be used for three-, five- and seven-class comparisons, as detailed in Figure 3.3. This study used the five-class and seven-class comparisons as well as the already-described two-class comparison. The five-class comparison combined all eye-related IC categories into a unified Eye IC category and all non-biological artifact ICs and unknown-source ICs into a unified Other IC category. The five-class comparison allowed comparison between the ICLabel candidates and final classifier and all IC_MARC versions, while the seven-class case only allowed comparisons between ICLabel candidates and final classifier.

Classifier performance was measured by comparing balanced accuracy and normalized confusion matrices after discretizing IC labels and predictions, receiver operating characteristic (ROC) curves after discretizing IC labels, ROC equivalent measures from "soft" confusion matrices [Beleites et al., 2013] termed here as *soft operating characteristics* (SOC) points, cross-entropy, and required time to calculate the IC classifications. Further explanation of these measures is given in Appendix 3.A.



vEOG: Vertical EOG; ℓEOG: Lateral EOG; LN: Line Noise; CN: Channel Noise

Figure 3.3: Categories labeled by the IC classifiers that were evaluated on the expert-labeled test set. The top five classifiers listed on the vertical axis are described in Section 3.2.2. The tree structure and colored boxes connecting labels of different classifiers signifies how the classifier labels are related and how they could be merged to allow comparisons between classifiers with non-identical IC categories. For example, all IC classifiers can be compared across two classes by merging all categories contained within the red box into the overarching category of Other ICs. Similarly, all categories in the green box can be simplified to form a single Eye IC category. The following acronyms are used in the above figure: "vEOG" for "vertical EOG activity", "ℓEOG" for "lateral EOG activity", "LN" for "Line Noise", and "CN" for "Channel Noise".

Table 3.1: Scalar performance measures of the tested publicly available independent component (IC) classifiers for different numbers of IC categories. Higher balanced accuracy and lower cross entropy indicate better classification performance.

Classes	Classifier	Balanced Accuracy $\frac{1}{C}\sum_{i=1}^{C}\frac{\mathrm{TP}_i}{\mathrm{TP}_i+\mathrm{FN}_i}$	Cross Entropy $\sum_i t_i \log p_i$
2	ICLabel$_{\mathrm{Lite}}$	**0.855**	**0.339**
	ICLabel	**0.841**	**0.342**
	IC_MARC$_{\mathrm{EF}}$	0.816	0.977
	IC_MARC$_{\mathrm{SF}}$	**0.870**	**0.377**
	ADJUST	0.585	-
	MARA	0.757	0.730
	FASTER	0.578	-
	SASICA	0.775	-
5	ICLabel$_{\mathrm{Lite}}$	**0.623**	**0.938**
	ICLabel	**0.613**	**0.924**
	IC_MARC$_{\mathrm{EF}}$	0.532	2.659
	IC_MARC$_{\mathrm{SF}}$	0.578	0.982
7	ICLabel$_{\mathrm{Lite}}$	0.579	1.287
	ICLabel	0.597	1.251

3.4 Results

3.4.1 ICLabel and prior methods

The ICLabel classifier and the ICLabel$_{\mathrm{Lite}}$ classifier, created as described at the end of Appendix 3.C, were compared against previously-existing, publicly-available IC classifiers. As described in Section 3.3.4, all IC categories besides “Brain” must be conflated to allow a comparison across all IC classification methods simultaneously on the expert-labeled test set. Considering balanced accuracy (higher values are better) and cross entropy (lower values are better) as shown in Table 3.1, in addition to ROC curves for the two-class case as shown in Figure 3.4, the only previously existing classifier competitive with ICLabel was IC_MARC$_{\mathrm{SF}}$. IC_MARC and ICLabel classifiers can be meaningfully compared across five IC categories, as shown in Figure 3.3, and disregarding the other classifiers eliminates the need to aggressively merge non-Brain ICs, allowing a more detailed comparison.

In the five-class comparison, IC_MARC$_{\text{SF}}$ showed marginally better performance than ICLabel when classifying Brain ICs, as measured by ROC curves. SOC points indicated comparable performance whereby IC_MARC$_{\text{SF}}$ achieved a slightly higher soft-TPR than ICLabel at the cost of also having higher soft-FPR. For Muscle ICs, IC_MARC$_{\text{EF}}$ outperformed all other methods as per the ROC curves, despite underperforming on nearly every other measure. Among the three other methods, IC_MARC$_{\text{SF}}$ achieved a higher recall for Muscle ICs after thresholding labels and predictions, as seen in the second row of each five-class confusion matrix (top row of Figure 3.5), despite the corresponding ROC curve not being superior to those of either ICLabel method. Both ICLabel methods performed exceptionally well on Eye ICs, greatly outperforming both IC_MARC versions, as indicated by both the SOC points and ROC curves.

Even though results are shown for Heart ICs, the expert labelers only communally selected one IC as "Heart" and, therefore, the statistical power of results regarding Heart ICs is too low to warrant further discussion. With regard to Other ICs, ICLabel and ICLabel$_{\text{Lite}}$ directly outperformed both IC_MARC models as measured by SOC points while ICLabel and IC_MARC$_{\text{SF}}$ shared the best performance in different regimes of the performance plane as shown by their respective ROC curves. The confusion matrices of Figure 3.5 indicate that most ICLabel errors were derived from over-classifying ICs as "Other", while the causes of IC_MARC$_{\text{SF}}$ errors are difficult to infer.

ICLabel and ICLabel$_{\text{Lite}}$ ROC curves remained nearly unchanged in the seven-class case compared to the five-class case except for Other ICs. SOC points gave similar results, although the distance between optimistic, expected, and pessimistic estimates are larger due to the increased number of IC categories. The additional Line Noise IC and Channel Noise IC categories were classified relatively well, as indicated by the ROC curves, although the scarcity of Line Noise ICs in the expert-labeled test set produced low-resolution ROC curves. SOC points indicate

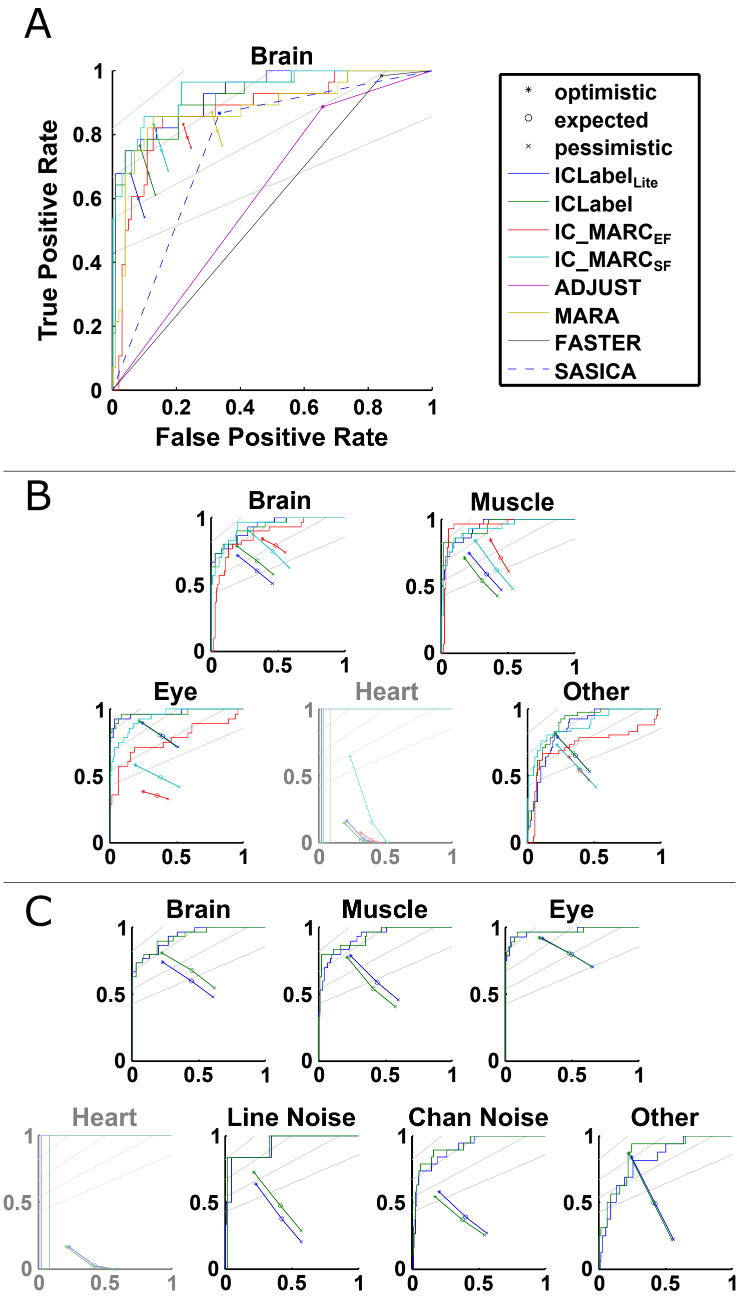
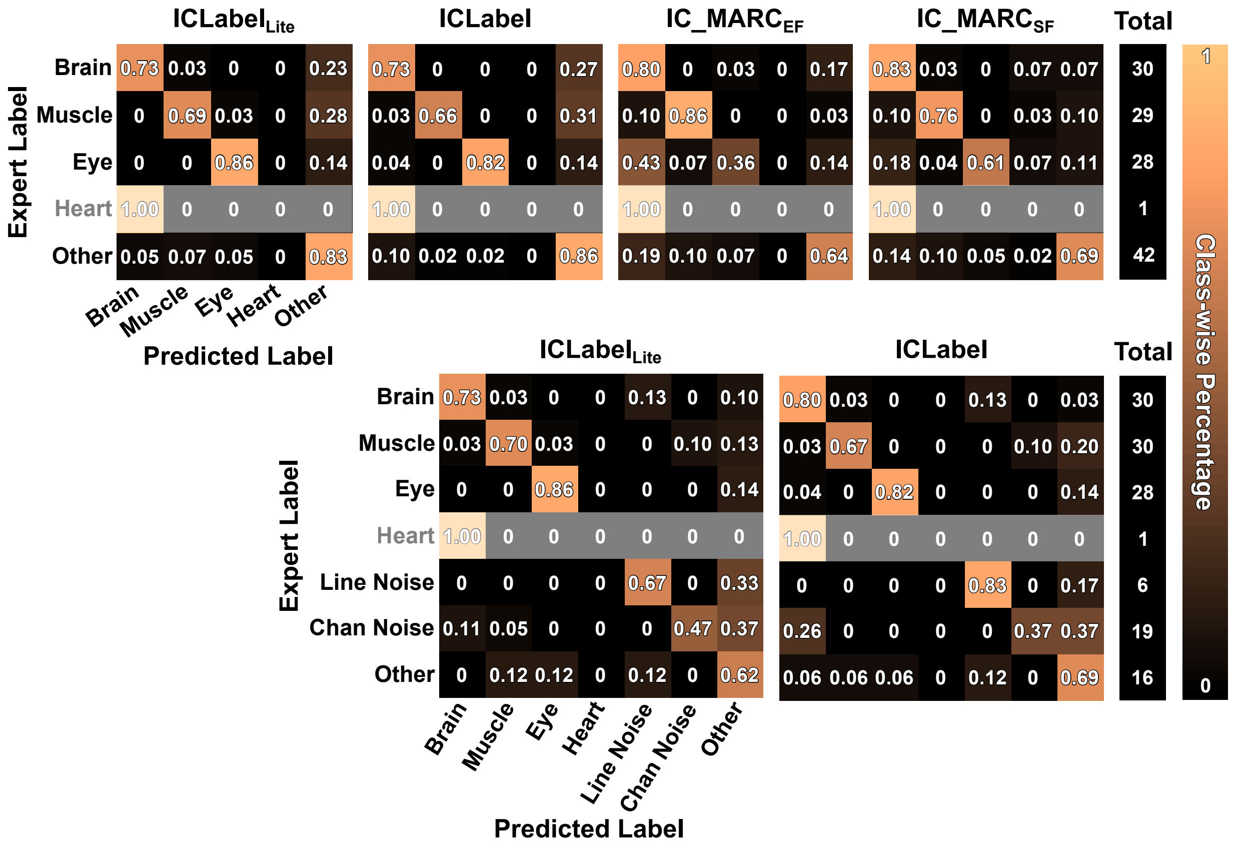


Figure 3.4: Comparison of ICLabel classification performance to that of several alternative publicly available IC classifiers. ROC curves and soft operating characteristics (SOC) points for the (A) two-class, (B) five-class, and (C) seven-class performances on the expert-labeled test set. Gray lines indicate F_1 score isometrics of 0.9, 0.8, 0.7, and 0.6 (from top to bottom). "Heart" plots have been grayed out because experts marked only one IC as being heart-related leading to largely uninformative SOC points and ROC curves for that category. Refer to Appendix 3.A for definitions of F_1 score, ROC curves (traced out by the detection threshold parameter), and SOC points (shown for optimistic, expected, and pessimistic performance estimates as described in Appendix 3.A).



ICLabel$_{\text{Lite}}$

	Brain	Muscle	Eye	Heart	Other
Brain	0.73	0.03	0	0	0.23
Muscle	0	0.69	0.03	0	0.28
Eye	0	0	0.86	0	0.14
Heart	1.00	0	0	0	0
Other	0.05	0.07	0.05	0	0.83

ICLabel

	Brain	Muscle	Eye	Heart	Other
Brain	0.73	0	0	0	0.27
Muscle	0.03	0.66	0	0	0.31
Eye	0.04	0	0.82	0	0.14
Heart	1.00	0	0	0	0
Other	0.10	0.02	0.02	0	0.86

IC_MARC$_{\text{EF}}$

	Brain	Muscle	Eye	Heart	Other
Brain	0.80	0	0.03	0	0.17
Muscle	0.10	0.86	0	0	0.03
Eye	0.43	0.07	0.36	0	0.14
Heart	1.00	0	0	0	0
Other	0.19	0.10	0.07	0	0.64

IC_MARC$_{\text{SF}}$

	Brain	Muscle	Eye	Heart	Other	Total
Brain	0.83	0.03	0	0.07	0.07	30
Muscle	0.10	0.76	0	0.03	0.10	29
Eye	0.18	0.04	0.61	0.07	0.11	28
Heart	1.00	0	0	0	0	1
Other	0.14	0.10	0.05	0.02	0.69	42

ICLabel$_{\text{Lite}}$

	Brain	Muscle	Eye	Heart	Line Noise	Chan Noise	Other
Brain	0.73	0.03	0	0	0.13	0	0.10
Muscle	0.03	0.70	0.03	0	0	0.10	0.13
Eye	0	0	0.86	0	0	0	0.14
Heart	1.00	0	0	0	0	0	0
Line Noise	0	0	0	0	0.67	0	0.33
Chan Noise	0.11	0.05	0	0	0	0.47	0.37
Other	0	0.12	0.12	0	0.12	0	0.62

ICLabel

	Brain	Muscle	Eye	Heart	Line Noise	Chan Noise	Other	Total
Brain	0.80	0.03	0	0	0.13	0	0.03	30
Muscle	0.03	0.67	0	0	0	0.10	0.20	30
Eye	0.04	0	0.82	0	0	0	0.14	28
Heart	1.00	0	0	0	0	0	0	1
Line Noise	0	0	0	0	0.83	0	0.17	6
Chan Noise	0.26	0	0	0	0	0.37	0.37	19
Other	0.06	0.06	0.06	0	0.12	0	0.69	16

Figure 3.5: Normalized ICLabel and IC_MARC confusion matrices calculated from the expert-labeled test set using five classes (top row) and seven classes (bottom row). Rows and columns of each confusion matrix contain all ICs labeled as a particular class by experts and the classifiers, respectively. Rows were normalized to sum to one such that each element along the diagonal represents the true-positive-rate (recall) for that IC category. The "Total" columns on the right indicate how many ICs were labeled as each class by the experts (used for normalization). "Heart" rows have been grayed out because experts marked only one IC as being heart-related leading to largely uninformative results for that row.

some level of disagreement between the experts and ICLabel with regards to the overall label composition on these two IC categories due to the lower soft TPR values shown. The seven-class confusion matrix showed ICLabel to have much lower accuracy on Channel Noise ICs than would be expected from the ROC curves, but corroborated the unfavorable SOC points. The ROC curves for Other ICs were slightly degraded with respect to those in the five-class case, despite the SOC points remaining comparable. This could be due to the apparent difficulty in discriminating between Channel Noise ICs and Other ICs (sixth row of the ICLabel confusion matrix in Figure 3.5).

Even though IC_MARC$_{\text{SF}}$ had 10% higher recall for Brain ICs than ICLabel in the five-class comparison, that gap nearly disappeared in the seven-class comparison. ICLabel's diminished recall of Brain ICs in the five-class case was likely a side effect of the approach used to merge classes. The summed probabilities of multiple, less probable classes can total to more than the probability of the maximal class in the unmerged comparison, possibly changing the IC classification of a single IC across the multiple comparisons. For example, while a label vector $\begin{bmatrix} 0.45 & 0.4 & 0.15 \end{bmatrix}$ has maximal probability of belonging to the first class type, if the second and third classes are merged, the label vector becomes $\begin{bmatrix} 0.45 & 0.55 \end{bmatrix}$ and the first class is no longer the most probable[1]. This only affected one and five ICs of the 130 total ICs for ICLabel$_{\text{Lite}}$ and ICLabel, respectively, when comparing the two-class and seven-class classifications.

3.4.2 IC classification speed

Empirically-determined IC classification speeds can be found in Figure 3.6. Both IC_MARC versions required similar run times: median 1.8 s per IC. ICLabel$_{\text{Lite}}$ and ICLa-

[1] This suggests an alternative means of performing the two-class and five-class comparisons: rather than first conflating the class probabilities through summation and then determining the maximal component, instead find the maximal IC category first and then combine the category labels. This method assures consistent discrete labels across varying numbers of IC categories. However, such a scheme prevents the use of measures dependent on predicted class probabilities such as cross entropy, ROC curves, and SOC points. It is for this reason that label conflation was performed as described in Section 3.3.4. Similar considerations are discussed further in Section 3.5.1.

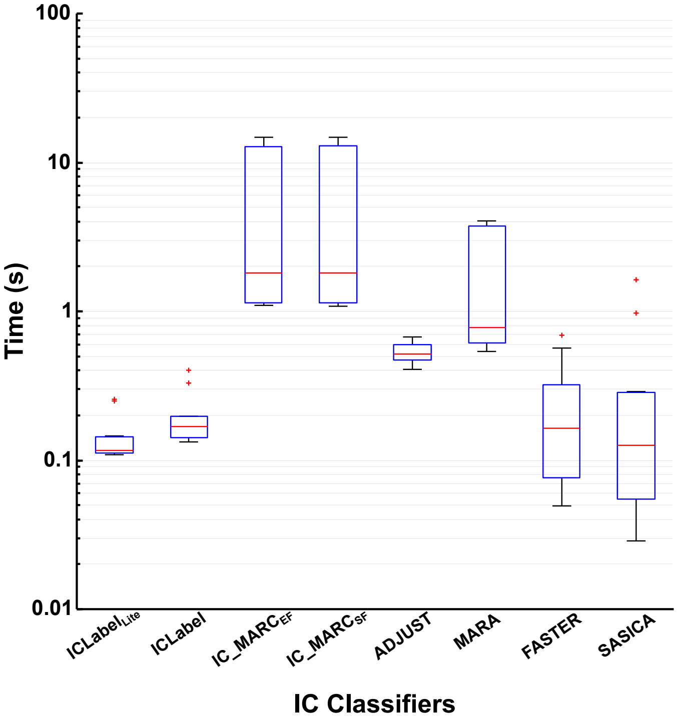


Figure 3.6: Time required to label a single IC, shown in logarithmic scale. Red lines indicate median time. Blue boxes denote the 25th and 75th percentiles, respectively. Whiskers show the most extreme values, excluding outliers which are denoted as small, red plus signs.

bel required median run times of 120 ms and 170 ms respectively. These were (median) 15.5 and 13.0 times faster than IC_MARC, respectively, and for single dataset averages up to a maximum of 88 and 64 times and a minimum of 9.8 and 6.7 times faster, respectively. Median IC classification speed for ICLabel$_{\text{Lite}}$ was 1.36 times faster than ICLabel, the difference required entirely due to the time taken to calculate the autocorrelation feature set. Details on the equipment used are provided at the end of Appendix 3.A.

3.4.3 Expert performance

As each IC in the ICLabel expert-labeled test set has been labeled by six experts, the opportunity exists to estimate the expected reliability of expert IC classifications. Table 3.2 shows the result of five such measures. The first three rows summarize how well each expert's classifications align with those of other experts and the last two rows summarize how well each expert's classifications align with those of the reference labels estimated with CL-LDA. Further descriptions of these measures are available in Appendix 3.A. These measures show that the agreement between experts is lower than one might expect with the optimistic approximation of agreement between experts being only 77% on average. By comparison, the agreement between experts and the CL-LDA-computed reference labels are always greater than or equal to those between experts.

3.5 Discussion

3.5.1 Using compositional IC classifications

Compositional labels like those produced by ICLabel may be used in multiple ways. When a single, discrete label is required, as is typical for multi-class classification, compositional labels may be summarized by the category with maximal probability. When such an approach is

Table 3.2: Measures of agreement both among experts and between experts and CL-LDA-computed reference. Measure descriptions are given in Appendix 3.A.

Measures	Experts						Mean
	A	B	C	D	E	F	
Inter-expert correlation	0.61	0.63	0.62	0.65	0.63	0.46	0.60
Inter-expert agreement (optimistic)	0.77	0.78	0.80	0.81	0.83	0.64	0.77
Inter-expert agreement (pessimistic)	0.55	0.57	0.55	0.58	0.55	0.46	0.54
Reference label correlation	0.82	0.84	0.82	0.81	0.78	0.60	0.78
Reference label agreement (optimistic)	0.86	0.86	0.92	0.85	0.87	0.64	0.83

Table 3.3: Independent component (IC) category detection thresholds for multi-label classification under various conditions. Each set of thresholds was determined by selecting class-specific thresholds that maximized the specified metric on the specified datasets.

Classifier	Dataset	Metric	Brain	Muscle	Eye	Heart	L.N.	C.N.	Other
ICLabel	Train	F_1	0.40	0.18	0.13	0.33	0.04	0.10	0.12
ICLabel	Train	Acc.	0.44	0.18	0.13	0.33	0.04	0.13	0.15
ICLabel	Test	F_1	0.14	0.29	0.04	0.03	0.84	0.05	0.26
ICLabel	Test	Acc.	0.35	0.30	0.04	0.03	0.84	0.05	0.26
$\text{ICLabel}_{\text{Lite}}$	Train	F_1	0.39	0.16	0.18	0.44	0.05	0.08	0.11
$\text{ICLabel}_{\text{Lite}}$	Train	Acc.	0.49	0.16	0.18	0.44	0.06	0.08	0.17
$\text{ICLabel}_{\text{Lite}}$	Test	F_1	0.05	0.04	0.06	0.10	0.42	0.02	0.29
$\text{ICLabel}_{\text{Lite}}$	Test	Acc.	0.53	0.17	0.06	0.10	0.42	0.15	0.29

F_1: F_1 Score; Acc.: Accuracy; L.N.: Line Noise; C.N.: Channel Noise taken, the value of the maximal probability can be interpreted as a measure of classifier confidence in the discrete classification. If the classification problem can be generalized to one of multi-label classification [Tsoumakas and Katakis, 2007], where each IC category is detected independent of other IC categories, each IC can be associated with zero or more different categorizations. In this case, class-specific thresholds can be applied to each IC category individually. This method can leverage ROC curves to estimate optimal class-specific thresholds. The estimated optimal thresholds from the ICLabel training set and expert-labeled test set were determined by taking the point on each ROC curve with either maximal F_1 score or accuracy and are shown in Table 3.3. Any element in a compositional IC label vector that matches or exceeds the corresponding threshold leads to a positive detection of the matching IC category. For example, using the thresholds determined from training set accuracy, if the ICLabel classifier produces an IC label vector $\begin{bmatrix} 0.71 & 0.04 & 0.03 & 0.01 & 0.01 & 0.02 & 0.18 \end{bmatrix}$, then the resulting detected labels would be $\begin{Bmatrix} \text{Brain}, & \text{Other} \end{Bmatrix}$because $0.71 > 0.44$ and $0.18 > 0.15$. By comparison, when applying the multi-class classification approach of selecting the class with maximal associated label probability, the implicit threshold for detection could be any value between that of the maximum class probability and that of the next most probable class. Because of this variable threshold, which is effectively different for every example classified, classifier performance for discrete labels is harder to quantify using ROC curves, as each point on the curve is potentially relevant to classifier performance. In the multi-label case, ROC curves provide a direct performance estimate; when a single threshold is chosen, the classifier is reduced to a single point on the ROC curve and, therefore, has a single performance value in terms of TPR and FPR as defined in Appendix 3.A. While multi-label classification is more flexible than multi-class classification, it allows for two possibly awkward outcomes: ICs with no IC category, and ICs with multiple IC categories. Depending on the use case, these outcomes may or may not be acceptable.

Compositional labels may also be used qualitatively to inform manual inspection. Compositional labels are more informative and easier to learn from than simple class labels [Hinton et al., 2015]. They are also helpful for recognizing clearly mixed components by (1) showing which category is most likely applicable to an IC while also (2) indicating other IC types the component in question resembles. Compositional labels are also more informative in cases of classification error, by showing which other categories may be correct if the most probable one is not. While direct use of the compositional labels retains the most information provided by ICLabel, compositional labels may also be difficult to use in an automated fashion.

3.5.2 Timing

The speed of ICLabel feature extraction and inference theoretically allows the classifier to be used in online, near-real-time applications. Even though ICLabel$_{\text{Lite}}$ was typically 36% faster than ICLabel, the average difference in calculation time per IC was only 50 ms. ICLabel is therefore sufficiently efficient for near-real-time use in most cases. A further consideration is that the times shown in Figure 3.6 are based on features extracted from the entirety of each EEG recording. Those PSD and autocorrelation estimates are non-causal and thus impossible to actualize in the case of real-time applications. Instead, those features are best estimated using recursive updates that not only fix the issue of causality, but may also spread the computational cost of feature extraction across time. By comparison, the proposed paradigm in Frølich et al. [2015] consisted of offline ICA decompositions of three-minute data segments at three-minute intervals, providing for intermittently-updated solutions with delays of six minutes. Also, these times were provided with the explicit assumption of heavily parallelized computation.

An online application for ICLabel is in the Real-time EEG Source-mapping Toolbox (REST) which implements an automated pipeline for near-real-time EEG data preprocessing and ICA decomposition using online recursive ICA (ORICA) [Hsu et al., 2016] and is described further in Chapters 5 and 6. REST can apply an IC classifier in near-real-time to the ORICA-decomposed EEG data, either to select ICs of interest or reject specified IC categories. The retained ICs can be used to reconstruct a cleaned version of the EEG channel data in near-real-time.

3.5.3 Differences between training set and test set results

ICLabel achieved higher scores on the cross-validated training data than on the expert-labeled test set. This could have occurred for three possible reasons: (1) overfitting to the ICLabel training set, (2) differing labeling patterns between the crowdsourced training set and the expert-labeled test set, and (3) high variance in expert-labeled dataset performance measures owing to the relatively small size of that dataset (130 ICs) and relatively few designated expert labelers (6). Overfitting during training (1) is unlikely to have played a major role due to the combined use of early stopping and cross-validation [Amari et al., 1997] but factors (2) and (3) could both be contributing factors. To resolve either problem would require more labeled examples, especially examples labeled by experts [Della Penna and Reid, 2012], a solution that is neither unexpected nor cheap. As more labels are submitted to the ICLabel website over time, these questions will become resolvable.

3.5.4 Cautions

As the primary purpose of an IC classifier is to enable automated component labeling, there is an implied trust in the results provided by that classifier. If the labels provided are incorrect, all further results derived from those labels are jeopardized. While the ICLabel classifier has been shown to generally provide high-quality IC labels, it is also important to be aware of its limitations, many of which are likely shared by other existing IC classifiers.

The accuracy of the ICLabel classifier, like that of any classifier using a sufficiently powerful model, is primarily limited by the data used to learn the model parameters. While the ICLabel training set is large and contains examples of ICs from many types of experiments, amplifiers, electrode montages, and other important variables which affect EEG recordings, the dataset does not contain examples of all types of EEG data. Infants, for example, are a population missing from the ICLabel dataset. As infant EEG can differ greatly from that of adults, spatially and temporally [Stroganova et al., 1999, Marshall et al., 2002], the results shown in Section 3.4.1 may not generalize to infant EEG. This issue was specifically raised by a user of the beta version of the ICLabel classifier who had anecdotal evidence of subpar performance when classifying Brain ICs in EEG datasets recorded from infants. While this is currently the only reported case of a possible structural failing of the classifier, more may exist relating to any other population of subjects or particular recording setting which is not sufficiently represented in the ICLabel dataset. Another likely source of datasets for which the ICLabel classifier could be unprepared is subjects with major brain pathology (brain tumor, open head injury, etc.). While recordings from subjects with epilepsy and children with attention deficit hyperactive disorder (ADHD) and autism are included in the ICLabel dataset, subjects with other conditions which might affect EEG may not be represented.

Another concern is the quality of the electrode location data used to create the IC scalp topographies. Ideally EEG data should be accompanied by precise 3D electrode location data (now obtainable at low cost from 3D head images [Lee and Makeig, 2018]), but the ICLabel dataset included some recordings that provided only template electrode location data, giving no simple means of controlling for localization error. All this variability should pose a challenge to training an IC classifier based on the IC scalp topographies. However, the broad source projection patterns inherent to scalp EEG mean that a scalp topography will vary relatively little when noise is added to the electrode positions used to compute it. Also, training on such a large number of IC scalp topographies should further moderate the effects of such electrode position error in the data.

3.5.5 An evolving classifier

The ICLabel project has the capacity to continue growing autonomously. Over time, as more suggested labels are submitted to the ICLabel website, automated scripts can perform the necessary actions of estimating “true” labels using CL-LDA, training a new version of the ICLabel classifier, and publishing the new weights to the EEGLAB plug-in repository. To maintain consistency, there should then be three versions of the ICLabel classifier available in the EEGLAB plug-in: the automatically-updated classifier, the classifier validated here, and the early version of the classifier released to the public prior to publication of this article (ICLabel$_{\text{Beta}}$). While the individual segments of such a pipeline already exist, the overall automation is not yet in place and is therefore left as a future direction for the project.

3.6 Conclusion

The ICLabel classifier is a new EEG independent component (IC) classifier that was shown, in a systematic comparison with other publicly available EEG IC classifiers, to perform better or comparably to the current state of the art while requiring roughly one tenth the compute time. This classifier estimates IC classifications as compositional vectors across seven IC categories. The speed with which it classifies components allows for the possibility of detailed, near-real-time classification of online-decomposed EEG data. The architecture and training paradigm of the ICLabel classifier were selected through a cross-validated comparison between six candidate versions. A key component of the greater ICLabel project is the ICLabel website (https://iclabel.ucsd.edu/tutorial) which collects submitted classifications from EEG researchers around the world to label a growing subset of the ICLabel training set. The evolving ICLabel dataset of anonymized IC features is available at https://github.com/lucapton/ICLabel-Dataset. The ICLabel classifier is available for download through the EEGLAB extension manager and from https://github.com/sccn/ICLabel.

3.7 Acknowledgements

Chapter 3, in part, has been submitted for publication of the material as it may appear in Luca Pion-Tonachini, Ken Kreutz-Delgado, and Scott Makeig. ICLabel: An automated electroencephalographic independent component classifier, dataset, and website. *NeuroImage (under review)*, 2019a. The dissertation author was the primary investigator of this paper. The expert-labeled test set was annotated with help from James Desjardins, Agatha Lenartowicz, Thea Radüntz, Lawrence Ward and Elizabeth Blundon, and Matthew Wisniewski. Their contributions are greatly appreciated. Thanks also to Francesco Marini for editorial comments. This work was supported in part by a gift from the Swartz Foundation (Old Field NY), by grants from the National Science Foundation under grant number GRFP DGE-1144086 and the National Institutes of Health under grant number 2R01-NS047293-14A1, and by a contract with Oculus VR, LLC. Nvidia Corporation donated a Tesla K40 GPU through its GPU Grant Program which was used to efficiently train all artificial neural network models.

3.A Evaluation Metrics

Balanced accuracy, an average of within-class accuracies (within-class recall), is defined as

$$\frac{1}{C}\sum_{i=1}^{C}\frac{\mathrm{TP}_i}{\mathrm{TP}_i+\mathrm{FN}_i}$$

where C is the number of distinct classes and TP_i is the number of true positive detections, the number of correct classifications of examples into a specific class, for class i and FN_i is the number of false negatives errors, the number of incorrect classifications of examples into any class other than the specific class, for class i. Although TP and FN are values that are typically calculated for binary classification, they can be easily adapted to the multi-class case by selecting one class as the “positive” class and combining all other classes into the “negative” class. In this way, TP_i is the number of correct classifications of examples into class i and FN is the number of incorrect classifications of examples from class i into any other class.

Cross entropy is a measure that can be interpreted as the negative data log-likelihood if labels are assumed to be categorically distributed or alternatively as the portion of the Kullback–Leibler divergence that depends on predicted values. More pertinently, cross entropy was the primary metric optimized while training the ICLabel candidate classifiers, though it was modified for both the wCNN and GAN paradigms. Cross entropy over an entire dataset is defined as

$$\sum_{n=1}^{N}\sum_{i=1}^{C} t_i^n \log p_i^n$$

where N is the number of data-points and t_i^n and p_i^n are the i^{th} elements in the "true" and predicted probabilistic label vectors, respectively, for the n^{th} IC.

The receiver operating characteristic (ROC) curve shows the changing performance of a binary classifier as the threshold for detection of the positive class is varied from zero to one by plotting false positive rate (FPR) against true positive rate (TPR) on the horizontal and vertical axes, respectively. TPR, also known as sensitivity or recall, is defined as $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$ which is the ratio of TP to total samples in the positive class. FPR is defined as $\mathrm{FP}/(\mathrm{FP}+\mathrm{TN})$ where FP is the number of false positive errors, the number of incorrect classifications of examples into the positive class; TN is the number of true negative detections, that is, the number of correct classifications of examples into the negative class. FPR can also be defined as $1-\text{specificity}$ where specificity is $\mathrm{TN}/(\mathrm{FP}+\mathrm{TN})$. As was explained for balanced accuracy, one way ROC curves can be adapted to the multi-class case is by selecting a single class as the positive class and treating the combination of all other classes as the negative class. The ROC curve for the i^{th} class is a function of a threshold detection parameter $\theta \in [0,1]$ and is defined as the parametric function

$$(\mathrm{FPR}_i(\theta), \mathrm{TPR}_i(\theta)) = \begin{cases} \mathrm{TPR}_i(\theta) = \frac{\sum_{=1}^{N} \chi\left(p_i^n \geq \theta\right) \chi\left(\arg\max_k \ t_k^n = i\right)}{\sum_{n=1}^{N} \chi\left(\arg\max_k \ t_k^n = i\right)} \\ \mathrm{FPR}_i(\theta) = \frac{\sum_{n=1}^{N} \chi\left(p_i^n \geq \theta\right) \chi\left(\arg\max_k \ t_k^n \neq i\right)}{\sum_{n=1}^{N} \chi\left(\arg\max_k \ t_k^n \neq i\right)} \end{cases} \qquad \theta \in [0,1]$$

where $\chi(\cdot)$ is the indicator function defined as

$$\chi(\texttt{condition}) = \begin{cases} 1 & \text{if } \texttt{condition} \text{ is true} \\ 0 & \text{if } \texttt{condition} \text{ is false} \end{cases} .$$

When comparing threshold-dependent classifier performance on the ROC curve, ideal classifiers reside in the top left corner while a chance-level classifier resides along the diagonal connecting the bottom left and top right corners (see Figures 3.4 and 3.8). To aid in visual recognition of better curves, F_1 score isometrics are plotted that denote all point in the performance plane with equal F_1 score (higher value is better). The F_1 score is the harmonic average of recall and precision where precision is $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$ and the harmonic average of x and y is $1/((1/x)+(1/y)) = (xy)/(x+y)$. The F_1 score is convenient as it rewards reasonable compromises between precision and recall with higher values. For the experiments described earlier in this section, ROC curves are calculated for each IC category individually.

Confusion matrices provide a matrix representation of the quantity and type of correct and incorrect classifications a classifier makes on a given dataset. As also explained in Appendix 3.D, each row is associated with a specific IC category determined through the crowd labeling effort, while each column is associated with a specific IC category as predicted by the classifier. Normally, the categories are in the same order for both the rows and the columns and therefore the diagonal elements are associated with true positive detections while the off-diagonal elements are associated with errors. Normalized confusion matrices constrain the elements of each row to sum to 1 by dividing those elements by the total number of examples of each IC category. Mathematically, the elements of a normalized confusion matrix may be computed as

$$\mathrm{CM}_{ij} = \frac{\sum_{n=1}^{N} \chi\left(\arg\max_k t_k^n = i\right) \chi\left(\arg\max_k p_k^n = j\right)}{\sum_{n=1}^{N} \chi\left(\arg\max_k t_k^n = i\right)}$$

where CM_{ij} is the element in the i^{th} row and the j^{th} column of the confusion matrix.

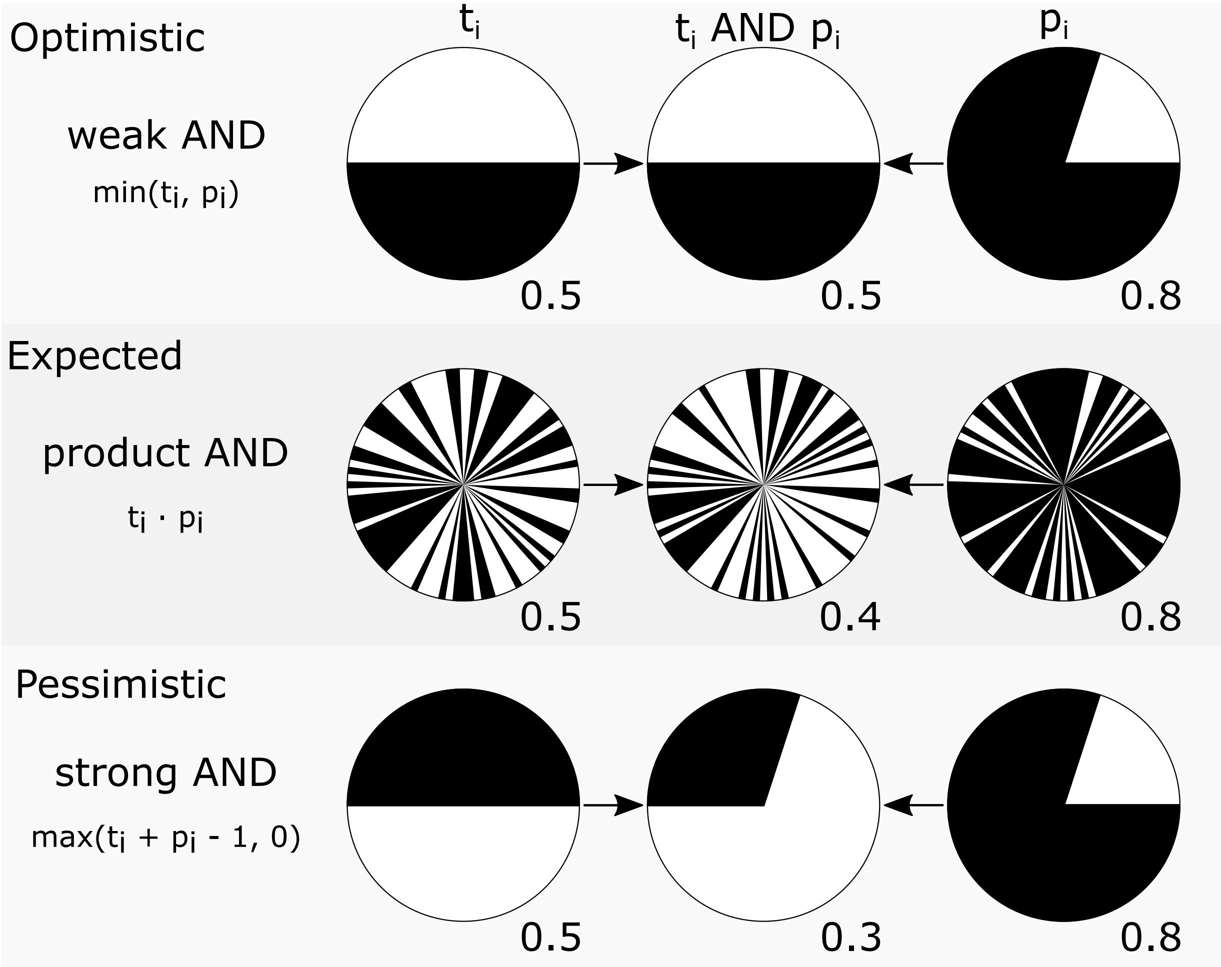


Figure 3.7: Visualization of three soft AND functions with which Boolean AND could be replaced for evaluating agreement between soft or compositional labels. The second and fourth columns from the left show how the reference and predicted class memberships (in black) might be distributed in a pie chart and the third row shows the resulting value of the Boolean AND of these soft-AND-related representative arrangements. Strong AND corresponds to the assumption of worst-case (least) overlap of actual and predicted labels; expected AND corresponds to a uniform and independent distribution of actual and predicted labels; and weak AND corresponds to the best-case (most) overlap of actual and predicted labels. This figure is modified after Figure 2 in Beleites et al. [2013].

Soft confusion matrix estimates account for the ambiguity of how soft labels and predictions might agree or differ [Beleites et al., 2013]. Rather than discretizing reference labels and predictions before counting how many match using the Boolean AND function, defined as

$$\mathrm{AND}(x,y) = \begin{cases} 1 & \text{if } x = y = 1 \\ 0 & \text{otherwise} \end{cases} \quad x,y \in \{0,1\},$$

as for traditional confusion matrices, soft confusion matrices operate directly on continuous-valued soft label vectors and therefore require a different but comparable soft AND function for comparison. The aforementioned ambiguity in comparing soft labels arises from the various possible functions with which that comparison can be made. For example, assuming an IC contains activity from both the brain and line noise in equal proportions (i.e., 50% “Brain” and 50% “Line Noise”, perhaps arising when the line noise activity was spatially nonstationary and therefore difficult to isolate through ICA decomposition), and that a classifier predicts that the IC is 20% “Brain” and 80% “Line Noise”, three possible soft AND functions that can be used for comparison (strong AND, product AND, and weak AND) are detailed in Figure 3.7. From an optimistic perspective, the “Line Noise”-related agreement could be measured as the minimum of the two “Line Noise”-related labels (weak AND) resulting in 50% agreement as shown in the right-most column of Figure 3.7. Alternatively the prediction of 80% “Line Noise” could have been wrongly based upon evidence originating from the brain-related aspects of the IC activity, therefore leaving only 30% of the prediction being correctly derived from line-noise-related evidence. This pessimistic interpretation leads to the same result and interpretation as strong AND as shown in the second column from the left in Figure 3.7. Weak AND and strong AND functions act as bounds on the possible ways that the labels and predictions conform and the actual agreement between label and prediction can be any value between those two, but assuming a uniformly distributed mapping of evidence to classifier prediction, the result would be 40%

agreement. This interpretation is associated with the product AND function and a visualization of such a uniform distribution of class-membership can be seen in the second column from the right in Figure 3.7. This example is adapted from the cancer tissue example in Section 2.2 of Beleites et al. [2013], wherein this topic is more thoroughly explored.

From these three continuous-valued replacements for the Boolean AND function, three different confusion matrices corresponding to pessimistic, expected, and optimistic estimates can be computed. These matrices can be combined to form pseudo-confidence intervals for elements of the soft confusion matrices and many of the statistics derived therefrom. Provided this fact, an equivalent to ROC curves, termed soft operating characteristic (SOC) points, may be computed by applying the TPR and FPR equations to the soft confusion matrices. As there is no discretization of the prediction in the soft case, the soft version of a class-specific ROC curve is only a single point per soft confusion matrix resulting in three total points in the performance plane per classifier and class. Following from the natural ordering of the strong, product, and weak AND functions, the three points making up each SOC are also ordered and are therefore connected by lines to show this relationship. Although soft-TPR and soft-FPR can be plotted on the same axes as classical ROC curves, the values along those the classical curves and the values derived from the soft confusion matrices are not directly comparable due to the conflicting assumptions guiding how each confusion matrix is calculated.

The conclusion of Beleites et al. [2013] lists four reason why a study might use soft confusion matrix statistics in place of the more commonly used statistics; these reasons are summarized here:

1. Label discretization, or “hardening”, leads to overestimating class separability.
2. Estimating ambiguous labels may be a part of the goal for the predictor.
3. Hardening explicitly disregards information present in the probabilistic labels.
4. Hardening increases label variance when trying to learn smooth transitions between classes.

Here, both ROC curves and SOC points are presented as the relevance of each measure depends on the intended application of a classifier.

IC classification speed was measured in terms of the time to extract features from and classify a single IC as measured by the MATLAB functions `tic` and `toc`. The publicly available implementations of each classifiers was run, one dataset at a time, and the total calculation time for each dataset was divided by the number of ICs present in that dataset. This was repeated for all 10 datasets in the expert-labeled test set. Computations were performed in MATLAB 2013a, with no specified parallelization of calculations, running in Fedora 28 using an AMD Opteron 6238 processor operating at 2.6 GHz.

Expert performance metrics listed in Table 3.2 are defined as follows:

- "Inter-expert correlation" is the mean correlation between an expert's classifications and those of other experts.

- "Inter-expert agreement (optimistic)" is the proportion of ICs for which an expert assigned at least one IC category in common with another expert, averaged across other experts.

- "Inter-expert agreement (pessimistic)" is the proportion of ICs for which an expert assigned all IC category in common with another expert, averaged across other experts.

- "Reference label correlation" is the correlation between an expert's classifications and the reference labels.

- "Reference label agreement (optimistic)" is the proportion of ICs for which an expert assigned the IC category to an IC which was most probably according to the reference labels.

3.B Generative Adversarial Networks

Generative adversarial networks (GAN) vie two competing artificial neural networks (ANN) against each other wherein one attempts to generate simulated data (generator network) and the other attempts to discern whether data is simulated or real data (discriminator network). Typically, GANs are trained in an a two-stage iterative fashion where in the first stage the generator network transforms random noise into simulated examples that the discriminator network classifies as either "real" or "fake". The generator network parameters are updated to make the discriminator more likely to label the generated examples as "real". In the second stage, the discriminator labels another set of generated sample as well as actual collected samples. The discriminator network parameters are then updated to make the discriminator network more likely to label the generated samples as "fake" and the actual samples as "real". These two stages are repeated until predetermined convergence criteria are achieved.

For SSGANs, instead of the discriminator network deciding between just real and simulated data, the "real" category is subdivided into multiple classes such as "Brain", "Eye", and "Other". The model used for the ICLabel classifier extended the SSGAN model to have multiple generator networks; one for each feature set used to describe ICs, that all shared the same random-noise input. As a final output, the SSGAN produced an eight-element compositional vector comprised of relative pseudo-probabilities for the seven IC categories described in Section 3.2.1 and that of the IC being produced by the generator network. Regarding classification, the last element can easily be ignored by removing it and renormalizing the remaining seven-element vector to sum to one.

SSGANs have been shown to improve classification performance over CNNs when there are few labeled examples, provided there are more unlabeled examples available [Odena, 2016, Salimans et al., 2016]. It has been theorized that the additional task of determining whether an example is real or generated helps the network to learn intermediate features helpful for classifying the examples into the categories of interest as well as discriminating actual from simulated ICs [Odena, 2016, Salimans et al., 2016]. Others theorize that GANs help with classification when they generate low-probability examples that may be hard to find actual examples of in collected datasets. These low-probability examples help the network learn where the decision boundaries should be placed in the potentially large space between some classes [Dai et al., 2017, Lee et al., 2018], similar to the concept motivating maximum-margin classifiers like support vector machines. The training paradigms in Dai et al. [2017], Lee et al. [2018], and Srivastava et al. [2017] were also attempted, but those results are omitted as they did not differ greatly from the modified SSGAN results shown in Appendix 3.C.

3.C ICLabel Candidate Classifier Selection

As described in Section 3.3.3, six candidate IC classifiers were created in three-by-two factorial design to compare classification performance across three model architectures and training paradigms and two different collections of features provided to the candidate classifiers. These were measured using a ten-fold cross-validation scheme on the ICLabel training set.

Regarding the first factor, model architecture and training paradigm, comparing ROC curves reveals that the GAN-based ICLabel candidates underperformed when compared to the other candidate models. This is visible across all seven classes in the ROC curves and most classes in the SOC points as presented in Figure 3.8. The exceptions for SOC points were “Channel Noise” components, where the GAN methods scored highest on the soft measures, and Brain ICs and Eye ICs for which the GAN and unweighted CNN models performed similarly. While consistent, minor differences between wCNN and CNN models exist in the ROC curves, as shown for Other ICs and Chan Noise ICs, stronger differences are indicated by the SOC points where wCNN models notably outperformed CNN models. The wCNN models displayed better pessimistic and expected SOC performance over all classes as well as the best optimistic performance for Muscle

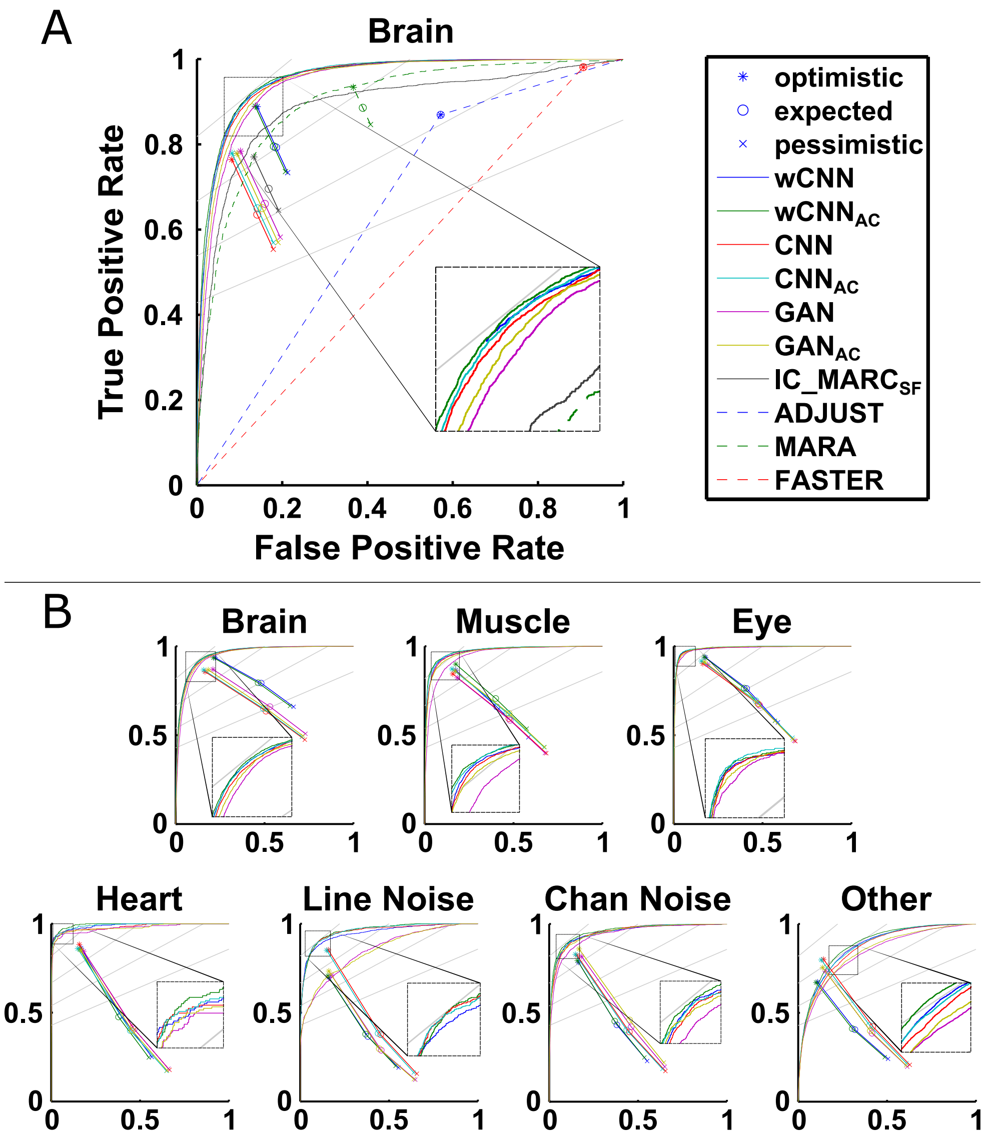


Figure 3.8: Color-coded ROC curves and soft operating characteristics (SOC) points calculated from soft confusion matrices to quantify IC classification performance on the cross-validated training data. The colors indicate the performances of the various candidate classifiers under consideration (see Sections 3.3.3 and 3.2.2 for the description of these classifiers). Part A of this figure contains the results merged into two classes, “Brain” and “Other”, while part B contains the results across all seven ICLabel IC categories. The large dashed black squares show magnified views of the smaller dashed black squares. Gray lines indicate F_1 score isometrics of 0.9, 0.8, 0.7, and 0.6 from top to bottom. Refer to Appendix 3.A for definitions of F_1 score, ROC curves, and SOC points. The best performing candidate architecture was consistently shown to be wCNN_{AC}. The worst performing candidate architectures were those based on generative adversarial networks.

ICs and Eye ICs. Despite exceptions in the case of Line Noise ICs and Other ICs, where the optimistic SOC points favored CNN models, the results generally favored wCNN models over CNN models.

For the second factor, feature sets provided to the candidate classifiers, the inclusion of autocorrelation as a feature set appeared to consistently improve performance across all classes. This was especially true for Muscle ICs and Other ICs, as evidenced by nearly uniform improvement measures by ROC curves and SOC points.

With these three findings, the official ICLabel classifier was trained using the wCNN_{AC} paradigm and is referred to simply as ICLabel. This new model underwent comparison against published IC classification methods and, eventually, was publicly released as an EEGLAB plug-in. Because the autocorrelation feature set requires additional time to calculate, another model based on the wCNN paradigm was also compared with published IC classification methods for situations when faster feature extraction time is imperative. This new wCNN-based model is referred to as $\text{ICLabel}_{\text{Lite}}$.

3.D CL-LDA Details and Hyperparameters

While reference labels (estimated "true labels") are the desired output for the purposes of training the ICLabel classifier, CL-LDA also simultaneously calculates estimates of labelers' skill, parameterized by a confusion matrix. For the ICLabel dataset, these confusion matrices take the form of seven-by-eight matrices where each row is associated with one of the seven IC categories mentioned in Section 3.2.1 and each column is associated with one of the eight possible responses allowed on the ICLabel website: the seven IC categories and "?". Each row of the confusion matrix can be interpreted as the estimated probabilities of the labeler providing each response conditioned on the IC in question being of that row's associated IC category. A perfect labeler would have ones in the entries for matching IC categories and responses, such as the intersection of the “Brain” response column and the “Brain” IC row, and zeros in the entries for mismatching IC categories and responses, such as the intersection of the “Eye” IC response column and the “Brain” IC row. These matrices start with prescribed values dependent on prior assumptions; but as labelers submit more labels, the labeler skill matrices become more dependent upon the submitted labels rather than those prior assumptions.

CL-LDA efficiently estimates model parameters by maintaining counts of how each labeler labels examples from each IC category. In this way, priors on the labeler matrices can be interpreted as pseudo-counts that add their value to the actual, empirical counts tracked by CL-LDA. Compositional label estimates are formed by CL-LDA in much the same way using a weighted count of how labelers associate an IC with each IC category. Just as with the labeler priors, the class priors add pseudo-counts to the empirical counts for each IC. Refer to Chapter 2 for more details. An implementation of CL-LDA can be found at https://github.com/lucapton/crowd_labeling.

Certain labelers were manually marked as “known experts” when the ICLabel website database was created while the rest were treated as labelers of unknown skill. The experts were assigned a favorable and strong prior distribution for their confusion matrix parameters while the labelers of unknown skill were assigned a favorable and weak prior distribution of their confusion-matrix parameters. Strong and weak priors correspond to how many submitted labels are necessary to overcome that prior’s influence; strong requiring more and weak fewer. Explicit priors used in this work are provided below. To maintain an acceptable level of quality for labeler skill estimates, only labels from labelers who submitted ten or more labels were considered. If this requirement were not in place, there would be many votes included by users who submitted fewer labels and very little could be known regarding their abilities.

The prior for expert confusion matrices was

$$\begin{bmatrix} 50.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 50.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 50.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 50.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 50.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 50.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 50.01 & 0.01 \end{bmatrix}$$

while the confusion matrix prior for labelers of unknown skill was

$$\begin{bmatrix} 1.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 1.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 1.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 1.25 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 & 1.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 1.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 1.25 & 0.25 \end{bmatrix}.$$

Class priors were approximately

$$\begin{bmatrix} 0.002973 & 0.001766 & 0.00079 & 0.00015 & 0.000573 & 0.00073 & 0.003022 \end{bmatrix}.$$

The class priors were set as the empirically-determined class prior probabilities divided by 100 and are ordered following the same IC category ordering of the labeler confusion matrices. The burn-in period for the CL-LDA Gibbs sampler was 200 epochs over the data and the labels were estimated over the next 800 epochs.

To estimate labels for the expert-labeled test data, CL-LDA was applied to the collected expert labels on the test set using the same procedure as was used for the training set. The prior for expert confusion matrices was

$$\begin{bmatrix} 5 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 5 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 5 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 5 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 5 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 5 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 5 & 0.01 \end{bmatrix}.$$

and class priors were approximately

$$\begin{bmatrix} 0.002263 & 0.001537 & 0.001753 & 0.000155 & 0.00063 & 0.001839 & 0.001822 \end{bmatrix}.$$

3.E Artificial Neural Network Architecture Details

The ICLabel candidate and final classifiers were each composed of individual neural networks for each feature set, the outputs of which were concatenated and fed into another network to produce the final classifications. Specifically, the IC scalp topographies were fed into a two-dimensional CNN using dilated convolutions. One-dimensional CNNs were used for all other features (PSD and/or autocorrelation). Scalp topography images were 32-pixels-by-32-pixels with one intensity channel. Both PSD and autocorrelation features sets were 100-element vectors. Scalp topographies and PSDs were scaled such that the maximum absolute value for each one was 0.99. Autocorrelation vectors were normalized such that the zero-lag value was 0.99 before removal. The discriminator and classifier scalp topography subnetworks were comprised of three convolutional layers while the PSD and autocorrelation subnetworks had three one-dimensional convolutional layers. The three generator subnetworks were comprised of four transposed convolutional layers each. As input, they took a shared 100-element vector of Gaussian noise with mean zero and a variance of one. This architecture was loosely based upon that of DCGAN [Radford et al., 2015]. Details on the layers used in these architectures are shown in Table 3.4 where "Topo" is used as shorthand for scalp topography and "AFC" for autocorrelation function. CNN and wCNN architectures only used layers in the "Classifier" network, while GAN-based classifiers used all listed layers during training and only used "Classifier" networks layers for inference. Classifier layer "Final" used seven filters for both CNN and wCNN architectures while GAN-based classifiers used eight filters during training and seven during inference by removing the filter for detecting IC features created by the generator networks. GAN-based classifiers applied a binary mask to the output of the scalp topography generator network setting peripheral pixels to zero to match the interpolation format of actual scalp topographies.

Training of the candidate and official models was accomplished using Adam [Kingma and Ba, 2014] with a learning rate of 0.0003, β_1 of 0.5, and β_2 of 0.999 to calculate parameter updates with a gradient cutoff of 20 and a batch size of 128 ICs. Labeled examples for each batch were selected with random class-balanced sampling to overcome class imbalances in the ICLabel training set. Holdout-based early stopping with a viewing window of 5,000 batches was used as a convergence condition to mitigate overfitting [Prechelt, 2012]. All architectures used input noise [Sønderby et al., 2016] to stabilize convergence. Batch normalization [Ioffe and Szegedy, 2015] was used only in the generator network from the GAN-based architecture. The GAN-based classifiers also used one-sided label smoothing [Salimans et al., 2016].

The ICLabel training set was augmented to exploit symmetries in scalp topographies through left–right reflections of the IC scalp topographies as well as negations of the IC scalp topographies. Negation of the scalp topography exploits the fact that if one negates both the ICA mixing matrix as well as the IC time-courses, the resulting channel data remain unchanged.

Table 3.4: Layers used in ICLabel candidate classifier architectures. CNN and wCNN architectures only use layers in the "Classifier" network, while GAN-based classifiers use all listed layers during training despite only using "Classifier" networks layers during inference. Classifier layer "Final" uses seven filters for both CNN and wCNN architectures while GAN-based classifiers use eight filters during training and seven during inference by removing the filter related to generated samples. "Topo" is used as shorthand for "scalp topography" and "ACF" for "autocorrelation function". "ReLU" is short for "rectified linear unit" [Nair and Hinton, 2010], "LReLU" is short for "leaky ReLU" [Maas et al., 2013] with a leakage parameter of 0.2., and "tanh" is short for "hyperbolic tangent".

Network	Layer	Filters	Kernel	Stride	Padding	Activation
Classifier	Topo-1	128	4×4	2	same	LReLU
Classifier	Topo-2	256	4×4	2	same	LReLU
Classifier	Topo-3	512	4×4	2	same	LReLU
Classifier	PSD-1	128	3	2	same	LReLU
Classifier	PSD-2	256	3	2	same	LReLU
Classifier	PSD-3	1	3	2	same	LReLU
Classifier	ACF-1	128	3	2	same	LReLU
Classifier	ACF-2	256	3	2	same	LReLU
Classifier	ACF-3	1	3	2	same	LReLU
Classifier	Final	7 or 8	4×4	2	valid	SoftMax
Generator	Topo-1	2,000	4×4	2	valid	ReLU
Generator	Topo-2	1,000	4×4	2	valid	ReLU
Generator	Topo-3	500	4×4	2	valid	ReLU
Generator	Topo-4	1	4×4	2	valid	tanh
Generator	PSD-1	2,000	3	1	valid	ReLU
Generator	PSD-2	1,000	3	1	valid	ReLU
Generator	PSD-3	500	3	1	valid	ReLU
Generator	PSD-4	1	3	1	valid	tanh
Generator	ACF-1	2,000	3	1	valid	ReLU
Generator	ACF-2	1,000	3	1	valid	ReLU
Generator	ACF-3	500	3	1	valid	ReLU
Generator	ACF-4	1	3	1	valid	tanh

As negating the time courses does not affect any of the other feature sets used, only the scalp topographies need be altered. Horizontal reflections of the scalp topographies exploits the (near) symmetry of human physiology. One notable exception to this symmetry is the heart being located only on the left side of the chest. However, Heart ICs were comparatively rare in the training set and left–right reflection of Heart IC scalp topographies did not create confusion with an other IC class scalp topography. This effectively resulted in a four-fold increase in the number of ICs in the dataset.

All ICLabel candidate and official classifiers were built and trained in python using Tensorflow [Abadi et al., 2015]. They were also converted to MATLAB using matconvnet [Vedaldi and Lenc, 2015] for distribution as an EEGLAB plug-in. Files involved in training the ICLabel classifier can be found at https://github.com/lucapton/ICLabel-Train.

Chapter 4

Artifact Rejection

4.1 Introduction

A universal challenge that hinders the decoding and application of electroencephalographic (EEG) data is that EEG recordings are almost always contaminated by artifacts such as electrode impedance changes caused by headset motion as well as eye-blink, eye-movement, neck muscle, and scalp muscle activities. This is a problem both in the case of channel-level as well as source-level analysis. In both cases, artifacts can skew metrics and bias models which can every easily lead to spurious conclusions. When performing source analysis, artifacts often have the added disadvantage of hindering the quality of the independent component analysis (ICA) decompositions learned on the artifact-contaminated EEG data. In this chapter we asses the performance of the artifact rejection algorithm: artifact subspace reconstruction (ASR) [Kothe and Jung, 2014], with a special focus on how it affects ICA decompositions and the resultant components.

4.2 Background

Traditionally, these artifacts were removed manually by visual inspection [Jung et al., 2000a], which could be time-consuming, laborious, subjective, and incompatible with online and real-time applications [Urigüen and Garcia-Zapirain, 2015].

To automate the artifact removal process, earlier methods have used channel-based statistical thresholding approaches to remove abnormal activities [de Cheveigné, 2016, Jas et al., 2017] or adaptive filters with additional reference channels to regress out targeted artifacts [Noureddin et al., 2012]. Unfortunately, these methods either cannot reconstruct clean data from spatially outspread artifacts or require auxiliary channels for specific artifacts.

Another popular approach is to separate artifacts from brain-related signals using blind source separation (BSS), especially ICA [Jung et al., 2000a, Urigüen and Garcia-Zapirain, 2015]. Since BSS cannot identify components categories automatically, a classifier is needed to identify and reject the artifact-related components. Most independent component (IC) classifiers are pre-trained, do not adapt to new datasets, and are trained on a limited set of experimental data [Radüntz et al., 2017, Frølich et al., 2015, Winkler et al., 2014, Bigdely-Shamlo et al., 2013] or they require pre-recorded target-artifact sections [Zhang et al., 2015] or auxiliary channels [Guarnieri et al., 2018]. Moreover, the ICA-based methods are usually less effective in removing transient, non-biological artifacts such as abrupt impedance changes from headset motions since those artifacts are not typically separated into individual ICs. Table 4.1 summarizes the state-of-the-art automatic, online-capable artifact removal methods for multi-channel EEG recordings [Islam et al., 2016].

To address the challenges the above methods encountered, Kothe and Jung [2014] proposed ASR, which is an automatic, online-capable, component-based artifact removal method for removing transient or large-amplitude artifacts. ASR is similar to principal-component-analysis-based (PCA-based) methods in which large-variance components are rejected and channel data

Table 4.1: State-of-the-art automatic, online-capable artifact removal methods for multi-channel EEG measurement. **Clear definition is in de Cheveigné [2016]. **Some of the features for classification cannot be used in online classification. ***Functional Link Neural Network and Adaptive Neural Fuzzy Inference System.*

Author	Year	Method Category	Artifact Types	Reference	Toolbox
Guarnieri et al. [2018]	2018	ICA-REG	EOG	EOG	N/A
Radüntz et al. [2017]	2017	ICA	EOG, Heart	None	N/A
de Cheveigné [2016]	2016	Statistic	Channel Noise*	None	STAR
Zhang et al. [2015]	2015	wavelet ICA	EOG, EMG	None	N/A
Frølich et al. [2015]**	2015	ICA	EOG, EMG, Heart	None	IC_MARC
Hu et al. [2015]	2015	FLNN-ANFIS***	EOG, EMG	EOG, EMG	N/A
Kothe and Jung [2014]	**2014**	**PCA-statistic**	**Large-amplitude artifacts**	**None**	**ASR**
Winkler et al. [2014]	2014	ICA	EOG,EMG	None	MARA
Bigdely-Shamlo et al. [2013]	2013	ICA	EOG	None	EyeCatch
Noureddin et al. [2012]	2012	Adaptive Filter	EOG	Eye Tracker	N/A
Gao et al. [2010]	2010	CCA	EMG	None	N/A

are reconstructed from remaining components. The main difference is that ASR automatically identifies and utilizes clean portions of data as a reference to determine thresholds for rejecting components.

In the EEG community, there has been an increasing use of ASR as a powerful, automatic data-cleaning method [Artoni et al., 2017, Mullen et al., 2015]. However, the effectiveness of ASR and the guidelines for choosing its parameter(s) have not been carefully evaluated and reported, especially on real EEG data.

Continuing the previous study described in Chang et al. [2018], this study systematically evaluates the effectiveness of ASR on 20 EEG recordings from ten subjects performing a simulated driving experiment, where artifacts induced by EEG headset motions and activities from eye-blink, eye-movement, and head and neck muscles are present. We first characterize the performance of ASR with different cutoff parameters that determine the rejection thresholds. Next, we apply ICA and the ICLabel classifier from Chapter 3 to separate and automatically identify artifacts and brain signals to allow a quantitative assessment of ASR's effectiveness in removing various types of artifacts and preserving brain activities. Finally, we report cross-subject results and provide guidelines to optimally choose ASR's parameter.

4.3 Materials and Methods

4.3.1 Dataset and data preprocessing

Experiment and data collection

To evaluate ASR, we used 20 EEG recordings from ten subjects performing a sustained attention task in a driving simulator [Huang et al., 2009]. In the 90-minute experiments, subjects reacted to randomly occurring lane-departure events by steering the car back to the center of their lane. Therefore, there were intermittent artifacts in the EEG data from electrical interference, EEG headset motions, and activities from neck and scalp muscles, eye blinks, and eye movements. For each subject, 32-channel EEG data were recorded using a NeuroScan System at 500 Hz sampling rate. Wet electrodes (Ag/AgCl) were placed on the scalp following the international 10–20 system [Klem et al., 1999].

Data preprocessing

To remove high-frequency noise, the EEG data were cleaned using a band-pass FIR filter (0.5–100 Hz) and then were down-sampled to 250 Hz. Next, we used `clean_rawdata`, an EEGLAB plug-in function [Delorme and Makeig, 2004], to remove channels with negligible activity (flat line threshold: 5), noisy signals (noisy line threshold: 4), or a poor correlation with adjacent channels (correlation threshold: 0.8).

4.3.2 Artifact subspace reconstruction (ASR)

This section describes the ASR algorithm with emphasis on the key aspects and advantages of ASR. A more detailed description of ASR is available in Kothe and Jung [2014].

The underlying concept is that the data segment X_t can be decomposed into latent components S_t using the mixing matrix M_r: $X_t = M_r S_t$. Artifact rejection is performed in the principal component (PC) space $Y_t = V_t^T X_t = V_t^T M_r S_t$, and thus the clean latent components $(S_t)_{clean}$ can be reconstructed using the pseudoinverse of the truncated $V_t^T M_r$: $(S_t)_{clean} = (V_t^T M_r)^+_{trunc} Y_t = (V_t^T M_r)^+_{trunc} V_t^T X_t$ where A^+ is the pseudoinverse of A. Projecting $(S_t)_{clean}$ back to channel-space using M_r yields the cleaned data in Equation 4.1.

The ASR process consists of three steps: (1) extracting reference data from raw data, (2) determining thresholds for identifying artifact components, and (3) rejecting the artifact components and reconstructing the resulting data. Figure 4.1 shows an overview of the three steps. The concepts and implementation details are described as follows.

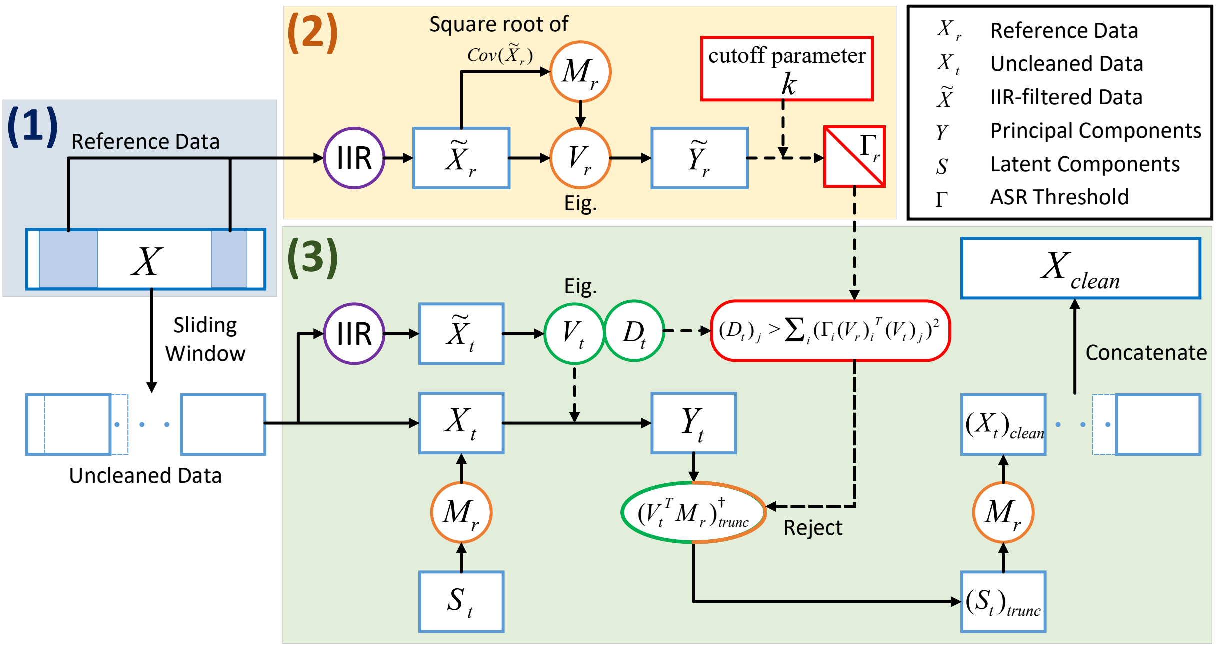


Figure 4.1: Flow chart describing artifact subspace reconstruction (ASR).

Extract reference data

ASR automatically selects clean portions of EEG data based on the distribution of signal variance. Specifically, ASR calculates channel-wise root-mean-square (RMS) values on 1-second windows, z-scores the values across all windows from each individual channel, identifies clean windows in which the z-scored values are within -3.5 and 5.5 [1], and concatenates the clean windows to obtain reference data X_r. A tolerance value, here we used 7.5%, is set to allow a small percentage of bad channels to remain in X_r otherwise the criteria for choosing X_r is too restrictive and there will not be enough reference data to calibrate ASR. It is worth noting that the length of the reference data found will vary with the noise level of the data.

Determine thresholds for identifying artifact components

ASR applies a carefully-designed IIR filter to the reference data X_r to suppress specific frequency-band activities typically associated with brain oscillations, obtaining $\tilde{X}_r$. ASR computes

[1] The RMS values are fit into a truncated Gaussian distribution.

the mixing matrix M_r, i.e., the square root of $\mathrm{Cov}(\widetilde{X}_r)$, and the eigenvalue decomposition of M_r to obtain the eigenvectors matrix V_r and eigenvalues vector D_r. Each column in V_r is the eigenvector corresponding to the eigenvalue in D_r. Once the data are projected onto the PC space $\widetilde{Y}_r = V_r^T \cdot \widetilde{X}_r$, ASR calculates the mean μ_i and standard deviation σ_i of RMS values across all 0.5-second windows of $\widetilde{Y}_r$ for each component i, and defines rejection thresholds $\Gamma_i = \mu_i + k \cdot \sigma_i$ where k is the user-defined cutoff parameter.

Reject artifact components and reconstruct cleaned data

ASR applies an eigenvalue decomposition to the covariance matrix taken across channels of the IIR-filtered uncleaned EEG segments $\mathrm{Cov}(\widetilde{X}_t) = V_t D_t V_t^T$ along a sliding window with a window size of 0.5 seconds and a step size of 0.25 seconds. The IIR filter here is the same as in step (2). For each window, ASR identifies whether j^{th} PC $(V_t)_j$ with variance $(D_t)_j$ is larger than the rejection thresholds Γ_i projected from V_r onto V_t: $(D_t)_j > \sum_i (\Gamma_i (V_r)_i{}^T (V_t)_j)^2$. If the inequality holds, then the values of that component's activities are replaced with zero vectors: $(V_t^T M_r)_{trunc}$. Finally, ASR reconstructs the cleaned data segment using the equation:

$$(X_t)_{clean} = M_r (V_t^T M_r)^{+}_{trunc} V_t^T X_t \tag{4.1}$$

The MATLAB scripts for performing ASR are available as an open-source plug-in function `clean_rawdata` in EEGLAB [Delorme and Makeig, 2004]. While many settings can be optimized, the most important user-defined parameter is the cutoff parameter k for determining the rejection thresholds in units of standard deviations. This study aims to characterize the effectiveness of ASR in removing artifacts and how k affects its performance.

4.3.3 Evaluating performance of ASR using independent component analysis

This study applies ICA to the ASR-cleaned data and utilizes the automatic IC classifier, ICLabel (see Chapter 3), to evaluate the effectiveness of ASR in both removing artifact signals and preserving brain activity.

Independent component analysis (ICA)

ICA has been widely used for separating stereotyped brain processes and various types of artifacts such as muscle, eye-blink, and lateral eye-movement activities [Jung et al., 2000a]. ICA assumes EEG data, x, can be modeled as a linear mixture A of statistically independent sources, s, and learns an unmixing matrix, W, such that the independent components (IC) recover the original sources, y.

$$x = As$$
$$y = Wx \approx s$$

In this study, we employ extended Infomax ICA [Lee et al., 1999], which is available in the `runica` function in EEGLAB, an open source MATLAB toolbox.

Changes in spatial distribution and temporal activities of independent components

With the ICA decompositions of ASR-cleaned data, we can quantitatively assess the extent to which ASR affects the activities of the brain and artifactual ICs in two ways. First, we compute the component-wise correlation coefficients of the best-matched ICs across ICA decompositions of EEG data with and without ASR cleaning, that is, rearranging the order of ICs to maximize $\sum_i \mathrm{Corr}((A^{(k)})_i, (A^{(*)})_i)$ where A refers to the linear mixing matrix of ICA, k refers to ASR's cutoff parameter, $*$ refers to no ASR cleaning, and i is the column index. The matching process was performed using the Hungarian method [Kuhn, 1955] in the `matcorr` function in EEGLAB. This enables assessment of the stability of ICs across different ASR thresholds (k) by examining whether ICs disappear, change, or remain the same. Second, we apply the spatial filter $W^{(*)} = (A^{(*)})^{+}$, obtained from the ICA decomposition of raw data, to ASR-cleaned data $X^{(k)}$. Then we calculate the IC activities,

$$Y^{(k)} = W^{(*)} X^{(k)} \tag{4.2}$$

and compare the mean power reduction for the IC activities.

$$\text{Power reduction} = \text{Mean}(\text{Var}(Y^{(k)})) - \text{Mean}(\text{Var}(Y^{(*)})) \tag{4.3}$$

This reveals the effectiveness of ASR at reducing the activities of artifactual ICs and preserving those of brain-related ICs.

IC classification

To summarize the ICA results across subjects, we classify the ICs from each decomposition using an automated IC classifier. We utilize the `iclabel` function from the ICLabel EEGLAB extension to classify ICs into seven classes: “Brain”, “Eye”, “Muscle”, “Heart”, “Line Noise”, “Channel Noise”, and a class for ICs which do not fit into the first six classes: “Other”.

Changes in dipole fitting result of ICs

The quality of an ICA decomposition can be measured by the number of dipolar ICs, whose spatial distribution over the scalp can be modeled by a current dipole in the brain, as suggested by Delorme et al. [2012]. Since large-amplitude artifacts usually disrupt ICA decompositions, we expect that ICA will find more dipolar ICs if the artifacts are removed from data. In this study, we employ the `dipfit` function in EEGLAB and consider ICs with residual variance, the mismatch between IC's spatial distribution over the scalp and the projection of fitting dipole, lower than 5% to be "dipolar" sources [Henderson et al., 1975, Delorme et al., 2012].

4.4 Results

4.4.1 Data modification and variance reduction through ASR cleaning

Figure 4.2 shows the percentage of data points modified by ASR (i.e., rejecting at least one component) and the average variance reduction of the data before and after ASR cleaning using different cutoff parameters k. The average portion of the reference data selected by ASR across 20 EEG recordings is 43.9% with standard deviation 14.6%.

In Figure 4.2A, when the cutoff parameter $k = 100$, less than 3% of data were modified while still reducing variance by more than 20%. When k is between five and seven as previously suggested in Mullen et al. [2015], ASR modified nearly 80% of data and reduced 80% of signal variance.

Figure 4.2B shows that the percentage of data modified and variance reduced started to increase when $k \leq 30$. When k is between five and seven, ASR modified 50% of reference data and reduced the signal variance by 30%. One thing to note is that, by visual inspection, there are still some eye and muscle activities in the reference data and ASR starts to reduce variance when the threshold falls bellow $k = 1000$ in Figure 4.2B.

4.4.2 Stability of ICs across choices of the ASR parameter

We examined the stability of IC by calculating component-wise correlation coefficients across ICA decompositions of the EEG data with and without ASR cleaning.

Through visual inspection of the results shown in Figure 4.3, we found that those ICs which were preserved by ASR with $k = 1$ (shown in the green box in Figure 4.3B) were likely to

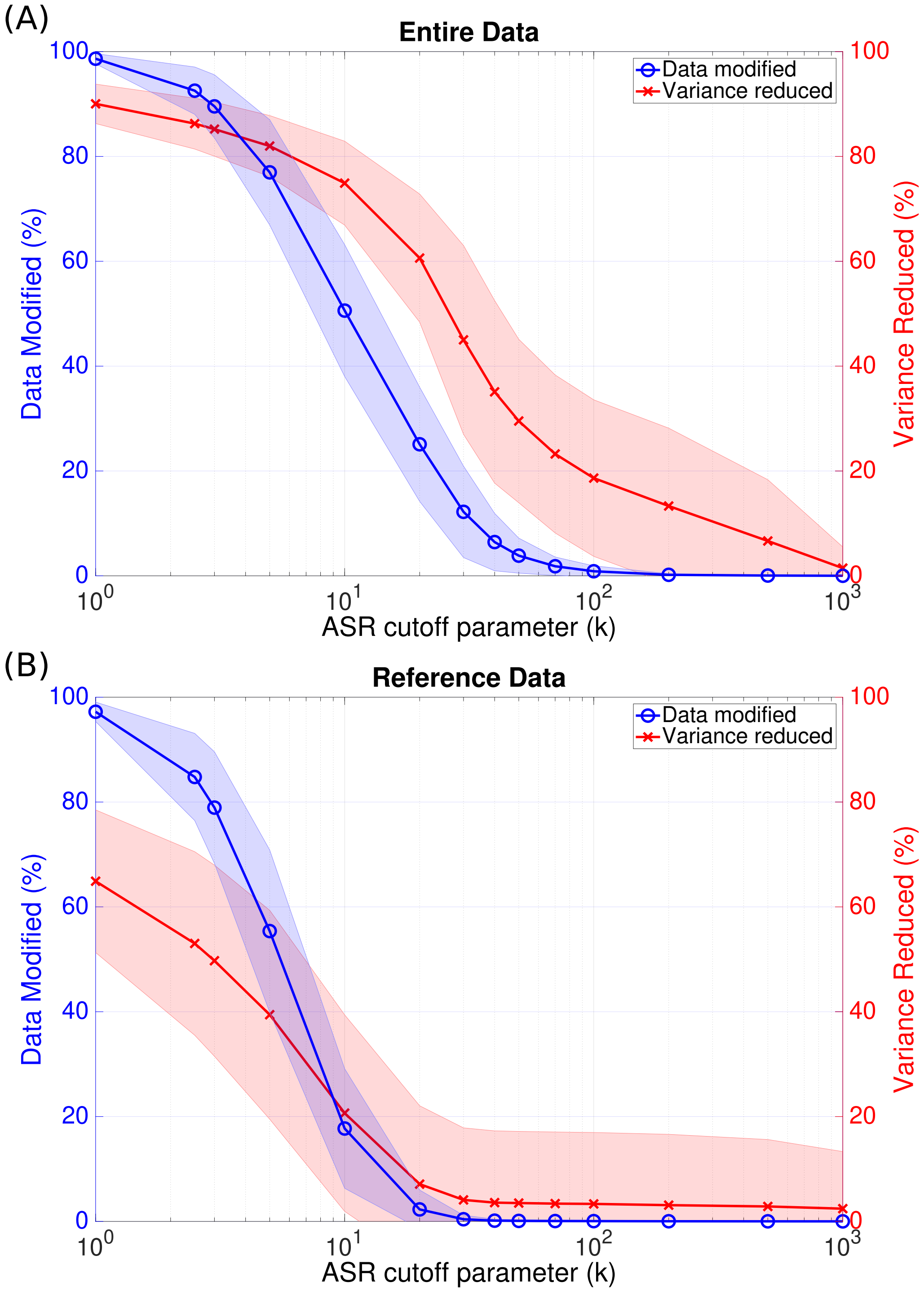


Figure 4.2: The percentage of data modified (blue) and variance reduced (red) by ASR with different cutoff parameters with respect to the same data without ASR cleaning. The shaded area shows one standard deviation across 20 EEG recordings. Figure 4.2A shows the result on entire data and Figure 4.2B only shows the result on the reference data, which ASR used to determine the value of thresholds.

be associated with brain activities (IC2, IC4 and IC5). These ICs were visually characterized by spatially homogeneous scalp maps [Delorme et al., 2012]. Interestingly, the ICs accounting for eye-blink (IC7) and eye-movement (IC3) activities were also consistently present when different values of *k* were used. On the contrary, those ICs which disappeared when the value of *k* was smaller than 70 were likely to account for artifacts due to single-channel noise (IC25 and IC29) or localized muscle activities (IC27 and IC28), visually characterized by scalp maps with sparse and localized activity.

To quantify the above results across subjects, Figure 4.4A depicts the percentage of preserved ICs at each ASR threshold from all 20 EEG recordings, categorized into five groups using classifications from the ICLabel classifier (see Section 4.3.3) and IC dipolarity (see Section 4.3.3). Figure 4.4A shows that, when $k = 5$, ASR altered 50% of ICs. However, the ratio of Dipolar Brain sources in preserved ICs increased, compared to the ratio in ICs without ASR cleaning, from 20% to 30%.

Figure 4.4B shows the percentage of preserved ICs within each group at each ASR threshold. When $k = 100$, almost 20% of ICs in the Eye and Muscle classes were removed. When $k \geq 20$, 90% of Dipolar Brain ICs were preserved while less than 70% of ICs in the other four classes were preserved. When using *k* between five and seven, less than 60% of Eye ICs and 50% of Muscle and Other ICs remained, but also removed 15%–25% Dipolar brain ICs.

4.4.3 Source power reduction by ASR cleaning

To further quantify how different types of signals were removed by ASR, we calculated the ICA decomposition of EEG data without ASR cleaning and applied the learned spatial filters (i.e., the IC scalp maps) to the same data after ASR-cleaning data, and computed their source activities retained after ASR cleaning with different cutoff parameters.

Although eye-blink-related ICs (IC7) and eye-movement-related ICs (IC3) were still present as reported in Section 4.4.2, Figure 4.5A shows that their power were reduced to the

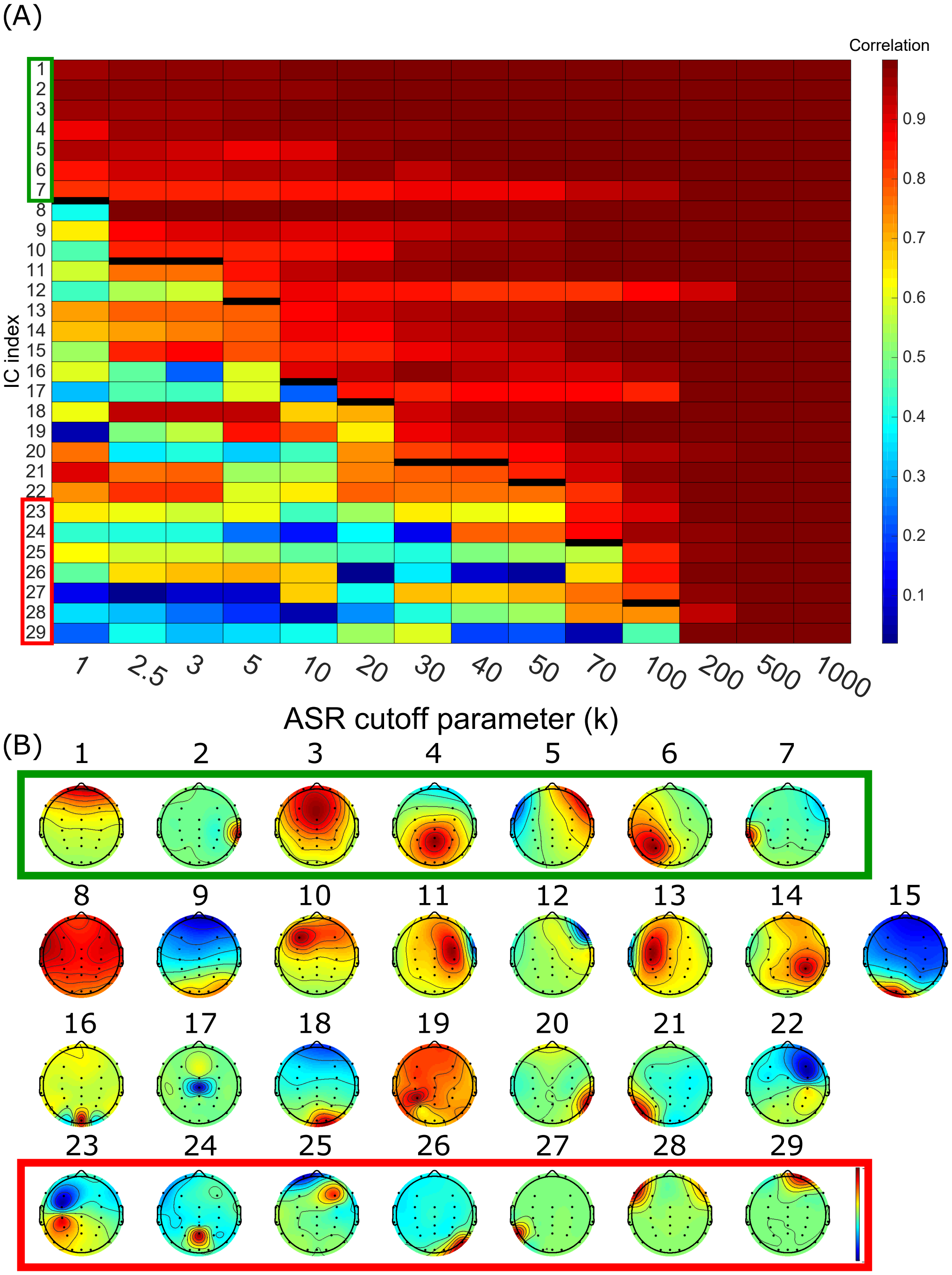


Figure 4.3: (A) Component-wise correlation coefficients of the best-matched ICs between ICA decomposition of the EEG data with ASR cleaning across different ASR cutoff parameters and without ASR cleaning. The IC index is sorted by whether ICs' correlation coefficient is higher than 0.8, indicating by the black lines. (B) The scalp maps, i.e., the spatial distribution of each source activities over the scalp channels, of the ICA decomposition without ASR cleaning (template ICs). The green box indicates preserved ICs after ASR cleaning while the red box indicates ICs which disappeared when $k \leq 50$.

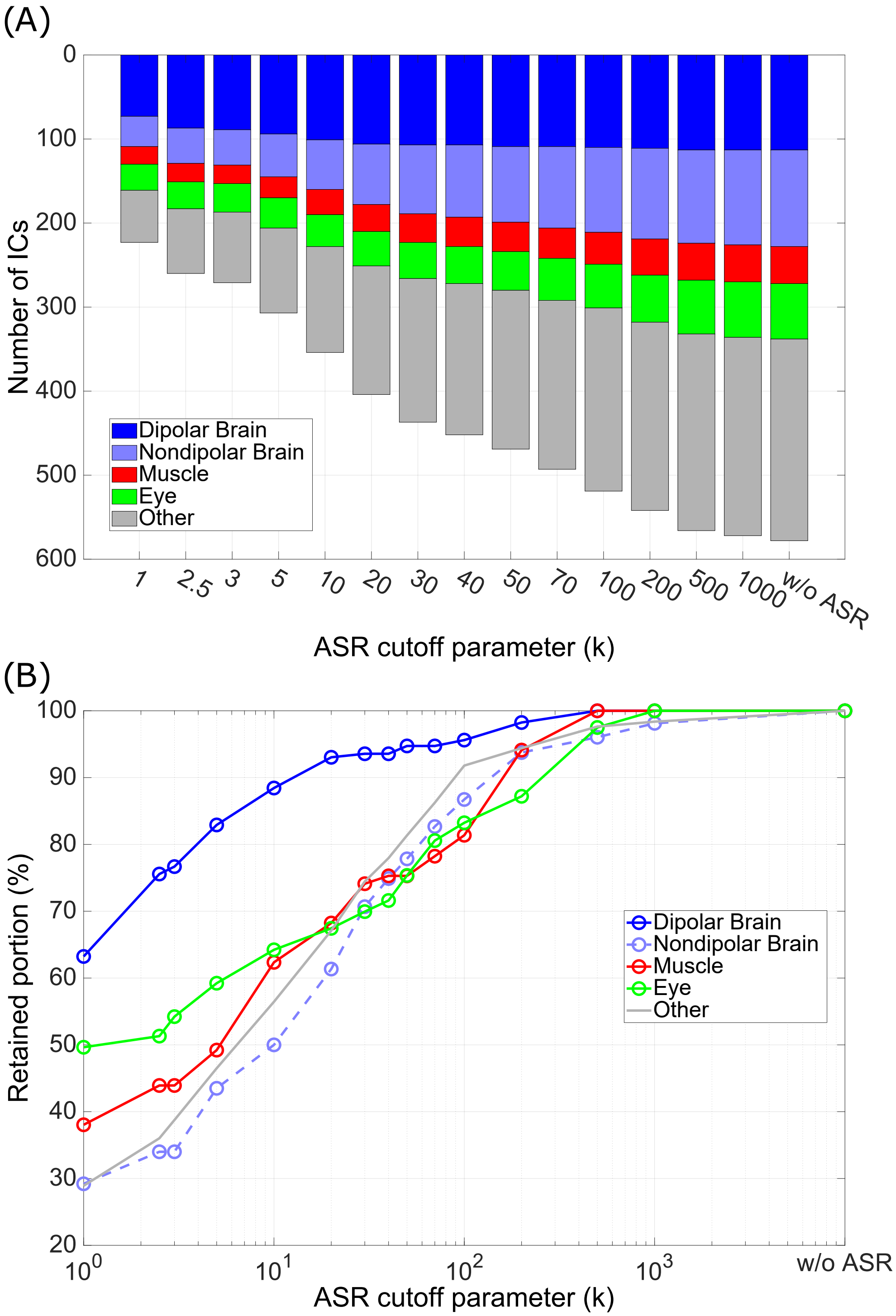


Figure 4.4: (A) Number of preserved ICs which have correlation coefficient higher than 0.8. The classification result of “Dipolar Brain”, “Nondipolar Brain”, “Muscle”, and “Eye” are shown in blue, light blue, red, and green respectively. ICs outside these four classes are labeled as “Other” in gray. ICs with residual variance $< 5\%$ were labeled as dipolar. (B) The percentage of preserved ICs within classes.

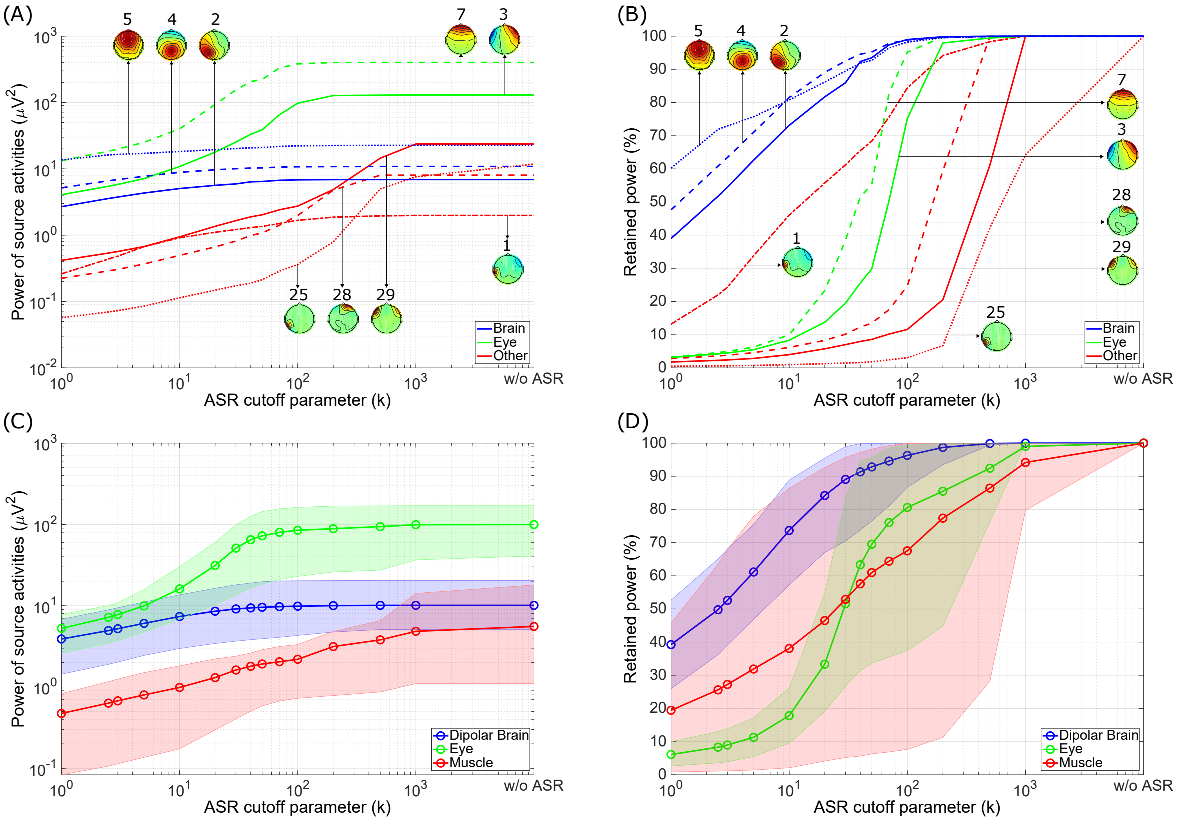


Figure 4.5: (A) The power of source activities of selected ICs in Figure 4.3 when different ASR cutoff parameters were applied. These ICs were classified as “Brain” (blue), “Eye” (green), and “Other” artifact (red) by visual inspection. The scalp maps and indexes of those ICs are also shown. (B) The percentage of retained source activity power of the same selected ICs from (A). (C) The average power of source activities of ICs from all subjects. The ICs were classified by the ICLabel classifier and the result Brain class (blue), Eye class (green), and Muscle class (red). The shaded areas represent 10% through 90% quantiles. (D) The percentage of retained power of source activities of the same classified ICs from (C).

same level as those of Brain sources when k is between five and seven. Moreover, Figure 4.5B shows that, when k is between five and seven, eye-related and likely-artifact ICs only retained 5% of their power after ASR cleaning. On the other hand, 70% and 90% of the power of the brain-related ICs (IC2, IC4, IC5) were retained when $k = 5$–7 and 30, respectively. Even though IC1 was preserved when $k = 1$ in Figure 4.3, ASR removed 65% and 40% of IC1’s power with $k = 5$ and $k = 30$, respectively.

This single-subject results was also seen across subjects. Figure 4.5C plots the source power of each of three classes ("Brain", "Eye", and "Muscle") averaged over all ICs in the same class across all 20 EEG recordings. To prevent the shaded area from exceeding the range of percentages (0 through 100), the shaded area shows 10% through 90% quantiles instead of standard deviation. The source power of ICs in the Eye class were ten times larger than those in the Dipolar Brain class in the data without ASR cleaning. However, when $k \leq 10$, the source power of Eye class and Dipolar Brain class were comparable. On the other hand, the source power of ICs in the Muscle class was comparable to the source power of ICs in the Dipolar Brain class in data without ASR cleaning when k was large, but became nine times smaller when $k \leq 100$.

Figure 4.5D shows that, when $k = 100$, ASR removed on average 30% of the source power of ICs in the Muscle class, and 10%-90% quantiles show that ASR's effectiveness varied drastically across Muscle ICs. When $k \geq 30$, ASR retained 90% of the power of Dipolar Brain ICs, while only retained 50% of the power of Eye and Muscle ICs. When using ASR with k between five and seven, the retained power in Eye and Muscle ICs were 10% and 30% respectively. However, 40% of the source power of Dipolar Brain ICs were removed as well.

4.4.4 Improvement of ICA decomposition

The Figure 4.3 and 4.4 provide qualitative and quantitative results of which ICs survive ASR cleaning. To further assess ASR's effect, Figure 4.6A reports the total number of dipolar ICs present after different levels of ASR cleaning. When $k \leq 50$, the ICA decomposition of the ASR-cleaned data found significantly more dipolar sources, which indicates a better decomposition according to [Delorme et al., 2012]. Furthermore, Figure 4.6B shows that the number of Dipolar Brain sources increased, on average, by 10% when $k = 5$ and 5% when $k = 20$.

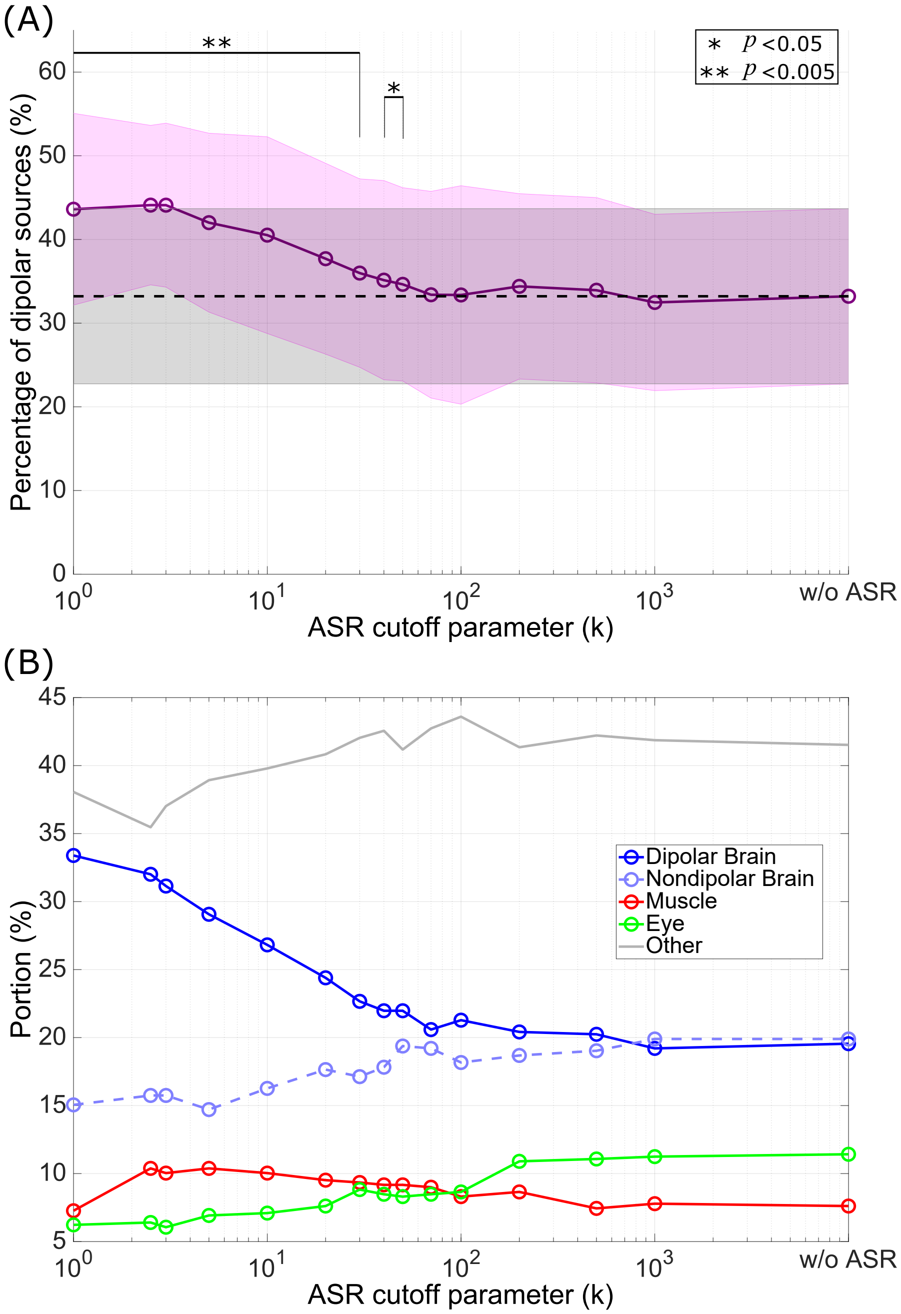


Figure 4.6: (A) The average percentage of all dipolar sources in ICA decomposition of EEG data with (solid line) and without (dashed line) ASR cleaning. The shaded area represents one standard deviation across subjects. The statistical significance between number of dipolar sources with and without ASR is calculated by bootstrap. (B) The percentage of ICs in each IC class after ASR was applied with different cutoff parameters. The classification results into the categories of "Dipolar Brain", "Nondipolar Brain", "Muscle", and "Eye" are shown in blue, light blue, red, and green respectively. ICs outside these four classes are labeled as "Other" in gray. ICs with residual variance $< 5\%$ were considered dipolar.

4.5 Discussion

Artifact Subspace Reconstruction is an automatic, online-capable artifact removal method which has been increasingly used in EEG pre-processing. However, ASR has not been properly validated and the optimal user-defined cutoff parameter is unknown. This study aims to systematically evaluated and quantitatively assessed the effectiveness of ASR on real EEG data using ICA decomposition with the following measures: (1) percentage of data modification versus variance reduction, (2) percentage of reference data that are affected, (3) how many artifact ICs remain and how much their powers are reduced, and (4) how many Brain ICs are preserved and their source activities are affected.

The empirical results show that the effectiveness of ASR heavily depends on the choice of its cutoff parameter k. As shown in Figure 4.2A, a mild threshold ($k = 100$) could remove sparse (1% of data) yet large-amplitude artifacts (20% of the variance). When k was 20, ASR started to affect the reference data, indicating that even the clean data ASR used to determine thresholds was modified. With the previously suggested values (k between five and seven) [Mullen et al., 2015], ASR modified 70% of data and removed up to 80% of the variance, which may affect brain signals and distort experiment results.

To assess the types of signals removed by ASR, we decomposed the ASR-cleaned EEG signals using ICA and classified the ICs as brain-, eye-, and muscle-related sources. We found that more Muscle, Eye, and Other ICs disappeared than the Dipolar brain ICs did after ASR cleaning (Figure 4.4B). When $k \leq 20$, more Dipolar brain ICs were affected and the ratio of removing artifact ICs versus Dipolar brain ICs deteriorated. Although some Muscle and Eye ICs were still present after ASR cleaning, their powers were strongly reduced (Figure 4.5). The retained power from the Eye ICs went from 80% ($k = 100$) to below 20% ($k = 10$); the retained power from the Muscle ICs went from 93% ($k = 1000$) to 40% ($k = 10$). When $k \geq 30$, 90% of power from the Brain ICs were still preserved, but the retained power decreased to 50%–60%

with k between five and seven.

Given the above observations, the recommended ASR cutoff parameter k is between 20 and 30. ASR with a conservative threshold $k = 30$ removed 25% of the Eye and Muscle ICs and reduced almost 50% of the power of the Eye and Muscle activities while only affecting less than 10% of the Dipolar Brain ICs and removing only 10% of their power. ASR with a lower threshold of $k = 10$ further removed 15%–30% of Eye and Muscle activities, but at the cost of reducing 15% more Brain signal power. The previously suggested value of k between five and seven is too aggressive in removing both artifact and brain signals and is not recommended.

Interestingly, ICA decompositions of the ASR-cleaned data found more Dipolar Brain sources when a smaller cutoff parameter was applied. Because ICA is sensitive to large-amplitude artifacts, applying ASR before ICA can increase the quality of an ICA decomposition, as shown by the increase in the number of dipolar EEG sources found [Delorme et al., 2012] (Figure 4.6).

Compared to other existing artifact removal methods, the benefit of ASR is that it can automatically adapt its thresholds based on the statistics of the EEG data. Moreover, ASR can remove transient, large-amplitude artifacts which ICA-based methods are usually incapable of dealing with. In fact, a combined use of ASR and ICA might be even more effective in removing different types of EEG artifacts and is discussed further in Chapter 6.

A recent paper [Gabard-Durnam et al., 2018] has also compared their automatic artifact removal method with ASR. Similar to our evaluation method, they evaluated ASR by comparing the variance reduction and the probabilities that the ICs surviving MARA's rejection are artifact-contaminated. In their paper, they found that ASR removed more variance from EEG data but retained higher artifact-contaminated probabilities after cleaning than theirs. However, they chose the cutoff parameter $k = 5$, which is too aggressive. Moreover, the accuracy and descriptiveness of MARA are not as good as ICLabel (see 3.1), which might be a concern when taking the probabilities calculated from MARA as an evaluation factor. In addition to the classification result, the current study investigates IC activities and IC dipolarity, which explains each IC's contribution to variance reduction and ICA decompositions respectively.

Even though ASR shows great effectiveness in removing large-amplitude artifacts, some points should be carefully considered when using ASR. As a variance-based artifact removal method, ASR might be limited in removing artifact such as eye activities up to the point where the power of the artifact is comparable to that of the brain signals. Also, if the artifacts are consistently present and therefore unavoidably included in the reference data, ASR will not be able to remove them. As a remedy, especially for eye-related artifacts, an ICA-based artifact removal method can be utilized after ASR cleaning, as is proposed in Chapter 6. Likewise, if brain signals are not present in the reference data, ASR could remove those brain signals as well. One potential solution is updating ASR thresholds by incorporating incoming clean data into the reference data.

4.6 Conclusion

This study demonstrates that Artifact Subspace Reconstruction is an effective automatic artifact removal approach, quantifies ASR's effectiveness in removing different types of signals as shown using Independent Component Analysis, and provides insights into the optimal choice of ASR's cutoff parameter. Our empirical results suggest using a cutoff parameter between 20 and 30 rather than the previously suggested and default values between five and seven [Mullen et al., 2015] where brain activities were excessively removed. This study also found that ASR improves the quality of ICA decomposition as evidenced by an increased number of dipolar independent components.

ASR has been implemented and disseminated in the Real-time EEG Source-mapping Toolbox (REST), described further in Chapters 5 and 6. With an appropriate choice of the cutoff parameter, ASR can be a powerful artifact removal approach for subsequent data analysis such as ICA and its online capability enables real-time artifact rejection for brain-computer interfaces and clinical applications.

4.7 Acknowledgements

Chapter 4, in part, has been submitted for publication of the material as it may appear in Chi-Yuan Chang, Sheng-Hsiou Hsu, Luca Pion-Tonachini, and Tzyy-Ping Jung. Evaluation of artifact subspace reconstruction for automatic EEG artifact component removal. *Transactions on Biomedical Engineering (under review)*, 2019. The dissertation author was the tertiary investigator of this paper. The authors gratefully acknowledge the valuable suggestions and comments from Makoto Miyakoshi and the author of artifact subspace reconstruction, Christian Kothe.

Chapter 5

Real-time Source Separation

5.1 Introduction

Electroencephalogram (EEG) source analysis combining independent component analysis (ICA) and source localization has generally been solved offline because of its computational cost. With faster processors and algorithmic advances, near real-time online applications are becoming even more viable. Bringing these analysis methods to the domain of real-time processing would allow for the use of more specific neurophysiological information in closed-loop brain-computer interfaces (BCI) and neurofeedback paradigms, and could also provide experimenters online feedback useful for data quality control. Streaming EEG data require a computational pipeline that is light enough to keep with the rate of data collection and that performs computations on data segments either recursively or independently. In this chapter we developed a toolbox for real-time EEG source analysis which automatically imports, processes, decomposes, and localizes streaming EEG data. The toolbox was developed with the twofold purpose of (1) demonstrating the utility and capabilities of these automatic, online-capable tools and (2) rendering their use easy enough to encourage adoption among EEG researchers. This toolbox is further extended in Chapter 6 with the methods covered in Chapters 3 and 4.

5.2 Background

A source-resolved imaging approach models the collected EEG as the sum of electric fields produced by many small patches of cortex whose local field activities are fully or partially synchronous, each such patch thus functioning as an effective EEG source with a scalp projection identical to that of a single equivalent current dipole (ECD). Source localization requires solutions to both the forward and inverse imaging problems: the forward problem (FP) determining the scalp projection patterns of the possible brain sources based on accurate modeling of head tissue geometries and conductivities, and the inverse problem (IP) estimating the locations and orientations or cortical surface distributions of one or more source projection patterns.

Many existing EEG processing toolboxes attempt to solve these problems, including core EEGLAB [Delorme and Makeig, 2004], BCILAB [Kothe and Makeig, 2013], LORETA-KEY [Pascual-Marqui et al., 1999], and Fieldtrip [Oostenveld et al., 2011]. They all operate offline or attempt to solve the IP by directly operating on, e.g. response-averaged EEG channel data. Approaching the IP directly from the EEG channel data complicates the problem by requiring determination of the number of sources to localize [Oostenveld et al., 2011], a problem whose computational cost and number of false local minima increase dramatically with the number of sources being estimated. Other approaches simply attempt a low-resolution joint spatial estimate of all the active sources [Pascual-Marqui et al., 1999]. Blind source separation (BSS) can be used as an initial 'unmixing' step to simplify an inverse problem by separating it into much simpler problems of finding the locations of the individual effective sources [Makeig et al., 1996, 2004, Marco-Pallares et al., 2005].

ICA has been shown to work exceedingly well when applied to EEG [Delorme et al., 2012] as EEG data and ICA share many important assumptions. ICA assumes that input data are the result of a linear mixing of spatially stationary independent time series or independent components (ICs). Here, we present the Real-time EEG Source-mapping Toolbox (REST), a collection of automated EEG analysis methods accessible through a graphic user interface (GUI). By applying Online Recursive ICA (ORICA) [Akhtar et al., 2012], we can estimate a solution to the source separation problem in near real-time, allowing low-latency access to source information, making possible innovations in experimental designs including a wide variety of clinical and non-clinical BCI paradigms. REST also allows the user to estimate the brain locations of the estimated sources using either LORETA [Pascual-Marqui et al., 1999, Ojeda et al., 2014] or minimum-variance ECD fitting [Oostenveld et al., 2011].

REST provides estimates of source activations and their current power spectra, plus source scalp maps (source scalp projection patterns) and cortical source locations. Below, we show the layout of the REST GUI and detail the measures used in its analysis pipeline. We then test its accuracy and efficacy by applying it to simulated EEG data with known source locations and activations. Finally, we demonstrate the real-world utility of REST and its ease of use by applying it in a common BCI paradigm recording session.

5.3 Methods

REST is coded in MATLAB using the EEGLAB environment. It uses a processing pipeline, shown in Fig. 5.1A, designed to run from beginning to end with minimal user input. Preprocessing and source separation are implemented as a BCILAB pipeline followed by source localization implemented in part using routines in MoBILAB [Ojeda et al., 2014].

5.3.1 Preprocessing

The toolbox pulls EEG data from a data stream received through the Lab Streaming Layer framework [Kothe, 2014]. The data are first preprocessed by IIR high-pass filtering. Artifact Subspace Reconstruction (ASR) [Kothe and Jung, 2014, Mullen et al., 2013], analyzed in Chapter 4, may be introduced as an additional preprocessing step to remove large movement-based

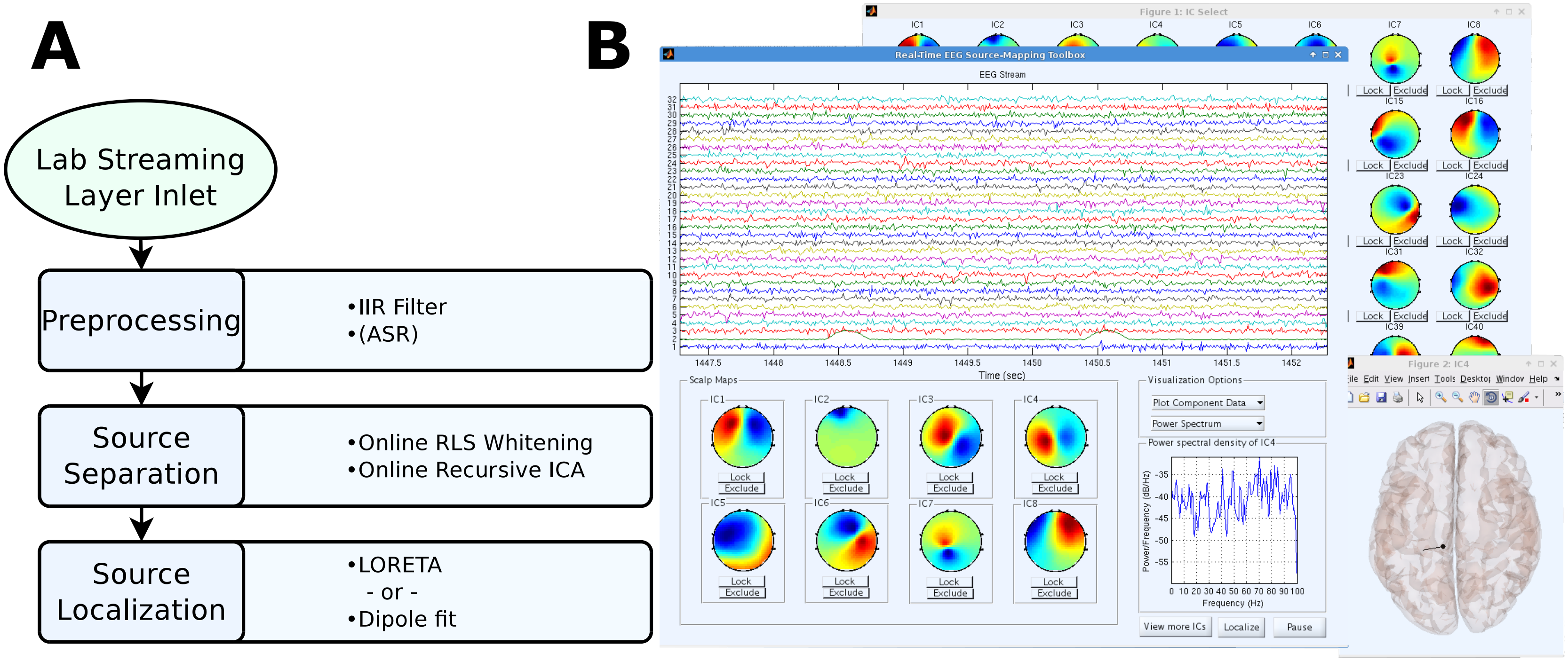


Figure 5.1: (A) The pipeline used in the Real-time EEG Source-mapping Toolbox (REST). (B) The toolbox GUI. The main window (left) shows the scrolling EEG channel or independent component (IC) activation data plus eight (constantly updated) IC scalp maps. A source location estimate for IC4 is shown (lower right). Behind this (upper right), another REST window shows all the estimated IC scalp maps.

artifacts as shown in Chapter 6.

5.3.2 Source separation

Next, the EEG data are whitened using an online RLS whitening algorithm to improve convergence and then linearly unmixed using ORICA. ORICA is, so far as we know, the only ICA implementation that is real-time capable with acceptable convergence rates for relatively large numbers of channels [Hsu et al., 2014]. The output of ORICA is a set of linear IC filters that are used to separate the IC activation time courses and scalp maps from the EEG channel data. When the data sources are spatially and statistically stationary, the ICs that ORICA provide asymptotically approach those that (offline) Infomax ICA [Lee et al., 1999] returns. Unlike Infomax ICA, ORICA can also adapt to source nonstationarities (more on that in Chapter 7).

5.3.3 Source localization

Estimated IC source locations are calculated using one of two cortically-constrained source models (either distributed or ECD). Distributed source location model estimates are calculated using cortically-constrained LORETA with Bayesian hyper-parameter estimation [Trujillo-Barreto et al., 2004] from MoBILAB, while the ECD model estimates are computed using minimum residual variance fitting. Both the ECD and distributed source methods require a MoBILAB head model object to be created in advance, which can be computed easily using the included helper function. The head model uses spatial meshes representing the geometry of the cortex, scalp and one or more intervening head tissue types (e.g. skull, CSF, white matter). A lead field matrix (LFM) is calculated (automatically) using OpenMEEG [Gramfort et al., 2010], as well as a surface Laplacian operator for the cortical mesh. By default, the included helper function creates a 3-layer (scalp, skull, cortex) boundary element method (BEM) head model based on the MNI Colin 27 brain. The primary input to the source localization methods is an estimated ICA scalp map for the source being localized.

5.4 Materials

5.4.1 Toolbox

The REST main window, on the left of Fig. 5.1B, displays either raw EEG or estimated IC activations. It also shows scalp maps and power spectra for the estimated ICs, as well as convergence statistics. All the visualized information updates in near real-time. The (partially occluded) window in the top rights of Fig. 5.1B provides an easy way to select which ICs are displayed on the main window. On the bottom right of Fig. 5.1B is the source localization window which shows the current estimated source location for an IC as either an ECD or a distributed source.

5.4.2 Experiments

To show the utility of REST, we designed two experiments. One, using simulated source-resolved EEG data, for which we know the ground truth, tested the integrity of the REST pipeline. The other used actual EEG collected during a steady-state visually evoked potential (SSVEP) BCI paradigm to test the utility of the toolbox in interactive paradigms.

Simulated EEG data

For the simulation, we used the default head model with ECD sources constrained to be normal to the cortical surface. We simulated 10 minutes of 64 channel EEG using SIFT [Delorme et al., 2011] by placing ECDs at various vertices of the cortical mesh and generated source activation time series for each. Two sources were handcrafted to imitate eye-blinks and occipital alpha activities while the rest were vector autoregressive processes driven by super-Gaussian noise. These are then mixed together using the LFM associated with the head model. As this was a test for accuracy rather than speed of convergence, we evaluated the accuracy of the ORICA decompositions and resulting source location estimates at the end of the simulated data collection. For information on the convergence properties of ORICA, see [Hsu et al., 2014].

Actual EEG data

To collect the actual human EEG we used a low-cost, 14-channel Emotiv headset. This setup wirelessly streams data to a computer via Bluetooth. The streaming data were transferred to an LSL stream for REST. During the experiment, 2 minutes of eyes-closed resting allowed ORICA to identify relevant ICs. This was followed by 2 trials in which the subject looked at flashing phone-pad style digits on a tablet. The subject first focused on the symbol “1” and then afterwards at the symbol “#” which were flashing at 9 Hz and 11.75 Hz respectively. This tested the adaptivity of the pipeline, as going from eyes-closed rest to viewing flashing stimuli could be expected to produce a noticeable change in brain sources and source activities.

5.5 Results

5.5.1 Simulated 64-channel stationary EEG

As shown in Fig. 5.2 and 5.3, ORICA and both source localization techniques perform as intended. Fig. 5.2 visualizes the full REST pipeline applied to three of the 64 simulated sources. In the first estimation step, ORICA successfully decomposes the sources, providing accurate scalp map estimates and source activations. In the second estimation step, the ECD estimates were very close to the ground truth (shown in the green simulation box) in both location and orientation. The distributed source estimates, despite not theoretically matching the model used during simulation, provided patches of active cortex that were well situated about the simulated dipole location.

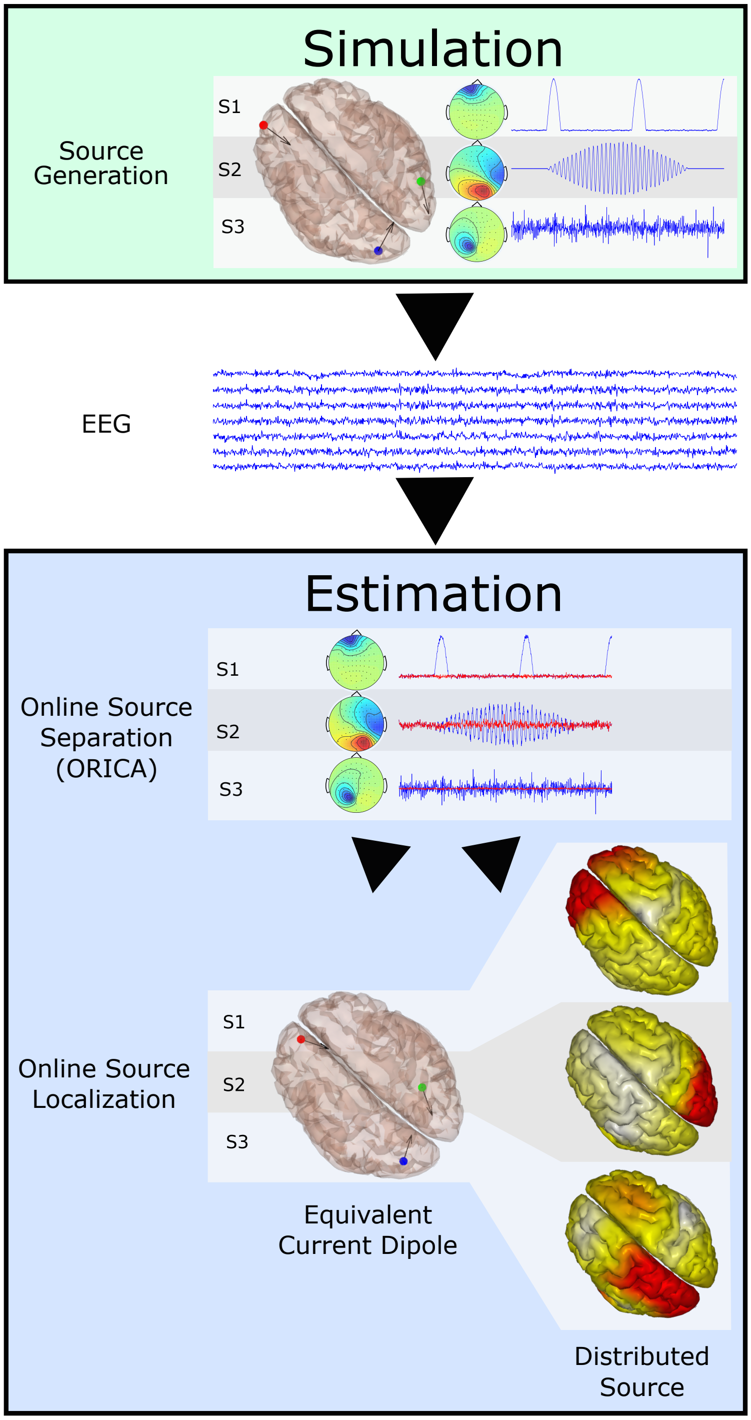

Simulation
Source Generation
S1
S2
S3
EEG
Estimation
Online Source Separation (ORICA)
S1
S2
S3
Online Source Localization
S1
S2
S3
Equivalent Current Dipole
Distributed Source

Figure 5.2: Visualizaton of the simulated data experiment: data simulation to source estimation. (Green box) The simulated data source activations are mixed. (Blue box) The simulated EEG data are first decomposed within REST into estimated independent components (ICs) using ORICA. Then the source location of each IC is estimated as either an equivalent current dipole (ECD, left) or as a low-resolution cortical distribution (right).

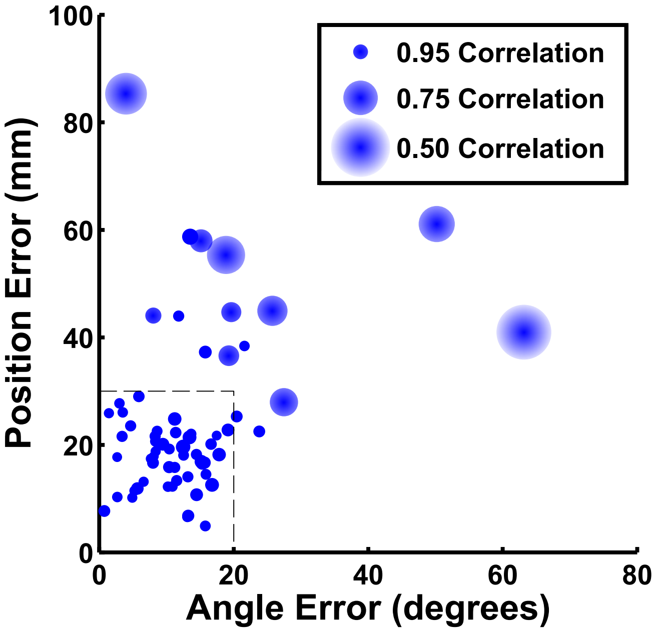


Figure 5.3: Source localization accuracy in the simulated data experiment using an equivalent current dipole (ECD) model for each estimated independent component (IC). Each disk represents an IC. Disk size shows how well the recovered IC scalp map correlated with the simulated source scalp map. For 48 of the 64 recovered sources, map correlations were above 0.95 with ECD model errors less than 3 cm and 20 degrees (lower left).

Fig. 5.3 illustrates the accuracy of all 64 estimated dipole positions and orientations, which were generally correct within 3 cm and 20 degrees respectively. The majority of the errors in dipole position were related to the depth of the dipole as the true source positions tended to be closer to the scalp than their estimates. Disk sizes in Fig. 5.3, which represent scalp map error, showed a clear correlation between localization error and scalp map error and provides a means of judging whether localization error is due to poor results from ORICA or error from the underdetermined nature of the IP. Here we used the same simulated FP head model to solve the IP, something not possible in actual use where the true FP head model can only be estimated. Nevertheless, these results indicate that REST can generate accurate source locations and activations provided a minimum level of data quantity and quality and sufficient head model accuracy.

5.5.2 Actual 14-channel EEG during SSVEP

The application of REST to data collected in an SSVEP paradigm showed that ORICA can converge to useful source solutions in real-life applications. Fig. 5.4 compares REST outputs during eyes-closed rest and attention to 9 Hz and 11.75 Hz flashing stimuli. Clearly, ORICA extracted an occipital IC, first during rest with a weak 10 Hz peak (top panel), and then during attention to 9-Hz (middle panel) and 11.75-Hz (lower panel) flashing stimuli. The ECD during the latter condition (not shown) changed in orientation as indicated by the change in its scalp map.

5.6 Conclusions

We have shown that REST can be accurate when applied to simulated data, and potentially usable in practice. There are many possible applications for real-time monitoring of sources of interest during an EEG experiment. The REST toolbox design allows possible extensions to implement near real-time computation, visualization, and application of other source-resolved EEG measures. REST could aid online data quality analysis, as when collecting EEG from particular sources if of specific importance. Additionally, the ORICA implementation in REST might be used to make a wide range of BCI paradigms more robust [Makeig et al., 2000]. We plan to add more flexible and detailed data preprocessing, since ICA can be highly influenced by large amplitude artifacts, and also automated IC classification. In theory, the ORICA decomposition and, with some modifications, the source localization methods in the REST pipeline should be as applicable to MEG as to EEG data. Finally, this work follows in spirit, and some details, previous work [Mullen et al., 2013] demonstrating a real-time application of the BCILAB [Kothe and Makeig, 2013] and SIFT [Delorme et al., 2011] toolboxes, into which the source identification and localization methods in REST might easily be introduced.

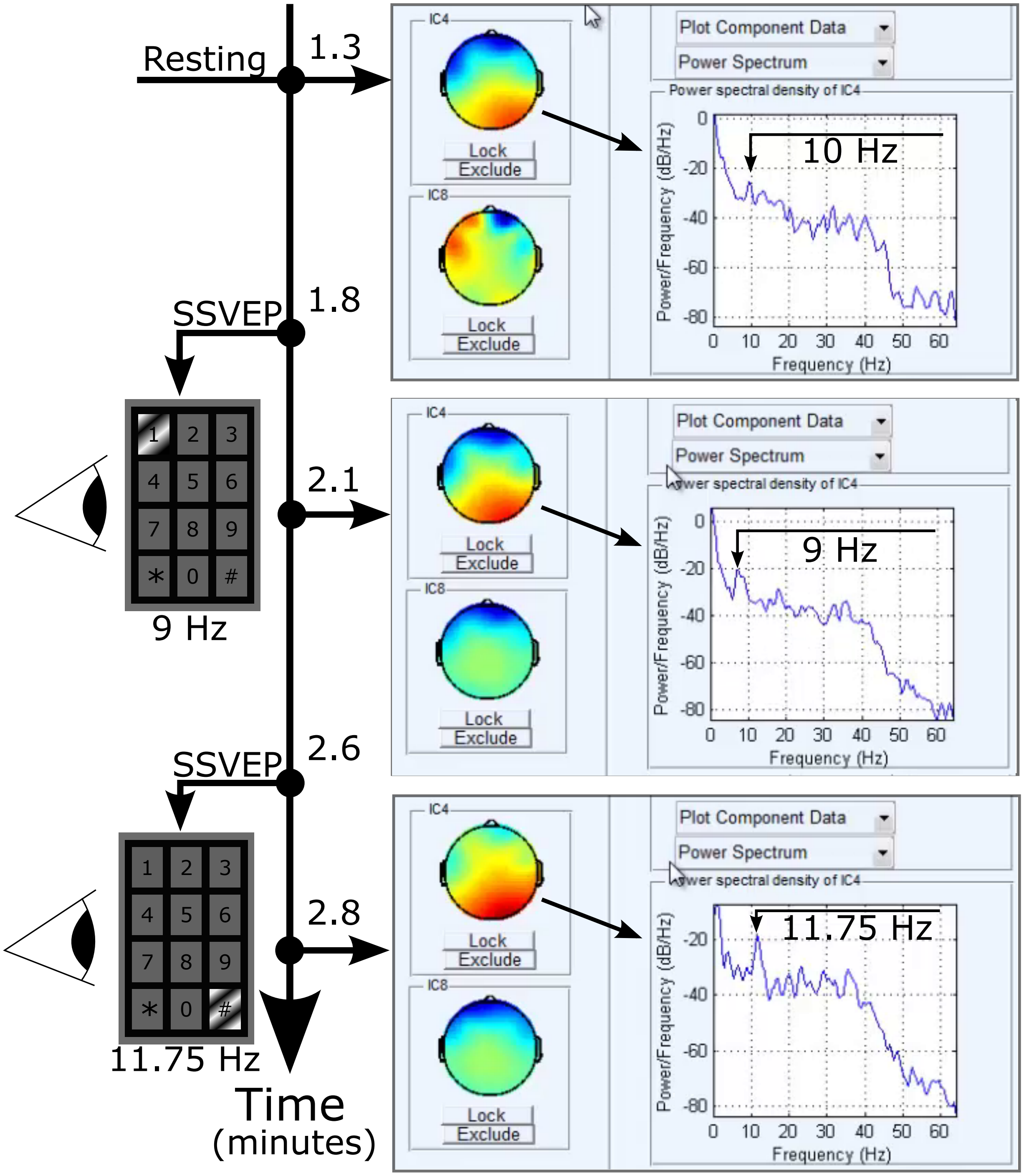


Figure 5.4: Screen captures of the REST GUI during an actual EEG experiment. At 1.3 min (top panel) during eyes-closed rest, the baseline PSD for IC had a peak at 10 Hz (alpha). At 2.1 min (middle panel), the subject attended the symbol “1” flashing at 9 Hz (note change in the IC4 spectrum). At 2.8 min (bottom panel), the subject attended the symbol “#” flashing at 11.75 Hz (note IC4 spectral shift and possible scalp map change).

5.7 Acknowledgements

Chapter 5, in part, is a reprint of the material as it appears in Luca Pion-Tonachini, Sheng-Hsiou Hsu, Scott Makeig, Tzyy-Ping Jung, and Gert Cauwenberghs. Real-time EEG source-mapping toolbox (REST): Online ICA and source localization. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 4114–4117. IEEE, 2015. The dissertation author was the primary investigator of this paper. Luca Pion-Tonachini and Sheng-Hsiou Hsu contributes equally to this work. This work was supported in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-10-2-0022 and by the National Science Foundation under grant numbers EFRI-M3C 1137279 and GRFP DGE-1144086.

Chapter 6

Real-time Artifact Rejection

6.1 Introduction

As described in Chapters 3 and 4, artifacts are a constant when working with electroencephalographic (EEG) data; especially when subjects must speak or move while recording EEG. For this reason, automation is a requirement for real-time applications. Humans cannot comfortably mark artifacts on a scrolling data plot, indicate which independent components (IC) are relevant to an analysis, and interpret the resultant signals all in real-time. The automation of EEG data preparation opens the possibility for applications in real-time, computational costs allowing.

In Chapter 5 we developed and the central pipeline which enable EEG source analysis in real-time. Here, we extend that pipeline to make it more robust by incorporating (1) channel-level artifact rejection as described in Chapter 4 and (2) automatic though IC selection and rejection using IC classifiers like the one developed in Chapter 3. We evaluate the effectiveness of these added methods through the difficult and relevant task of real-time artifact rejection. In this case, the EEG source pipeline summary provided in Section 1.2 is not quite applicable. Specifically, the steps following (5) change as noted below:

1. Recording and importing data

2. Preprocessing
3. Transient artifact removal
4. ICA decomposition
5. IC selection
6. Channel reconstruction
7. Channel analysis

Rather than performing source analysis, this pipeline evaluation instead uses EEG source measures as an additional method for segregating and removing non-brain-related signals from the recorded data. Similar pipelines using the same methods can also be created using the open-source Real-time EEG Source-mapping toolbox (REST) to better acquire and analyze EEG source data .

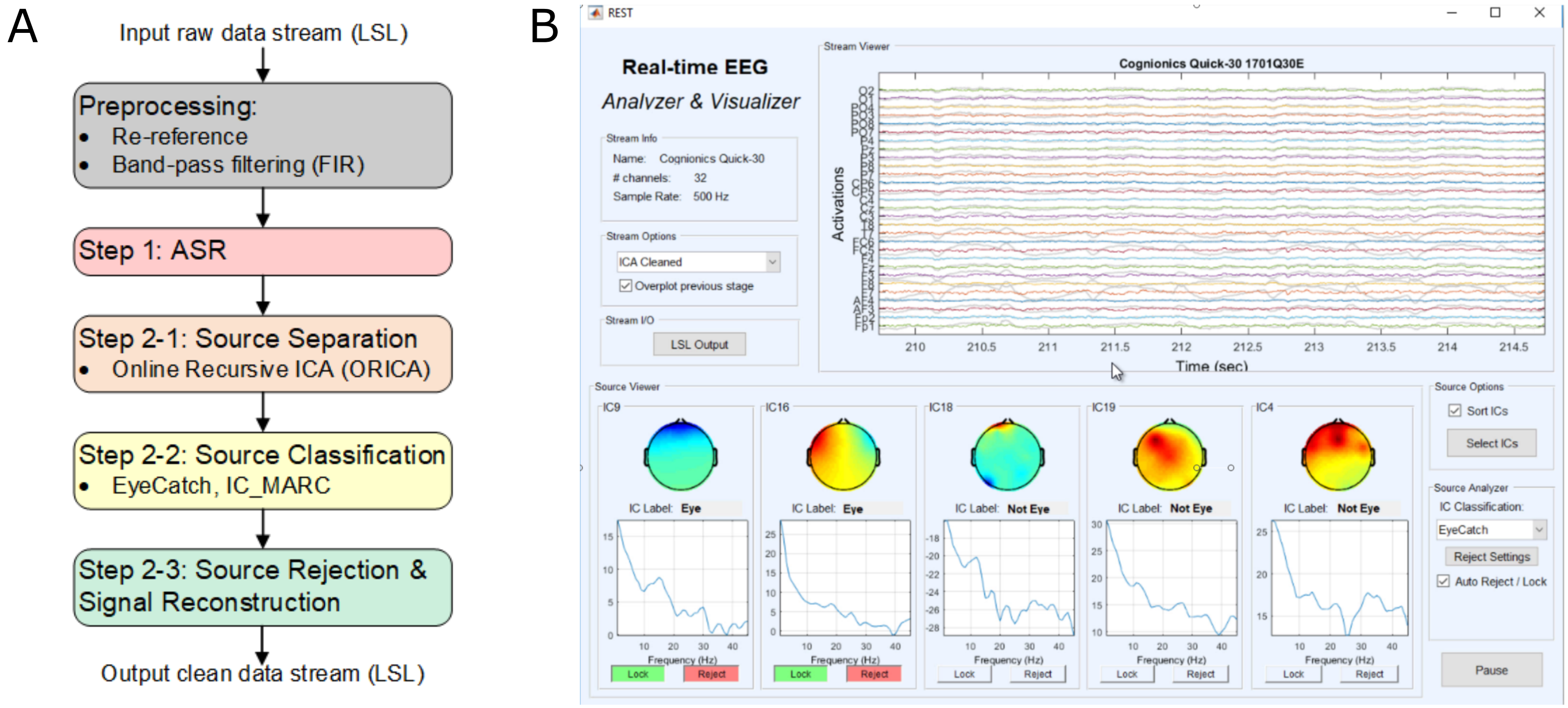
6.2 Background

Brain computer interface (BCI) and other real-time EEG applications often suffer when artifacts (unwanted signals included in a recording) are present [Minguillon et al., 2017, Urigüen and Garcia-Zapirain, 2015]. In EEG recordings, typical examples include perturbations induced by the retinal electrical dipoles during eye movements, high-frequency signals from scalp muscle activity, and large signal spikes and shifts from electrode impedance changes during subject movements (Figure 6.2). While the definition of “artifact" is largely context dependent, artifacts are typically detrimental to signal analyses as the amplitudes of these artifacts can easily eclipse the brain-generated EEG activity.

Previous methods for cleaning EEG data have taken several approaches: spectral-intensity thresholds [Gevins et al., 1977], filtering based on simultaneous electrooculographic (EOG) recordings [Joyce et al., 2004], complexly stacked wavelets, blind source separation (BSS), and wavelet-based classifiers [Mammone et al., 2012]. Each method comes with its own assumptions and may therefore fail in some situations. For example, none of the above-mentioned methods support real-time processing with the exception of EOG filtering – and that requires a dedicated EOG recording in addition to the EEG electrode montage. To the authors' knowledge, no existing EEG artifact cleaning method effectively cleans the data in near real time without a need for recording EOG or other artifact reference channels.

Independent component analysis (ICA) has been widely used for separating spatially stereotyped artifacts such as saccades and eye-blink activities from EEG data [Jung et al., 1998]. Online recursive ICA (ORICA) [Akhtar et al., 2012] has successfully converted the computationally-expensive ICA algorithm into incremental, recursive update rules that enable online, near real-time ICA decomposition [Hsu et al., 2016]. However, ICA decomposition is sensitive to unique, large-amplitude artifacts which can severely degrade the learned ICs and ICA-based methods have historically required visual inspection to manually identify the artifact-related ICs.

Here we apply artifact subspace reconstruction (ASR) [Kothe and Jung, 2014], an automated, near real-time-capable algorithm, prior to ORICA. We demonstrate that ASR can effectively remove transient, large-amplitude artifacts from EEG data and thus stabilize the ICA decomposition and improve artifact separation in real-time. Next, we apply a real-time capable IC classifier, here EyeCatch [Bigdely-Shamlo et al., 2013], to automate recognition and removal of artifact-related ICs. The full online, real-time, automatic artifact rejection (AR) pipeline, featuring ASR, ORICA, and EyeCatch, is available in REST, developed in Chapter 5 and illustrated in Figure 6.1B. REST can be downloaded from https://github.com/goodshawn12/REST.


A
Input raw data stream (LSL)
Preprocessing:
• Re-reference
• Band-pass filtering (FIR)
Step 1: ASR
Step 2-1: Source Separation
• Online Recursive ICA (ORICA)
Step 2-2: Source Classification
• EyeCatch, IC_MARC
Step 2-3: Source Rejection & Signal Reconstruction
Output clean data stream (LSL)
B
REST
Real-time EEG
Analyzer & Visualizer
Stream Viewer
Cognionics Quick-30 1701Q30E
Activations
Time (sec)
Source Viewer
IC9
IC16
IC18
IC19
IC4
Frequency (Hz)
Lock
Reject
Source Options
Select ICs
Source Analyzer
Reject Settings
Pause

Stream Info
Name: Cognionics Quick-30
# channels: 32
Sample Rate: 500 Hz

Stream Options
ICA Cleaned
☑ Overplot previous stage

Stream I/O
LSL Output

Source Options
☑ Sort ICs

Source Analyzer
IC Classification: EyeCatch
☑ Auto Reject / Lock

IC9 — IC Label: Eye
IC16 — IC Label: Eye
IC18 — IC Label: Not Eye
IC19 — IC Label: Not Eye
IC4 — IC Label: Not Eye

Figure 6.1: (A) An overview of the REST data-cleaning pipeline combining ASR, ORICA and an IC classifier. (B) The REST graphic user interface during a period containing repeated eye movements. In the lower left, two eye-related components have been found and removed by EyeCatch. In the upper right, the reconstructed channel signals are shown in color against uncleaned (gray) data traces in which the eye-movement artifacts are still visible.

6.3 Materials and Methods

6.3.1 Review of methods

ASR is an automated, variance-based EEG cleaning algorithm. It uses a short initial recording of artifact-free EEG data from which it learns a statistical model of the EEG data. Then for each incoming data window, ASR applies a principal component analysis (PCA) like linear transformation to the data using a transform matrix learned from the initial calibration data. If any principal component (PC) of the new data window is much larger than in the calibration data, that PC is removed from the window. An inverse transform then projects the data window back into the original channel coordinates.

ORICA consists of two stages. The first, online whitening, can be thought of as an online, recursive form of PCA. This is done to facilitate learning in the subsequent stage – online recursive ICA. This optimizes the same objective as offline Infomax ICA. Finally, the ICA solution from the second stage is projected to the nearest orthogonal matrix which further facilitates model convergence.

EyeCatch classifies ICs as either eye movement-related or not by first calculating the maximum correlation value of each IC scalp map to thousands of IC scalp maps in the method's library which account for eye-movement activities. An IC scalp map represents the relative contributions of a given IC to the scalp channels. Any IC for which the maximum correlation value, called the similarity score, is greater than a preset threshold is marked for removal. REST uses a modified version of EyeCatch that is computationally lighter and has a lower rejection threshold to support its use in a real-time setting.

6.3.2 Experimental design

To evaluate the proposed AR pipeline in REST, we first collected an eyes-closed EEG dataset in which a healthy subject performed a series of cued artifact-inducing actions comprising jaw clenching, scalp electrode tapping, head turning, and jumping. The subject rested for 2 minutes before performing each type of action for ten seconds, with 5-second inter-action intervals. Our goal was to evaluate the effects of the concomitant artifacts on ORICA convergence and to determine whether the proposed pipeline could mitigate those effects.

Next, we recorded two minutes during which the subject blinked at 1-sec intervals, followed by two minutes in which the subject performed lateral eye-movements using voluntary saccades at 1-sec intervals. Before and after each artifact period the subject rested with eyes closed for two minutes. We wanted to characterize the performance of the proposed pipeline in automatically identifying and rejecting eye-related ICs, thus clean the recording of eye-movement artifacts.

6.3.3 Dataset recording and analysis

The EEG data was recorded using a Cognionics Quick30 headset with a 500-Hz sampling rate using dry electrodes for which electrode impedances are in the range of hundreds of Ohms. Electrode P07 was excluded from the analysis because of a known cap hardware issue.

We processed both datasets using REST in a simulated real-time setting by rebroadcasting the data through the lab-streaming-layer [Kothe, 2014]. REST applied the proposed pipeline consisting of common-average re-referencing, FIR bandpass-filtering (1–50 Hz), ASR, ORICA, IC classification and rejection using EyeCatch, and channel data reconstruction. This pipeline is shown in Figure 6.1A. We processed the data from the first experiment, both with and without ASR; we processed the second dataset both with and without applying ORICA and EyeCatch.

To quantify the results of this analysis, we calculated the signal-to-noise ratio (SNR) after applying each different pipeline (e.g. with or without ASR). For zero-mean signals, SNR is the ratio of signal variance to noise variance, σ_s^2/σ_a^2. As we cannot claim to know those values exactly, we approximated the SNR by dividing the average channel variance during the rest periods, σ_s^2, by the average channel variance during artifact periods, σ_n^2. In reality,

$\sigma_n^2 \approx (\sigma_s + \sigma_a)^2$ since it reflects variability in the summed artifact and brain EEG activity. Therefore this SNR approximation is conservative and forms a lower bound on true SNR.

We computed the correlations of scalp maps from the ORICA decomposition to those from the offline Infomax ICA decomposition of the same data. We used offline Infomax ICA applied to the common-average referenced and FIR bandpass-filtered data with artifact periods manually removed as a gold-standard solution to compare against. Excluding the artifacts, these recordings are relatively stationary and so the offline solution is effectively the best-case solution possible for ORICA.

Finally, to get a sense of how ORICA learns under these different conditions, we also look at the dynamics of the non-stationary index (NSI) which quantifies the magnitude of the ORICA gradient over time [Hsu et al., 2016]. When data is very improbable under the current ORICA model, e.g. during transient artifacts, the NSI will have a large value relative to baseline.

6.4 Results and Discussion

6.4.1 Motion and muscle artifacts

ASR successfully removed a significant portion of the artifact-induced signal features in the first experiment. As shown in the top four frames of Figure 6.2, while not all of the artifact signals were removed by ASR, the exceptionally strong artifact periods were consistently reprojected to a more normal range for brain-dominated EEG. This effect can also be seen in Table 6.1 by the consistent rise in approximate SNR in the ASR and ASR-ICA columns as compared to the other two, which forgo ASR. Furthermore, Figure 6.3 indicates that ASR stabilized the ORICA decomposition in the presence of large-amplitude artifacts. This is indicated by the lower correlation values following the artifact events when ASR was omitted. In such cases, the ORICA model rapidly changed to try to better fit the artifact activity. When ASR preprocessing was added, there was little change in the ORICA model before and after artifact occurrences,

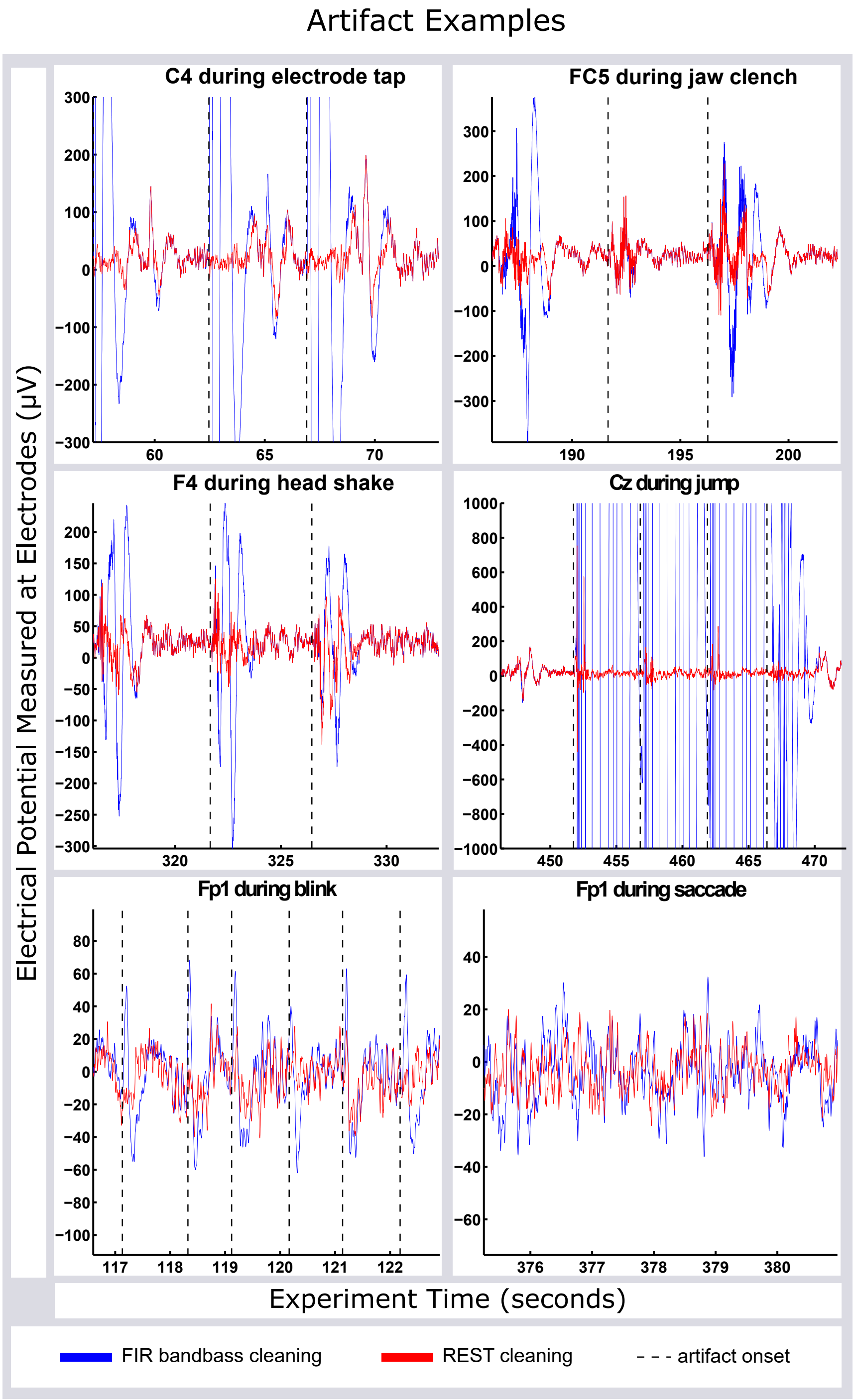


Figure 6.2: Examples of common EEG artifacts. From top to bottom and left to right: electrode tapping, jaw clenching, head shaking, jumping, blinking, and eye movements. The blue traces are cleaned with an FIR bandpass filter, while the red traces are further processed with ASR and ICA-based cleaning using ORICA and EyeCatch. Artifact onset times are indicated by black dotted lines.

Table 6.1: Signal to noise ratio (SNR) at distinct stages of processing for each artifact type tested.

Artifact	FIR	ASR	ASR-ICA	ICA
Electrode Tap	0.174	1.09	1.05	0.176
Jaw Clench	0.586	0.723	0.693	0.580
Head Shake	3.78×10^{-5}	0.349	0.339	3.78×10^{-5}
Jump	1.05×10^{-4}	0.529	0.511	1.05×10^{-4}
Blink	0.465	0.465	0.791	-
Saccade	0.498	0.810	1.01	-

demonstrating increased robustness to non-brain EEG “noise." This is particularly evident after subject jumps. Without ASR preprocessing, the entire model was lost after the jumps, i.e., not a single IC from the ORICA decomposition correlated highly with any ICs in the offline Infomax ICA solution. This is pivotal because, even though the ICA portion of the pipeline was not used in this experiment, typical EEG recording will have both eye movement-related artifacts as well as body motion artifacts. If the ORICA decomposition is lost every time a body motion artifact occurs (as it does without ASR preprocessing), then any eye movement-related ICs may not be found and could not then be cleanly removed from the data.

For transient artifacts, Table 6.1 shows the addition of ICA-based cleaning produced a minute decrease in SNR; likely because there were no stereotyped artifacts for ORICA to learn and remove. All ICA-based cleaning could do is find occasional false-positives, which in this case increased the power of the signal negligibly. The NSI traces in the bottom panel of Figure 6.3 show that ORICA follows the same general learning patterns with and without ASR, but exhibits more extreme NSI values without ASR preprocessing, as ORICA may be more directly exposed to effects of high-amplitude artifacts. It is worth noting that ORICA was able to find many brain-related ICs including those shown in the bottom-right of Figure 6.3. This suggests possible further uses of REST beyond data cleaning, in particular as a tool for real-time monitoring and source analysis, given the added robustness provided by ASR.

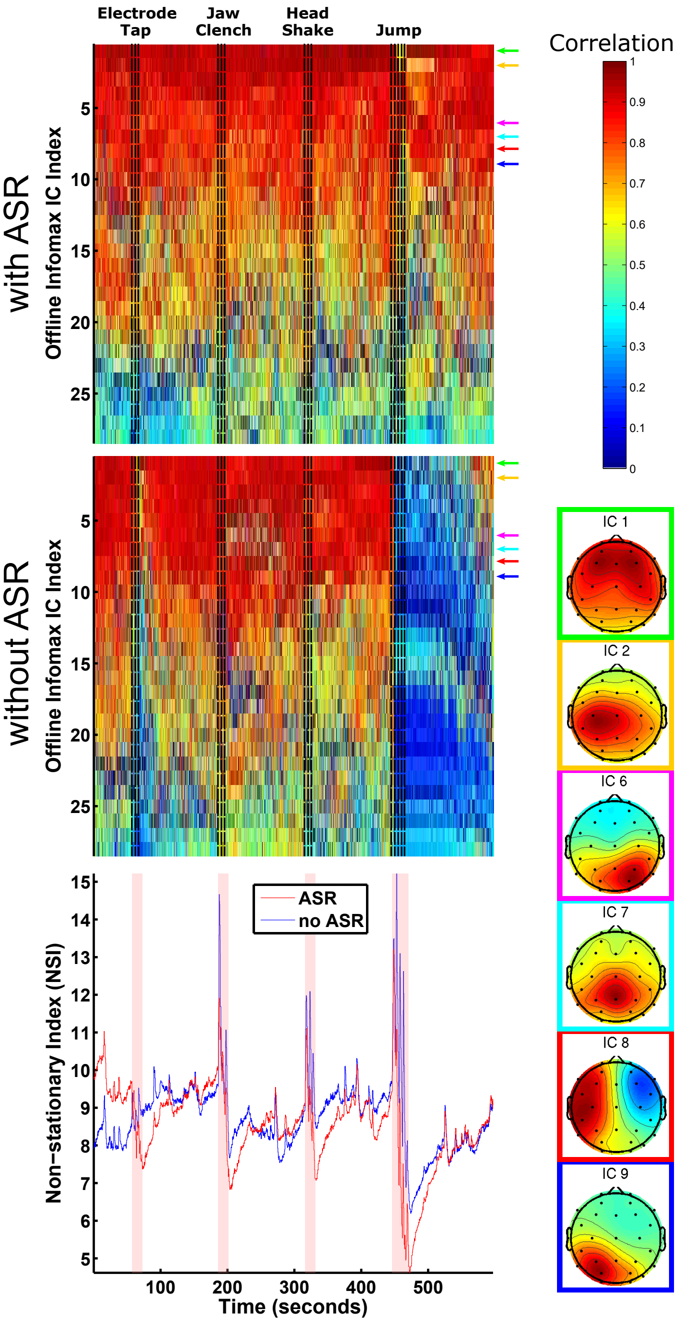


Figure 6.3: The two top images indicate how well, at different times during the transient artifact recording, the ORICA decomposition results match results of an offline Infomax ICA decomposition of the whole dataset. The top plot shows results of applying ASR before ORICA while the middle one does not use ASR. Example scalp maps for selected rows are shown, matched by border color to arrows pointing to the corresponding row in the plots. The bottom plot traces the non-stationary index values throughout the recording, both with and without use of ASR. In the top plots, artifact onsets are shown as black dashed lines while in the bottom plot they are shown as red lines.

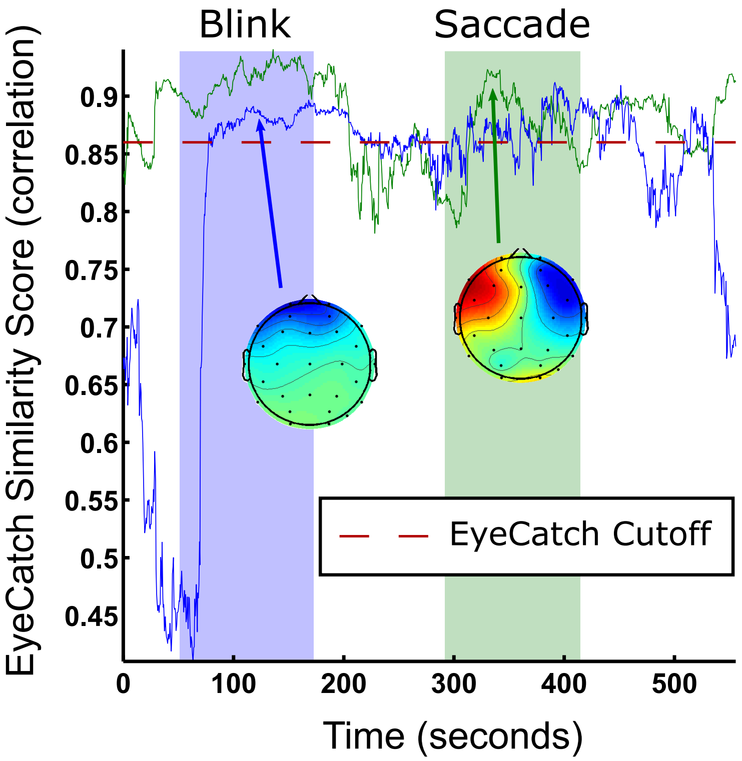


Figure 6.4: The above traces show how the eye-related independent components (IC) learned by ORICA were rated by EyeCatch. The threshold for removal, 0.86, is indicated as a red dashed line. The period where the subject was blinking is indicated by the blue shaded region while the period where the subject looked back and forth is indicated by the green shaded region. The subject kept his eyes closed during all other time periods (white background). The blink IC was found and removed quickly during blinking while the saccade IC did not remain suprathreshold during lateral eye-movement.

6.4.2 Eye-induced artifacts

For the eye movement-related artifacts in the second experiment, Table 6.1 indicates ASR had a negligible effect on eye-blink artifacts and a more significant effect on saccade artifacts. However, SNRs during both types of eye activity were further improved by the addition of ICA-based cleaning. The speed with which ORICA found the eye movement-related ICs is shown in Figure 6.4. Once the subject opened his eyes and began blinking, it took ORICA twenty-six seconds to converge well enough on the blink-related IC for EyeCatch to remove it. Even after two more minutes of eyes-closed resting, the maximum blink IC scalp map correlation remained near the EyeCatch threshold level and subsequently increased again when the subject reopened his eyes. However, the saccade-related IC EyeCatch score fluctuated across the rejection threshold as the subject performed lateral eye-movements resulting in incomplete saccade artifact rejection. It appears the altered version of EyeCatch used here was not ideal, as the changes seem to have introduced some instability in the correlations found, though no better option is currently available.

6.5 Conclusion

We have introduced a new pipeline for real-time EEG artifact removal that combines the use of ASR, ORICA, and an IC classifier (here EyeCatch) using the Real-time EEG Source-mapping Toolbox (REST). We studied how the pipeline performed in the presence of six different types of artifacts common in EEG recordings and found it removed the majority of the artifact-induced signal features. We also compared the performance of the pipeline with and without an initial application of ASR and found that the presence of ASR stabilized the ORICA decomposition, which is desirable for cleaning the data of eye movement-related artifact. The pipeline is available as part of REST, which is freely available at the url: https://github.com/goodshawn12/REST.

6.6 Acknowledgements

Chapter 6, in part, is a reprint of the material as it appears in Luca Pion-Tonachini, Sheng-Hsiou Hsu, Chi-Yuan Chang, Tzyy-Ping Jung, and Scott Makeig. Online automatic artifact rejection using the real-time EEG source-mapping toolbox (REST). In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 106–109. IEEE, 2018. The dissertation author was the primary investigator of this paper.

Chapter 7

Brain Network Analysis

7.1 Introduction

Up to here, we have developed and assessed methods which can be described as “data preparation” for the purpose of performing electroencephalogram (EEG) source analysis. Automating source analysis itself is a more difficult task because it is inherently unspecific. The analyses that are performed on datasets are largely dependent on the experimental tasks and the questions that researchers attempt to answer through those experiment. Certain categories of analysis are already facilitated by existing tools. For example, BCILAB [Kothe and Makeig, 2013] automates many types of brain-compute interface (BCI) paradigms which can be applied to data decomposed using independent component analysis (ICA). Many programs also exists for automatically applying statistical tests to EEG data such as EEGLAB [Delorme and Makeig, 2004, Delorme et al., 2011] and Statistical Parametric Mapping (SPM) [Penny et al., 2011]. Still, there remain many analyses which require extensive manual intervention to perform. This is especially true when experiments are not designed around repeated trials.

In this chapter, we provide an example of partially automated EEG analysis through unsupervised time series segmentation. We evaluate adaptive mixture ICA (AMICA) as a means to quantify continuously shifts in mental state. This can be performed as a stand-alone analysis or as an initial processing step by which meaningful features are extracted from the EEG data.

7.2 Background

An expanding focus in neuroscience has been on endogenous temporal dynamics of neural network activity that gives rise to fluidity and rapid adaptability in cognition and behavior. A growing body of evidence suggests that these temporal dynamics may arise from continual formation and dissolution of interacting cortical and allied subcortical source activities in large-scale brain regions whose joint electrical activities can be described as dynamic systems featuring continuous transitions between intermittently stable states [Chu et al., 2012, Betzel et al., 2012].

Earlier methods applied nonparametric statistical approaches that used EEG power spectral density, autocorrelation function, and entropy measures [Natarajan et al., 2004] to detect change points allowing segmentation of EEG into piecewise stationary processes [Kaplan et al., 2001]. Microstate analysis (see Khanna et al. [2015] for a review) takes the spatial distribution of electrodes into account and attempts to define quasi-stable "microstates" in terms of unique electric potential patterns across the multichannel EEG scalp electrode montage during behavioral states or resting states [Lehmann et al., 1987, Van de Ville et al., 2010]. The global functional connectivity approach [Chu et al., 2012, Betzel et al., 2012] measures inter-electrode channel signal synchrony to attempt to characterize brain states as stable functional networks. However, both the microstate and global connectivity models analyze scalp electrode signals that in themselves are highly correlated through common volume conduction and summation at the electrodes of potentials arising from brain and also non-brain sources (eye movements, ECG, etc.). The results of both methods have few or no interpretable connections to particular brain source activities that underlie the observed scalp phenomena. Hidden Markov Models (HMM) form another family of generative models with a rigorous temporal structure used to measure nonstationary functional connectivity.

Such models have been largely applied to source-space signals from MEG recordings [Baker et al., 2014, Vidaurre et al., 2016, 2017, Nielsen et al., 2017], where a source separation or localization step is prerequisite.

The recent study [Hsu and Jung, 2017] hypothesized that transitions to a different cognitive state may involve cortical macro- or meso-dynamics in new networks of cortical brain areas that can be identified by distinct ICA models trained on data recorded before and after the state transition, respectively. This hypothesis motivates the application of an extension of ICA – the ICA mixture model (ICAMM) [Lee et al., 2000] – an unsupervised learning approach to modeling EEG activities in different brain states and detecting brain dynamic state changes associated with cognitive state changes [Jung et al., 2000b]. The ICAMM assumes distinct ICA models may better characterize different segments of nonstationary data, i.e., $\boldsymbol{x}(t) = \boldsymbol{A}_h\boldsymbol{s}_h(t)$ where h is the model index. By allowing multiple ICA models to focus simultaneously on different parts of the data, ICAMM relaxes the spatial-stationarity assumptions and allows more total sources to be learned than the number of channels. ICAMM is thereby capable of modeling nonstationary, multi-state data and thus is a promising approach to studying dynamic changes in cognition and brain states. While the few prior attempts to apply ICAMM to EEG data were able to monitor attention [Jung et al., 2000b], to detect microarousals during sleep [Salazar et al., 2010a], and to detect mental state changes during a memory test [Safont et al., 2017], the full power of the ICAMM approach has not yet been demonstrated, including modeling of multiple brain states, tracking of state transitions in continuous recordings, consistency of the learned models across subjects, and more precise physiological interpretation of those models.

Here we report an EEG study using an unsupervised ICAMM to investigate dynamics of cognitive states. For this we chose adaptive mixture ICA (AMICA), proposed by Palmer et al. [2008], that adaptively learns individual source probability density functions (PDFs) as well as source scalp projection patterns. Palmer et al. [2008] has also provided an efficiently optimized algorithm for learning an ICAMM from multichannel data using a parallel implementation (the code is available at https://sccn.ucsd.edu/~jason/amica_web.html and also as an open source plug-in for EEGLAB [Delorme and Makeig, 2004] at https://sccn.ucsd.edu/wiki/EEGLAB_Extensions_and_plug-ins). In the following sections, we will show: 1) AMICA can learn the ground truth in the simulated quasi-stationary data – we test the effect of numbers of ICA models on AMICA performance; 2) AMICA usefully characterizes sleep EEG dynamics, producing consistent results across subjects that can be applied to classify six sleep stages; 3) AMICA can quantitatively assess subjects' continuous changes in attention and drowsiness levels during simulated driving and thereby can track brain dynamic state changes at single-trial level with millisecond resolution; and 4) AMICA provides interpretable models allowing computation of the spatial distribution and frequency content of active sources in each brain state.

7.3 Materials and Methods

7.3.1 Datasets and preprocessing

Dataset I: simulated quasi-stationary data

To systematically validate AMICA, we use the EEG data simulator in the Source Information Flow Toolbox (SIFT) [Delorme et al., 2011] to simulate a quasi-stationary dataset in which underlying sources are alternatingly active and inactive. With a 3-layer boundary-element-method (BEM) forward model, we obtain three 3-min segments of simulated 16-channel EEG data, each with a different set of 16 active super-Gaussian distributed sources. More details are included in Hsu et al. [2015].

Dataset II: CAP sleep database

We used 17 human EEG recordings, each consisting of 6–10 hours of sleep, from the CAP sleep database [Terzano et al., 2002] on PhysioNet [Goldberger et al., 2000]. Excluding subjects whose recordings had less than five channels gave seven EEG datasets from healthy subjects. We also used EEG recordings from ten patients with nocturnal frontal lobe epilepsy (NFLE), selected on the basis of data quality, i.e., longer data length, higher number of channels and more balanced numbers of sleep labels. We included NFLE patient recordings in an attempt to test the ability of the proposed approach to generalize across subjects and patients.

The EEG data comprise 6–13 bipolar channels (e.g. F3-C3, C3-P3, P3-O1, O1-A1, without common reference) affixed at scalp sites in the International 10–20 System and recorded with a sampling rate of 128 Hz using a Galileo System (Esaote Biomedica). After collection, the EEG signals were band-pass filtered between 0.5 Hz to 25 Hz. The hypnograms had been annotated by expert neurologists at 30-second intervals using standard Rechtschaffen and Kales (R&K) criteria into six sleep stages: wake (W), rapid eye movement (REM), and one to four non-REM sleep stages (N1, N2, N3 and N4). More detailed description of the data and their hypnograms can be found at https://physionet.org/pn6/capslpdb/.

Dataset III: drowsiness fluctuation in simulated driving

Ten healthy volunteers participated in a 90-min experiment in an immersive VR-based driving simulator, performing an event-related lane-departure task [Huang et al., 2009]. The subjects experienced visually presented lane-departure events every 8–12 seconds (with randomized event onset asynchronies) and were instructed to steer the car back to the cruising position quickly using a steering wheel. The duration between the onset of a lane-departure event to the onset of a responsive steering action was defined as subject reaction time (RT), which can be used to index degree of subject alertness / drowsiness [Lin et al., 2010]. The RT data were transformed to reaction speed (RS = 1/RT) to partially normalize the highly skewed RT distribution. For more details on the subjects and the experiment, refer to Lin et al. [2010].

For each subject, 30-channel EEG data were recorded with a 500-Hz sampling rate using a NeuroScan System (Compumedics Ltd., VIC, Australia) with electrode sites according to the

International 10–20 System. The EEG data were band-pass filtered (1–50 Hz) and downsampled to 250 Hz. Using the PREP pipeline [Bigdely-Shamlo et al., 2015], poorly recorded channels in the recordings, such as channels with flat signals arising from poor electrode contacts, and channels whose signals were poorly correlated with those of neighboring channels were removed. Two to six channels were so identified and removed for each of the ten subjects. In addition, artifact subspace reconstruction (ASR) [Mullen et al., 2015], implemented as a plug-in to the EEGLAB environment [Delorme and Makeig, 2004], was applied using a mild threshold (burst repair $\sigma = 20$) to reduce data contamination by high-amplitude artifacts. These artifact-correction methods were chosen to facilitate convergence of the ICA models. Detailed description of the data pre-processing can be found in Hsu and Jung [2017].

7.3.2 Method description

Comprehensive formulation of the ICAMM problem and detailed derivation of the AMICA algorithm have been presented in Lee et al. [2000] and Palmer et al. [2008] respectively. The following sections give a brief summary of the multi-model AMICA approach in an attempt to provide intuition and facilitate readers' understanding.

Adaptive mixture ICA (AMICA)

Figure 7.1 gives a schematic overview of the architecture of AMICA and its models. AMICA is, conceptually, a 3-layer mixing network: the top two layers constitute one or more ICA mixture models and the bottom layer, specific to AMICA, focuses each learned model on accounting for a subset of the data.

Starting from the top layer, the key assumption of multiple mixture models is that data $\boldsymbol{X} = \{\boldsymbol{x}(t)\}$ (N-channel by T-time samples) are nonstationary, so that different models may be dominant in characterizing the data at different times, i.e., $\boldsymbol{x}(t) = \boldsymbol{x}_h(t)$ where h is the model index. Previous studies have provided evidence that EEG activities during different brain states

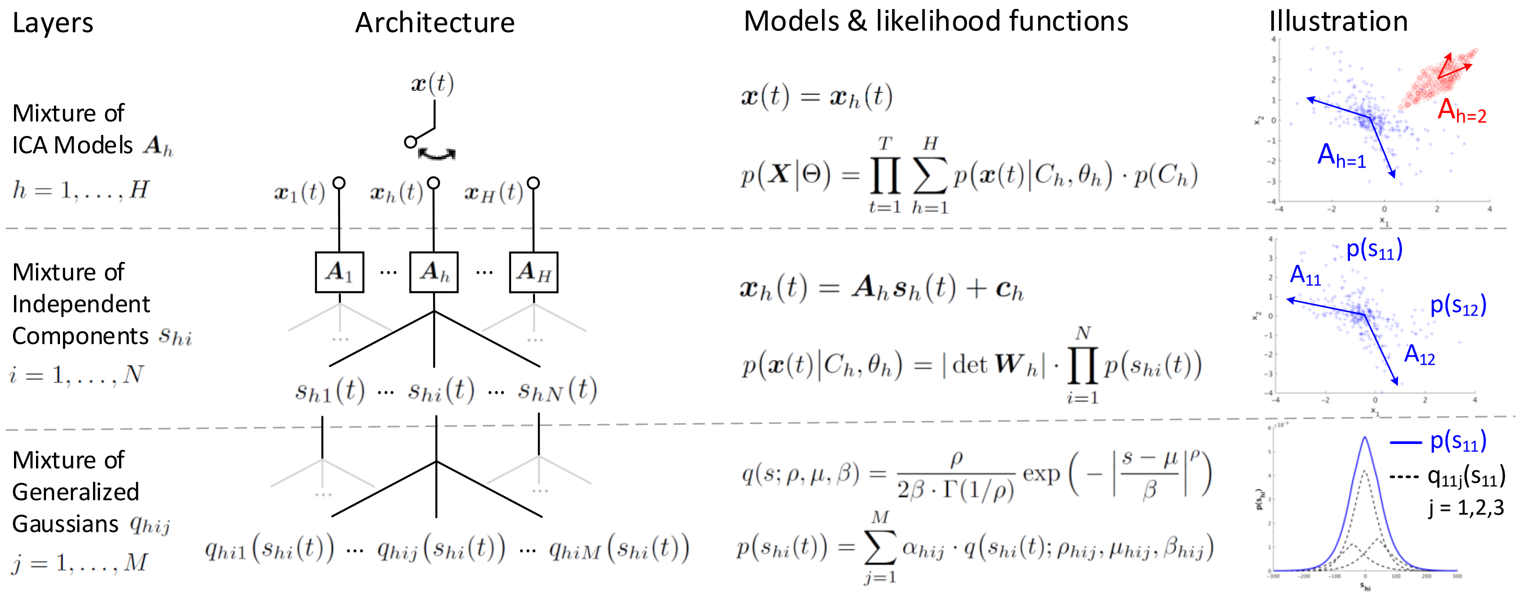

Layers
Architecture
Models & likelihood functions
Illustration
Mixture of ICA Models $\boldsymbol{A}_h$
$h = 1, \dots, H$
$\boldsymbol{x}(t)$
$\boldsymbol{x}_1(t)$ $\boldsymbol{x}_h(t)$ $\boldsymbol{x}_H(t)$
$\boldsymbol{x}(t) = \boldsymbol{x}_h(t)$
$p(\boldsymbol{X}|\Theta) = \prod_{t=1}^{T} \sum_{h=1}^{H} p(\boldsymbol{x}(t)|C_h, \theta_h) \cdot p(C_h)$
$A_{h=1}$
$A_{h=2}$
Mixture of Independent Components s_{hi}
$i = 1, \dots, N$
$\boldsymbol{A}_1$ ··· $\boldsymbol{A}_h$ ··· $\boldsymbol{A}_H$
$s_{h1}(t)$ ··· $s_{hi}(t)$ ··· $s_{hN}(t)$
$\boldsymbol{x}_h(t) = \boldsymbol{A}_h \boldsymbol{s}_h(t) + \boldsymbol{c}_h$
$p(\boldsymbol{x}(t)|C_h, \theta_h) = |\det \boldsymbol{W}_h| \cdot \prod_{i=1}^{N} p(s_{hi}(t))$
$p(s_{11})$
A_{11}
$p(s_{12})$
A_{12}
Mixture of Generalized Gaussians q_{hij}
$j = 1, \dots, M$
$q_{hi1}(s_{hi}(t))$ ··· $q_{hij}(s_{hi}(t))$ ··· $q_{hiM}(s_{hi}(t))$
$q(s; \rho, \mu, \beta) = \frac{\rho}{2\beta \cdot \Gamma(1/\rho)} \exp\left(-\left|\frac{s-\mu}{\beta}\right|^{\rho}\right)$
$p(s_{hi}(t)) = \sum_{j=1}^{M} \alpha_{hij} \cdot q(s_{hi}(t); \rho_{hij}, \mu_{hij}, \beta_{hij})$
— $p(s_{11})$
---- $q_{11j}(s_{11})$
$j = 1,2,3$

Figure 7.1: Adaptive mixture ICA (AMICA) in a nutshell. AMICA consists of three layers of mixing. As shown in the illustration, the first layer is a mixture of ICA models $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ that learn the underlying data clusters, simulated based on Laplace and uniform distributions respectively. The second layer is a mixture of independent components $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_{12}$ that decompose the data cluster into statistically independent sources' activations $\boldsymbol{s}_{11}$ and $\boldsymbol{s}_{12}$. The third layer is mixture of generalized Gaussian distributions q_{11j} that approximate the probability distribution of the source activation $p(\boldsymbol{s}_{11})$.

(e.g. alert versus drowsy) are nonstationary and can be modeled by a finite set of distinct ICA models [Hsu and Jung, 2017].

In the top two layers of the AMICA network, a standard ICA model is employed to model the data $\boldsymbol{x}$ as an instantaneous linear mixture $\boldsymbol{A}$ ($N \times N$ matrix) of statistically independent components $\boldsymbol{s}$, i.e., $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$. The first two layers consist of the ICA mixture model:

$$\boldsymbol{x}(t) = \boldsymbol{x}_h(t) = \boldsymbol{A}_h \boldsymbol{s}_h(t) + \boldsymbol{c}_h, \quad h = 1, \ldots, H \tag{7.1}$$

where $h = h(t)$ and $\boldsymbol{A}_h$ is the dominant or active model at time t with source activities $\boldsymbol{s}_h(t)$ and bias $\boldsymbol{c}_h$. For simplicity, it is assumed that only one of the H models is active at each time and that the model index h and the data $\boldsymbol{x}(t)$ are temporally independent. Hence the likelihood of data given the ICA mixture model can be written as:

$$p\big(\boldsymbol{X}\big|\Theta\big) = \prod_{t=1}^{T} \sum_{h=1}^{H} p\big(\boldsymbol{x}(t)\big|C_h, \theta_h\big) \cdot p(C_h) \tag{7.2}$$

where $\Theta = \{\theta_1, \ldots, \theta_H\}$ contains the parameters of ICA models and $p(C_h) = \gamma_h$ is the probability of the h-model being active that satisfies $\sum_{h=1}^{H} \gamma_h = 1$.

Given the assumption of statistical independence between components $s_{hi}(t)$ ($i = 1, \ldots, N$), the likelihood of the data given the active ICA model $\boldsymbol{W}_h = \boldsymbol{A}_h^{-1}$ is:

$$p\big(\boldsymbol{x}(t)\big|C_h, \theta_h\big) = |\det \boldsymbol{W}_h| \cdot \prod_{i=1}^{N} p\big(s_{hi}(t)\big) \tag{7.3}$$

In the third layer of the AMICA network, the probability density function (PDF) of each component $p\big(s_{hi}(t)\big)$ is approximated by a mixture ($j = 1, \ldots, M$) of generalized Gaussian distributions $q(s)$ [Palmer et al., 2006, 2008]:

$$p\big(s_{hi}(t)\big) = \sum_{j=1}^{M} \alpha_{hij} \cdot q\big(s_{hi}(t); \rho_{hij}, \mu_{hij}, \beta_{hij}\big) \tag{7.4}$$

where α_{hij} is the weight for each PDF. The generalized Gaussian distribution parameterized by shape ρ, scale β and location μ is defined as:

$$q(s;\rho,\beta,\mu) = \frac{\rho}{2\beta\cdot\Gamma(1/\rho)}\exp\Big(-\Big|\frac{s-\mu}{\beta}\Big|^{\rho}\Big) \tag{7.5}$$

It is worth noting that most standard ICA mixture models, in contrast to AMICA, assume predefined PDFs for sub-Gaussian and super-Gaussian sources [Lee et al., 2000]. A previous study has shown that by adaptively learning the PDFs for each source, AMICA can achieve higher mutual information reduction while also returning a larger number of biologically interpretable dipolar sources than other ICA approaches when applied to real 70-channel EEG data [Delorme et al., 2012].

In the 3-layer AMICA mixing network, the parameters to be estimated are

$$\Theta = \Big\{\boldsymbol{W}_h,\;\; \boldsymbol{c}_h,\;\; \gamma_h,\;\; \alpha_{hij},\;\; \beta_{hij},\;\; \rho_{hij},\;\; \mu_{hij}\Big\}$$

that correspond to the model index $h = 1,\ldots,H$, the component index $i = 1,\ldots,N$ and the PDF index $j = 1,\ldots,M$. The next section describes an efficient approach to estimating these parameters.

Parameter estimation and interpretation

The expectation-maximization (EM) algorithm is employed to estimate the parameters $\hat{\Theta}$ that maximize the data likelihood function in Equation 7.2. The algorithm consists of two-step iterative learning involving alternating E-steps and M-steps. The E-step uses Equation 7.3 and Equation 7.4 to construct the expectation of the likelihood function in Equation 7.2 using current estimates of the parameters $\hat{\Theta}^l$. The M-step maximizes the likelihood function returned by the preceding E-step. Instead of using standard or natural gradient approaches [Lee et al., 2000, Salazar et al., 2010b], AMICA uses the Newton approach as derived by Palmer et al. [2008]

based on the Hessian (matrix of second-order derivatives) to achieve quadratic, and thus faster, convergence. For a detailed derivation and learning rules, see Palmer et al. [2008].

As an unsupervised approach with generative models, the Θ parameters learned by AMICA provide rich information about the underlying data clusters and their temporal dynamics. As illustrated in Figure 7.1, ICA models $\boldsymbol{W}_h$ and $\boldsymbol{c}_h$ can characterize distinct data clusters that represent different quasi-stationary states in the data. In addition, the corresponding source activations $\boldsymbol{s}_{hi}$ can be better estimated by α_{hij}, β_{hij}, ρ_{hij}, and μ_{hij} instead of assuming a fixed PDF as in many other ICA models including the original Infomax ICA [Bell and Sejnowski, 1995]. Furthermore, the activation of each ICA model $h(t)$ can be represented as the likelihood of the data sample $\boldsymbol{x}(t)$ given the estimated parameters of the model θ_h, using Equations 7.2, 7.3, and 7.4:

$$L_{h(t)} = p(C_h) \cdot |\det \boldsymbol{W}_h| \cdot \prod_{i=1}^{N} \sum_{j=1}^{M} \alpha_{hij} \cdot q\big(s_{hi}(t); \rho_{hij}, \mu_{hij}, \beta_{hij}\big) \tag{7.6}$$

Therefore the probability of activation of each ICA model at time t can be calculated by normalizing $L_{h(t)}$ across all models:

$$p\big(h(t)\big) = L_{h(t)} \Big/ \sum_{h=1}^{H} L_{h(t)} \tag{7.7}$$

This value, $p\big(h(t)\big)$, characterizes the temporal dynamics of activations of distinct states modeled by ICA and is referred to as "ICA model probability" in following sections.

Application of multi-model AMICA

Multi-model AMICA decompositions were applied to all datasets described in Section 7.3.1 with the parameters specified in Table 7.1. For datasets II and III, rejection of data samples based on their posterior probabilities was applied to alleviate the effects of transient artifacts, such as data discontinuities, that might disrupt ICA learning. In addition, a sphering transformation of the EEG data (i.e., inverse matrix square root of the EEG covariance matrix) was applied prior to AMICA decomposition to facilitate the learning process. An efficient implementation of AMICA

Table 7.1: AMICA Parameters

Attributes	Datasets		
	I	II	III
Models (H)	2–6	8	2–4
Sources (N)	16	6–13	24–28
PDF components (M)	3	3	3
Rejection steps	0	15	15
Rejection thresholds	N/A	3	3
Max learning steps	2,500	2,000	2,000

with parallel computing capability by Palmer et al. [2008] was used in this study. The code for that implementation is available at https://sccn.ucsd.edu/~jason/amica_web.html and also as an open source plug-in for EEGLAB [Delorme and Makeig, 2004] at https://sccn.ucsd.edu/wiki/EEGLAB_Extensions_and_plug-ins.

7.3.3 Validation and quantitative analyses

Decomposition errors of ICA models

To determine whether AMICA could accurately decompose the simulated quasi-stationary data, three different measures were employed: model errors for unmixing matrices $\boldsymbol{W}_h$, the signal-to-interference ratios (SIR) for source activities $\boldsymbol{s}_h$, and the symmetric Kullback-Leibler (KL) divergence [Kullback and Leibler, 1951] for parameters of the source probability densities $\Theta = \{\alpha, \beta, \mu, \rho\}$. The model error quantifies the normalized total cross-talk errors that account for scale and permutation ambiguities. In the case of perfect reconstruction, the model error equals zero. The SIR estimates the log-scaled normalized mean-squared errors of the decomposed time series of the component, compared to the corresponding ground-truth source activities. KL divergence measures the difference between the estimated and ground-truth source PDFs.

Classification of sleep stages

To quantitatively assess results of unsupervised segmentation of the sleep EEG data by AMICA decomposition, we used ICA model probabilities (Equation 7.7) as features and applied a Gaussian Bayes classifier to 30-second data windows to classify the data into six sleep stages. The Gaussian Bayes classifier models the features of each class as a multivariate Gaussian distribution, $G(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the covariance matrix estimated from the training data with the same class. To classify a test data window, the classifier compares the posterior probabilities of each class given the test data $\boldsymbol{x}$:

$$C_k = \arg\max_k p(C_k|\boldsymbol{x}) = \arg\max_k G(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\cdot p(C_k) \tag{7.8}$$

where $p(C_k)$ is the prior distribution of class k. In this study, the relative proportion of labels in each class is used as the prior distribution.

Five-fold cross-validation was performed for each subject data set. To ensure each fold had enough training data for each class, the data were first pooled according to their labels and then divided into five folds. The cross-validation accuracy and the confusion matrix were computed and the results summarized across subjects. The effect of the number of features used, i.e., model probabilities, on classification accuracy was also tested. It is worth noting that the current cross-validation approach was applied to the model probabilities of the AMICA decompositions on the combined training and testing data. Also, a generative classifier like the Gaussian Bayes classifier was here employed not to produce optimal classification accuracy but to illustrate the separability of EEG activities into six sleep stages using the feature space learned by AMICA decomposition.

Distinguishing alert versus drowsy behavior

A relational analysis was performed for dataset II (Section 7.3.1) to quantitatively evaluate the relationship between ICA model probabilities and drowsiness level as indexed by decreased reaction speed to driving challenges introduced into a simple driving simulation. Here, AMICA model probabilities were first computed for 5-second data windows immediately preceding onsets of lane-departure events (as might be produced during actual driving by unseen cross-winds). Pearson correlation coefficients were computed between preceding model probabilities and reaction speeds across all driving challenge trials. To assess longer-lasting fluctuations in behavioral drowsiness level over a driving session, a 90-sec smoothing window was applied [Makeig and Inlow, 1993]. Median reaction speed and model probabilities were computed across the 5–10 trials in each 90-sec window. The effects of model probability smoothing length is discussed in Hsu and Jung [2017].

Clustering ICA models across subjects

To examine the consistency of the learned AMICA models across subjects, we established template models defined by their relative model-dependent sleep-stage probabilities and matched each subject's models to the template models using iterative template-matching. Mean model probabilities were obtained for each combination of subject, model, and sleep stage to generate a matrix, P_i, of six stages (rows) by eight models (columns) for each subject, i, normalizing each column to sum to one. To begin, a subject was selected at random and the corresponding subject matrix P_i was used as the initial template, $P_0^{(1)}$. The AMICA models for each subject were greedily matched with template models by iteratively selecting model pairs with maximal Pearson correlation above a threshold of 0.9 using $matcorr()$ from EEGLAB [Delorme and Makeig, 2004]. The matched AMICA models (columns of model probabilities across sleep stages) were averaged over the N subjects to obtain a new template $P_0^{(2)} = \frac{1}{N}\sum_i^N \hat{P}_i$ in place of $P_0^{(1)}$ and subject models that did not exceed the correlation threshold were ignored when approximating the next template.

The above template-matching process was iterated until the total absolute difference between new and old templates was smaller than a predefined threshold, i.e., $\sum_i \sum_j |P_{0,ij}^{(t+1)} - P_{0,ij}^{(t)}| \leq \varepsilon$ for t-th iteration. This study used $\varepsilon = 0.1$ to ensure that the results were consistent regardless of the choice of template subject.

Clustering independent components across subjects

Clustering of independent components (ICs) was performed to identify across-subject IC equivalences within model classes. The independent component (IC) clusters were obtained using the CORRMAP plug-in [Viola et al., 2009] to EEGLAB using component similarity assessed by scalp map correlations with IC templates. An IC scalp map is a vector of relative contribution or projection weights of the IC source to the scalp channels. The IC templates were selected visually with the constraint that each template IC must be well modeled by a single equivalent dipole model (i.e., a dipolar source, whose scalp map has small residual variance (10%) from the projection of the best-fitting dipole model) using the EEGLAB plug-in DIPFIT (version 2.3) [Oostenveld et al., 2011], evidenced by the observation that independent EEG sources are typically dipolar [Delorme et al., 2012]. The number of ICs contributed by each subject was limited to two for the centro-occipital cluster and one for the other clusters. The following correlation thresholds were used: 0.9 for eye-blink and eye-movement clusters; 0.85 for the other clusters. These parameters were carefully chosen to avoid assignments of near-duplicate ICs to different clusters and to reduce variability produced by template selection.

7.4 Results

7.4.1 Dataset I: Validation using simulated data

Automatic data segmentation by ICA model probability

Figure 7.2 shows mean and upper/lower-bound model probabilities of the model clusters, smoothed using a 1-sec window, across 100 repeated runs each decomposed using 3-, 4-, 5- and 6-model AMICA. All the 3-, 4-, 5- and 6-model AMICA decompositions successfully segregated data within the three simulated quasi-stationary segments, assigning them distinct ICA models (those with the highest probabilities, here labeled M1-M3).

Variability in model cluster probabilities across simulations, indicated by the heights of the shaded regions representing the 90th and 10th percentiles of the probability distributions, increased as the numbers of models used were larger than the simulated ground truth (3 models). For these (over-complete) mixture model decompositions, model clusters M4-M6 were more probable than model clusters M1-M3 only in small portions (3% to 7%) of the data. Under-complete 1-model and 2-model AMICA decompositions (Figure 7.2) tended to model ground truth in one or two of the three simulated data segments. Overall, complete and over-complete AMICA decompositions accurately segmented the nonstationary simulated data in an unsupervised manner.

AMICA decomposition errors

Figure 7.3A shows that 3- and 4-model AMICA decompositions achieved model errors comparable to the combined results of Infomax ICA decompositions of the single-model data segments (difference probability, $p = 0.11$ by unpaired t-test), demonstrating the three ground-truth mixing matrices could be learned accurately by AMICA without identified model boundaries. By comparison, the performance of 5-model AMICA was slightly worse and model errors for 6-model AMICA were significantly higher for 3- and 4-model AMICA decompositions. Nevertheless, 6-model AMICA still outperformed under-complete 1-model and 2-model AMICA

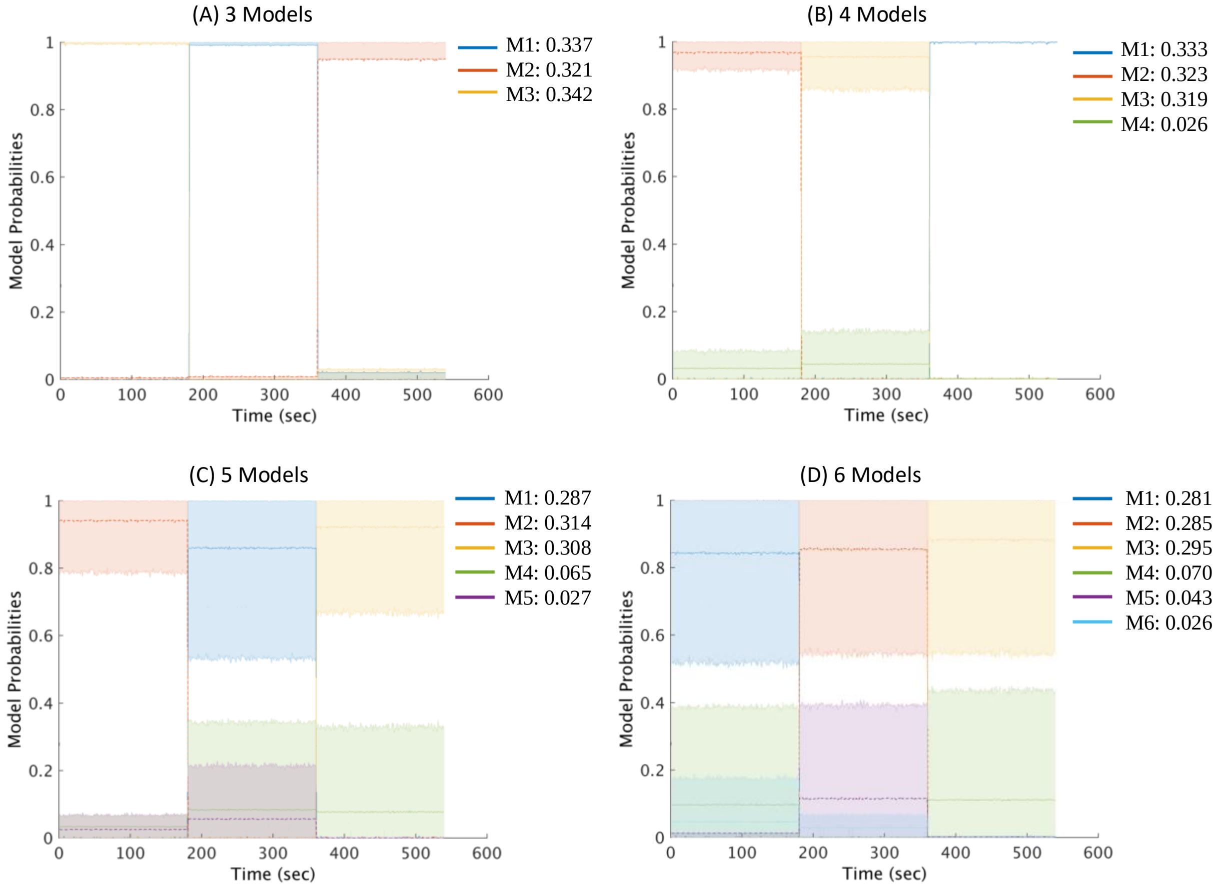


Figure 7.2: Mean changes in AMICA model probabilities clustered across AMICA decompositions of 100 repeated runs applied to the simulated quasi-stationary data. (A) AMICA decompositions using three models, (B) four models, (C) five models, and (D) six models. Upper and lower edges of the shaded regions represent the 10th and 90th percentiles of the cluster normed probability distribution. Figure legends give the mean probabilities $p(C_h)$ for each model cluster.

decompositions applied to the three data model segments.

Figure 7.3B shows that 3- and 4-model AMICA decompositions gave the highest SIR, 5- and 6-model decompositions marginally lower and 1- and 2-model AMICA decompositions still lower SIR. Both 3- and 4-model AMICA decompositions achieved SIR results comparable to Infomax ICA run on the single-model data segments ($p = 0.24$ and 0.14 respectively). These results show that the ground-truth source activities for each model segment were well reconstructed by complete (or, here, slightly over-complete) AMICA decompositions.

Figure 7.3C shows that 3- to 6-model AMICA decompositions produced the smallest (on average, near-zero) KL divergence values, suggesting that the source probabilities densities were also properly approximated. Here, 2-model AMICA performed slightly worse ($p < 0.05$) and 1-model AMICA much worse.

In summary, 3-model AMICA decomposition could simultaneously and accurately learn the true mixing matrices, source activities, and probability densities for three independent component models used to simulate 3-segment quasi-stationary data. AMICA performance using an unsupervised learning approach was comparable to Infomax ICA applied to each segment separately in a supervised fashion. Further, slightly over-complete (4-model) AMICA decompositions produced nearly comparable results, and performance only marginally decreased as the number of AMICA models was further increased.

7.4.2 Dataset II: classify sleep stages

We applied 8-model AMICA to 17 sleep EEG datasets to evaluate AMICA performance applied to actual EEG data and to assess its capability to distinguish the six conventional sleep stages from the data themselves without regard to changes in spectra or other time series properties.

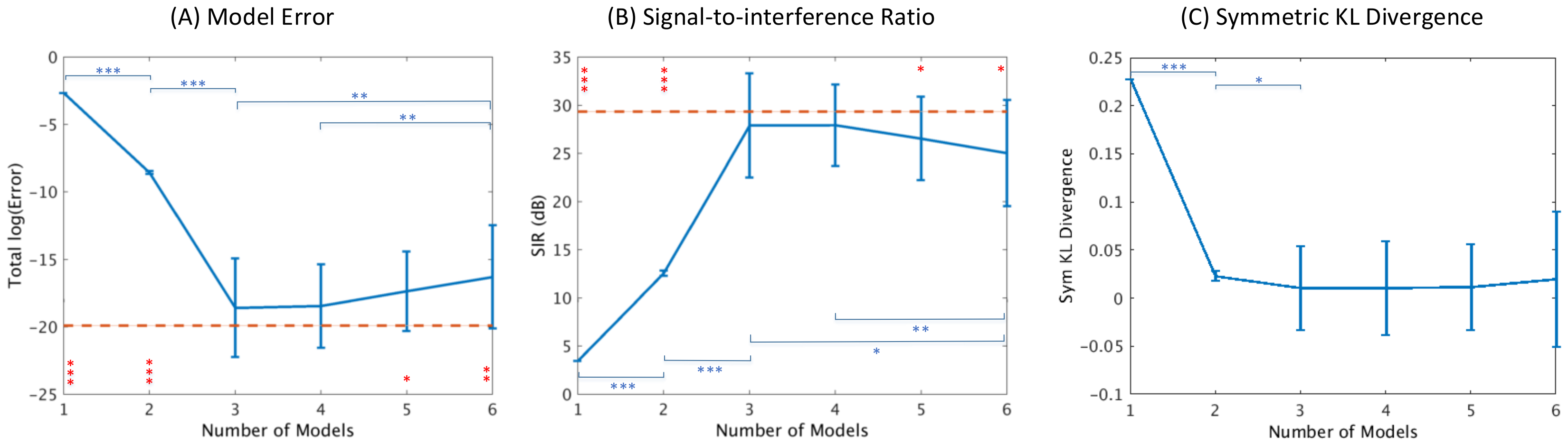


Figure 7.3: (A) Model errors in the learned model unmixing matrices versus simulated ground truth, (B) signal-to-interference ratios (SIR) of the decomposed model source activities, and (C) symmetric KL divergence of the learned source probability densities for AMICA decompositions using 1–6 models each averaged across 100 simulations. Red dashed lines indicate the performance of 1-model Infomax ICA applied to each of the known data segments (whereas AMICA has to learn the segmentation). Significant differences in unpaired t-tests are shown (* $p < 0.01$, ** $p < 1 \times 10^{-4}$, *** $p < 1 \times 10^{-6}$). Red asterisks denote comparisons between AMICA and Infomax ICA models; blue asterisks denote comparisons between AMICA model orders. Overall, model errors were lowest for veridical (3-model) and slightly over-complete (4-model) AMICA decompositions.

Model probabilities characterize sleep dynamics

To illustrate the temporal dynamics learned by 8-model AMICA from the sleep EEG data, Figure 7.4 shows the sleep stages annotated by experts and the probabilities of AMICA models ordered by overall data likelihood in one sleep session. Four distinct patterns of model probability changes were observed: (1) Models M1 and M2 were relatively active, i.e., had high model probabilities, during light sleep (N1 and N2) and had low probabilities during deep sleep (N4). Model M1, however, was more probable in rapid eye movement (REM) sleep than model M2. (2) In contrast, both models M3 and M5 were active only during deep sleep (N3 and N4). These first two patterns sufficiently characterized changes from light sleep to deep sleep and back again (red-shaded regions) over the course of the sleep session. (3) Model M4 was most probable (gray-shaded regions) during REM sleep and in the wake state. (4) Probabilities of models M6, M7 and M8 rose only sporadically, mainly in the wake state.

Thus, the probabilities of the eight learned ICA models for this session had notable relationships to the annotated sleep stages, but ICA model probabilities could not be mapped one-to-one with sleep stages. Some ICA models appeared to jointly characterize a sleep stage (e.g. M1 and M2 for N2, and M3 and M5 for N4), while probabilities for other models rose in different sleep stages (e.g. M1 probability rose briefly during N1, N2 and REM stages).

The dynamics of the model probabilities suggested that the changes in EEG activities during transitions between sleep stages were continuous as opposed to discrete – unlike as indicated by the hypnogram (scored by convention in successive 30-second intervals). Transition times varied across sleep stages. For example (red-shaded regions), major model probability shifts for models M1, M2, M3 and M5 had slower transitions (5–10 min) from stage N2 to N4 than from N4 to N2 (2–5 min). Some model probability transitions began before changes in the annotated sleep-stage labels. These results provide compelling evidence that AMICA model probabilities might be used to study the dynamics of EEG changes during sleep at much finer (e.g. approaching sample-by-sample) temporal resolution than offered by standard sleep scoring.

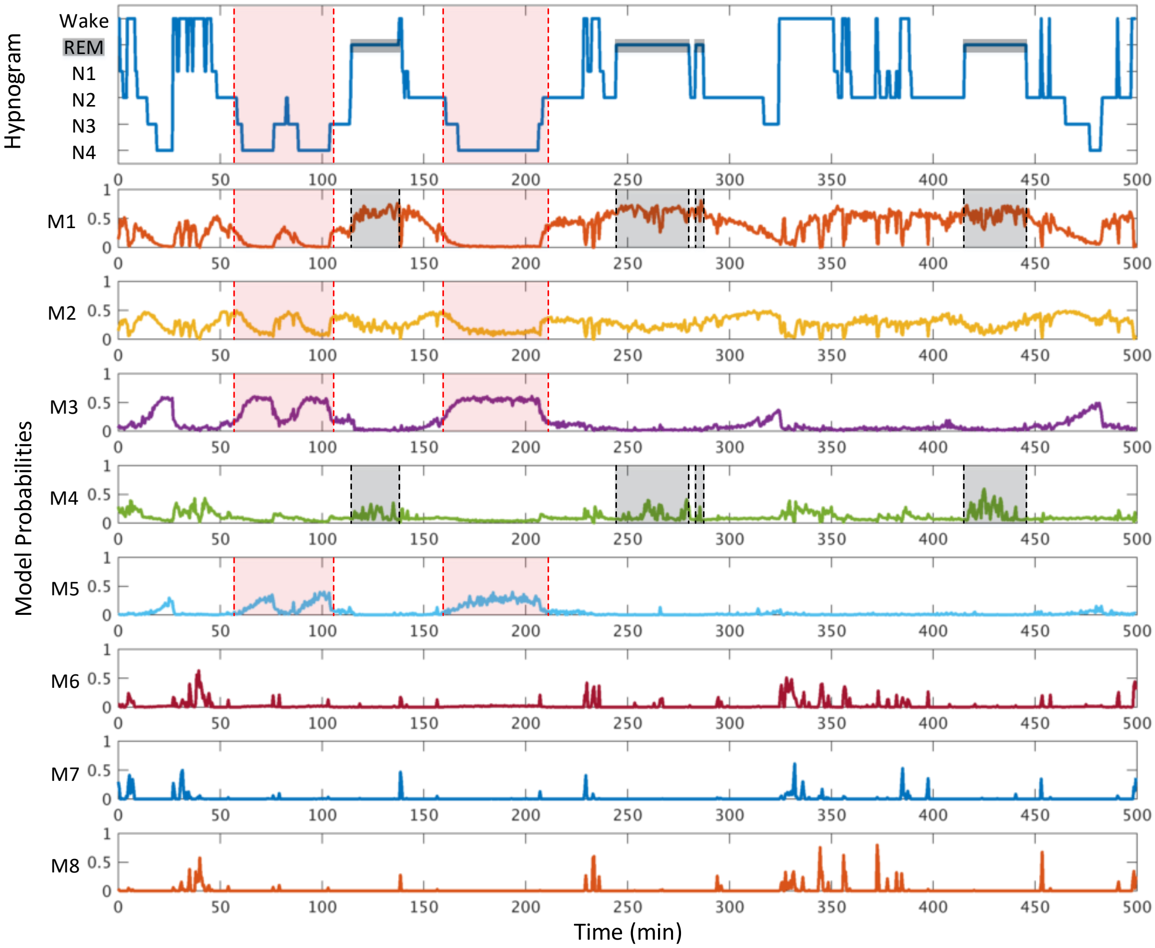


Figure 7.4: The top panel shows the hypnogram, i.e., sleep stages annotated from the EEG record by a sleep expert, of a sleep session from a single subject. Bottom panels show mean probabilities, within each 30-sec sleep scoring interval, of ICA models learned by an 8-model AMICA decomposition applied to the EEG record. Red-shaded regions highlight changes in model probabilities for relevant models during transitions to and periods of deep sleep (N4). Gray-shaded regions highlight probability value changes for relevant models during REM sleep.

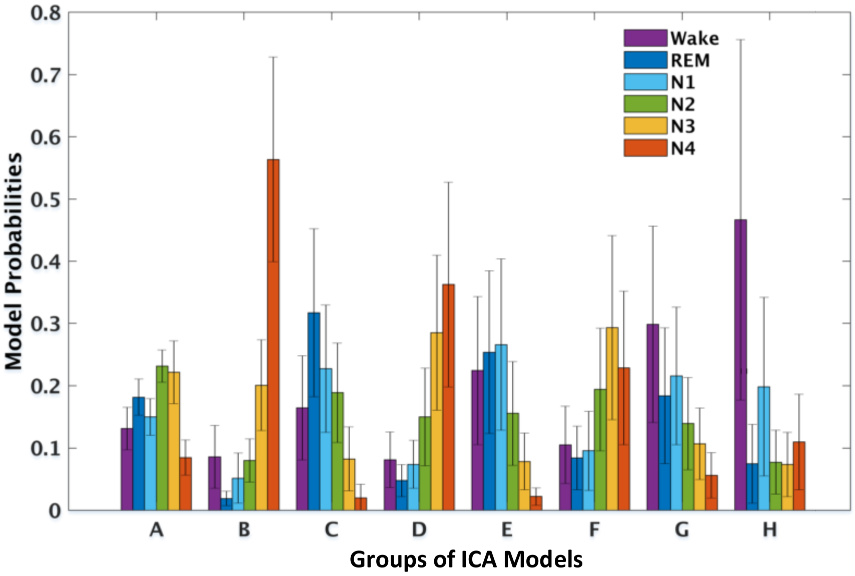


Figure 7.5: Cross-subject mean (plus one standard deviation) model probabilities of eight AMICA model clusters in six sleep stages. Model clusters were composed of best-matched models across subjects, as found by iterative template matching.

Relationships between ICA models and sleep stages across subjects

Next, we explored relationships between ICA models and sleep stages to assess if these relationships could be generalized across subjects using iterative template-matching of models from different subjects (Section 7.3.3). Figure 7.5 shows that ICA model clusters across subjects could be built based on relationships between data-driven model probabilities and annotated sleep stages. Resulting standard deviations of cluster model probability in each sleep stage were surprisingly small. Furthermore, each AMICA model cluster probability profile across sleep stages was distinct. For example, model A was relatively active in lighter sleep (N2 and N3), models B and D in deep sleep (N4), models C, E, and F in REM and stages N1 and N3, respectively. Models G and H were most probable during the wake state.

To visualize relationships between ICA model probabilities and sleep stages, Figure 7.6 presents 30-sec window-mean model probabilities for model clusters A, B, and C (cf. Figure 7.5) for all 17 subjects. Model probability values in the different (color-marked) sleep stages are

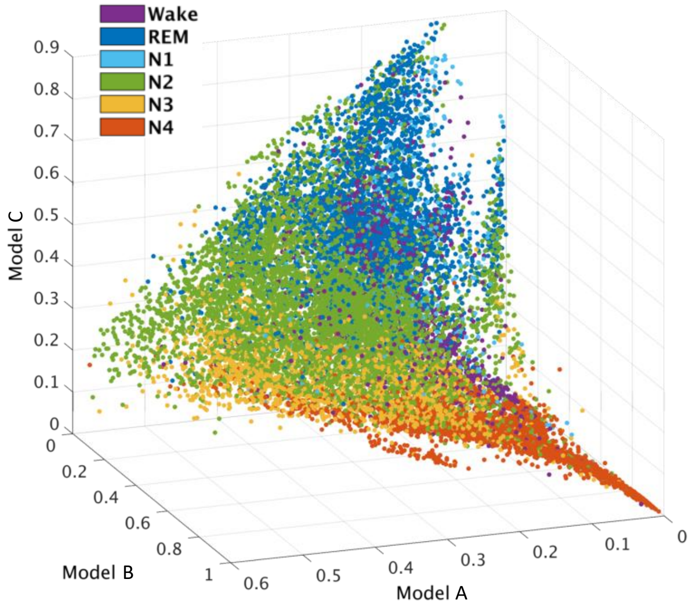


Figure 7.6: Scatter plot of window-mean model probabilities for AMICA model clusters A, B, and C (cf. Figure 7.5); each point representing mean model probability within a 30-sec data segment from sleep recordings of seven healthy subjects and ten patients. Colors represent expert-designated sleep-stage labels for the same data segments. Note the distinct deep sleep (N4) pattern and the relative closeness of wake and REM sleep characteristics.

clearly separated in this feature space. The progression from light sleep (N2, green) to deeper sleep (N3, yellow, to N4, red) is associated with smooth changes in cluster model probabilities. Model probabilities in the (purple) wake state were mostly low (near the *(0,0,0)* corner). These characteristics were consistent across AMICA models from seven healthy subjects and ten patients with nocturnal frontal lobe epilepsy.

Quantitative analysis: classification accuracy

To quantitatively assess the potential utility of model probabilities for separating sleep stages, we entered the window-mean model probabilities from the 8-model AMICA decomposition into a Gaussian Bayes classifier that fits a Gaussian distribution of 8-model probability vectors for each of the six annotated sleep stages (Section 7.3.3), and measured classification accuracy using 5-fold cross validation for each subject.

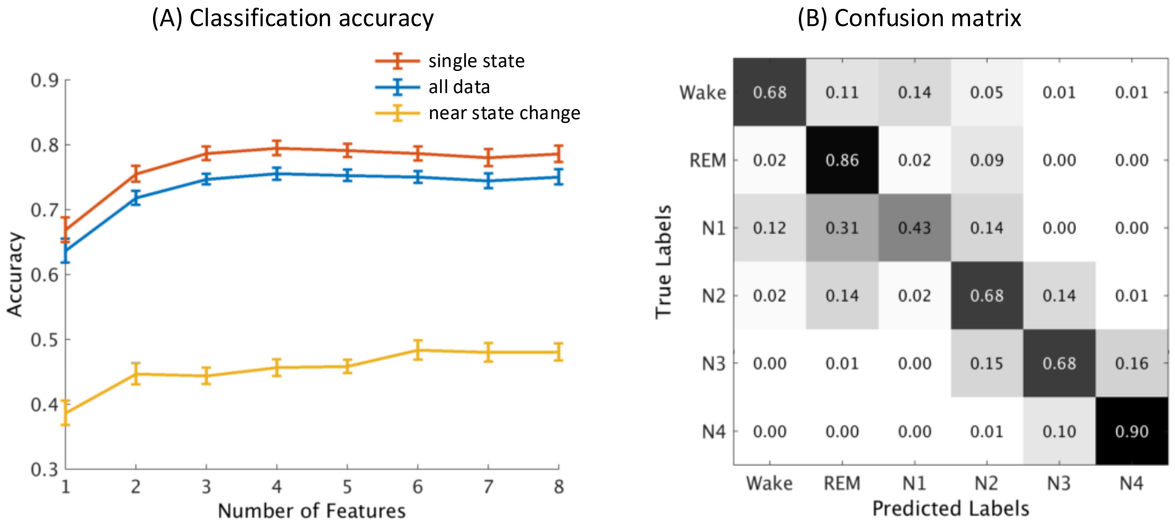


Figure 7.7: (a) Means and standard deviations of classification accuracy between six sleep stages across the 17 subjects using cluster model probabilities for different numbers of models as features. Results were separated into two conditions, depending on whether the data window was or was not near a sleep state change. (b) Confusion matrix of 6-class classification across all the data using eight cluster model probability features.

Figure 7.7A shows classification accuracy across all subjects. Accuracy improved when the number of model clusters was increased up to the use of the first three clusters. For all data (blue curve), mean accuracy was 74% - 76% when using three or more cluster model probabilities as features (no significant difference was observed by paired t-test). Classification accuracy was much lower (to 45% - 49%, yellow curve) for 30-sec data windows near a state change (e.g. when the sleep-stage label was different from that of the previous or succeeding windows). Accuracy was higher (78% - 80%, red curve) when the window was not near a state change.

Note that classification accuracy was biased by the unbalanced class sample sizes. Figure 7.7B shows the sleep-stage confusion matrix for the classification using all eight model cluster probabilities. For the most distinctive sleep stages (REM and N4), the sensitivity (true positive rates) were 86% and 90%. For sleep entry stage N1 (with fewer class samples), sensitivity was significantly lower (43%), in line with clinical expectation. In addition, misclassification between sleep stages shown as nearest neighbors in Figure 7.7C accounted for 87% of the total errors.

7.4.3 Dataset III: estimating behavioral alertness

Given the results using multi-model AMICA decompositions on sleep stage classification, described in the previous section, we assessed whether nonstationary AMICA decomposition can be used to estimate more continuous state transitions, e.g. changes in drowsiness level defined by changes in behavior in a continuous performance task.

Model probability shifts accompanying changes in behavioral alertness level

Figure 7.8 plots model probability time courses for a 3-model AMICA decomposition of data from one subject, with the subject's reaction speed in response to driving challenges. The probability of model M1 correlated positively with reaction speed ($r = 0.594$), implying that this model was dominant during (more alert) periods when the subject responded quickly to driving challenges. In contrast, the probability of model M2 was strongly negatively correlated with reaction speed ($r = -0.825$), rising when subject reaction speed was low (less alert or drowsy periods). Surprisingly, model M3 was active at the beginning of the experiment and during quick transitions from slower to faster responding (arrows in Figure 7.8). These single-subject results provide evidence that model probabilities learned by three-model AMICA may co-vary with changes in reaction speed (often used, in long experiment sessions, as an index of behavioral alertness), and that the three models each accounted for EEG activity under a different set of performance conditions. Below, we will call models whose model probabilities have the most positive and negative correlations to reaction speed as “fast-response models” and “slow-response models” respectively. The remaining models may be dubbed “intermediate-response models”.

Relationships of model probabilities to performance changes

In Figure 7.9, we report subject mean correlations between model probabilities and reaction speed to study inter-subject variability and compare results against a multi-model ICA-based approach [Hsu and Jung, 2017] in which fast- and slow-response models were learned

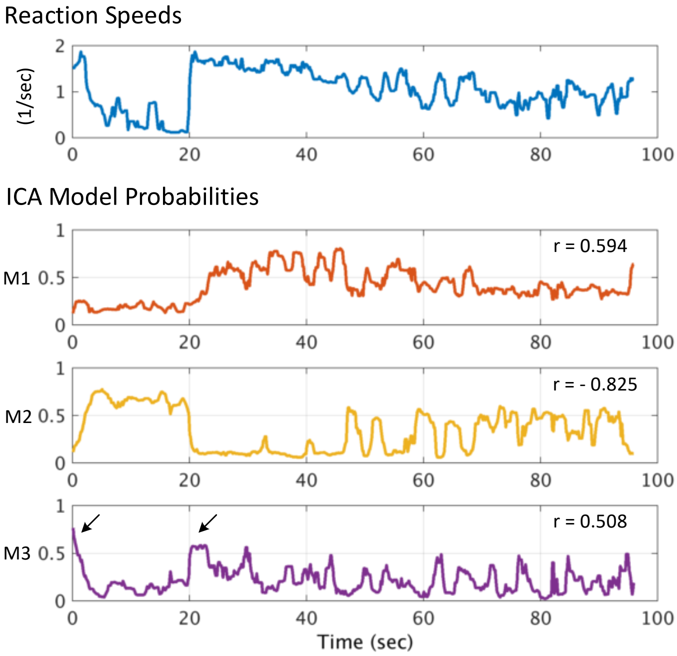


Figure 7.8: The top panel shows reaction speed changes (inverse of reaction times) in response to lane-departure challenges in one simulated driving session. The three bottom panels show the 5-second smoothed probabilities of the models learned by a 3-model AMICA decomposition of the whole EEG data session before lane-departure events. Correlation coefficients (*r*) between each model probability time course and reaction speed are indicated. Black arrows in the lower panel mark brief (alert) periods when model M3 was dominate and reaction speed high.

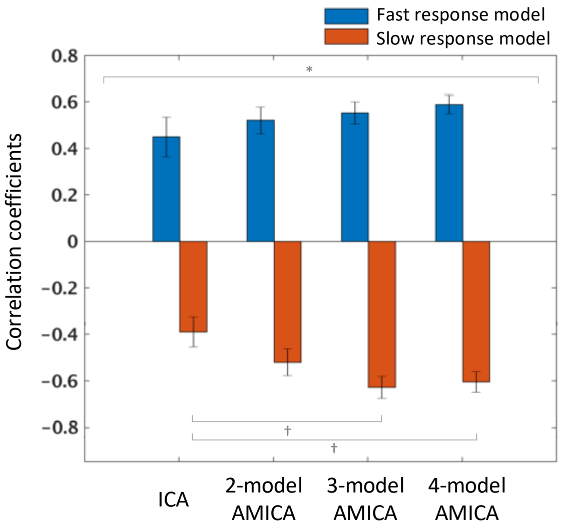


Figure 7.9: Across-subject mean correlation coefficients between reaction speed and model probabilities for fast-response versus slow-response models learned by unsupervised 2-to-4 model AMICA and by separate (supervised) decompositions of fast-response and slow-response periods using separate single-model ICA [Hsu and Jung, 2017]. Standard errors of the mean (I-bars) and results of two-way ANOVA (* $p < 0.05$) and post-hoc multiple comparisons with paired t-test († $p < 0.10$) are shown.

from 90-sec EEG data segments where reaction speeds were fastest and slowest, respectively. For all subjects, AMICA decompositions with two to four models always included at least one fast-response model and one slow-response model, i.e., models whose model probability correlations to reaction speed were significantly positive and negative, respectively. This is a striking result: AMICA, an unsupervised learning approach, automatically and consistently identified two linearly unmixed source models of EEG data acquired when subjects were producing faster and slower responses, respectively.

We used a two-way repeated measures ANOVA on the correlation coefficients reported in Figure 7.9 with the factorial design of two (model types: fast- and slow-response) by four (decomposition methods: ICA and 2-model to 4-model AMICA). The ANOVA with bootstrap significance testing showed a significant interaction ($p < 0.05$) between the model types and the decomposition methods. To identify the source of the significant interaction, we performed post-hoc multiple comparisons by paired t-test between the multi-model ICA and other multi-model AMICA for fast- and slow-response models ($3 \times 2 = 6$ comparisons) with false discovery rate correction (FDR; Benjamini and Yekutieli [2001]). The result revealed weak tendency to significance at $p < 0.10$ level between the ICA and 3- and 4-model AMICA for slow-response models (Figure 7.9).

Rapid model switching dynamics during driving challenges

Changes in model probabilities can also characterize moment-by-moment state changes within single trials. The Figure 7.10 plots, for each latency across trials sorted by driver reaction speed, the index of the highest probability AMICA model time locked to the driving challenge onset, the driver's response onset or response offset. The results for the same subject as in Figure 7.8) are shown above the results for all the trials from the ten drivers to demonstrate that the results generalize across subjects and across (vertical) smoothing of smaller (top) or larger (bottom) numbers of trials.

Figure 7.10A shows that in trials with faster responses, before driving challenge onset, the (blue) fast-response model best fit the data, while before driver challenges in slow-response trials, the (red) slow-response model best fit the data. Switching between the two models occurs as driver response onsets increase from 0.9 sec to 1.1 sec (single subject, top) and from 1 sec to 1.2 sec for all drivers (bottom).

The dynamic switching between best-fitting AMICA models documented in Figure 7.10 thus measure brain dynamic changes preceding behavior on a near-millisecond time scale. Plotting the same trials time locked to driver response onsets (Figure 7.10B) shows that from 0.9 sec to 1.2 sec before response onset (white vertical trace), and again in the 1 sec following response onset, the third, (green) "intermediate" model became dominant briefly, possibly indicating brief hypnagogic ("dreamy") periods moving into and again out of relative alertness. Note that circa 0.5 sec spent by drivers in the relative (blue) alert state preceding response onsets in

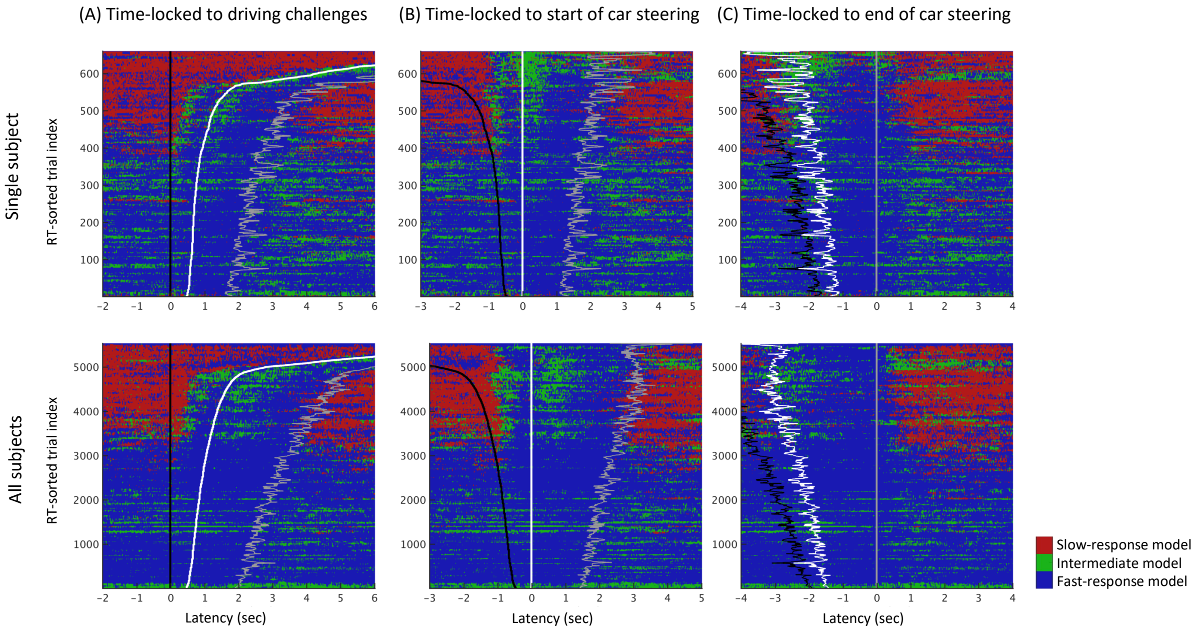


Figure 7.10: Event-related changes in the dominant AMICA model in 3-model AMICA decompositions within data trial epochs (horizontal colored lines) sorted by driver reaction speed. Model probabilities were computed in non-overlapping 20-msec windows. The same trials in the same top-to-bottom order shown are time locked either to (A) driving challenge onsets (black traces), (B) subsequent driver response onsets (white traces), or (C) driver response offsets (gray traces). Top panels show results for 600+ epochs for one subject (same as in Figure 7.8). AMICA models associated with fast, slow, and intermediate response speeds, respectively, were found among each subject's AMICA models. Bottom panels merge model cluster results for all 5000+ available epochs from all 10 subjects. Results shown are smoothed across trials (vertically) using a (single subject) 3-trial or (all subjects) 50-trial sliding window. Note the dominance of the (red) "slow-response" AMICA model (top panels) or model cluster (lower panels) results preceding and following driving challenge onsets in trials in which drivers responded relatively slowly. Notice also the transient dominance of the (green) "intermediate" models following driving-challenge and driver-response onsets in slower-response trials.

slow-response (upper) trials is close to the minimum time required by the drivers to respond to driving challenges in fastest-response (lower) trials. All these details are consistent with the driver challenge (lane deviation) and driver response (car-steering action) constituting a (briefly) arousing event sequence. Figure 7.10C shows that in (upper) slower-response trials the slow-response model learned by AMICA dominated for less than 0.5 sec after the offset of the car-steering action, suggesting that the drivers then relapsed into a more drowsy state, e.g. as soon as attention could safely be withdrawn from the task for some seconds.

Clustering ICs within AMICA models

So far we have demonstrated that shifting AMICA model probabilities can accompany changes in EEG dynamics supporting different cognitive and brain states. Another substantial advantage of the AMICA approach is that it learns generative models, i.e., sets of independent components and their activities and probability density functions (pdfs), that can be related to neurophysiological locations and functions, thereby enabling biologically plausible interpretations.

Figure 7.11 shows IC clustering results for fast-response and slow-response models across the ten subjects (clustering details are described in Section 7.3.3). Both model class clustering solutions included ocular, frontal, central, parietal, and occipital clusters. Slow-response models included more dipolar sources (i.e., with a small residual variance of dipole fitting, see Section 7.3.3 for details) and source clusters (108 ICs, 15 clusters) compared to fast-response models (72 ICs, 12 clusters). This difference appears most notable in right lateral clusters. found only among ICs in the slow-response models. By contrast, the slow-response model left central, parietal, and occipital clusters included 25 ICs, while the corresponding clusters for fast-response models included only 15 ICs.

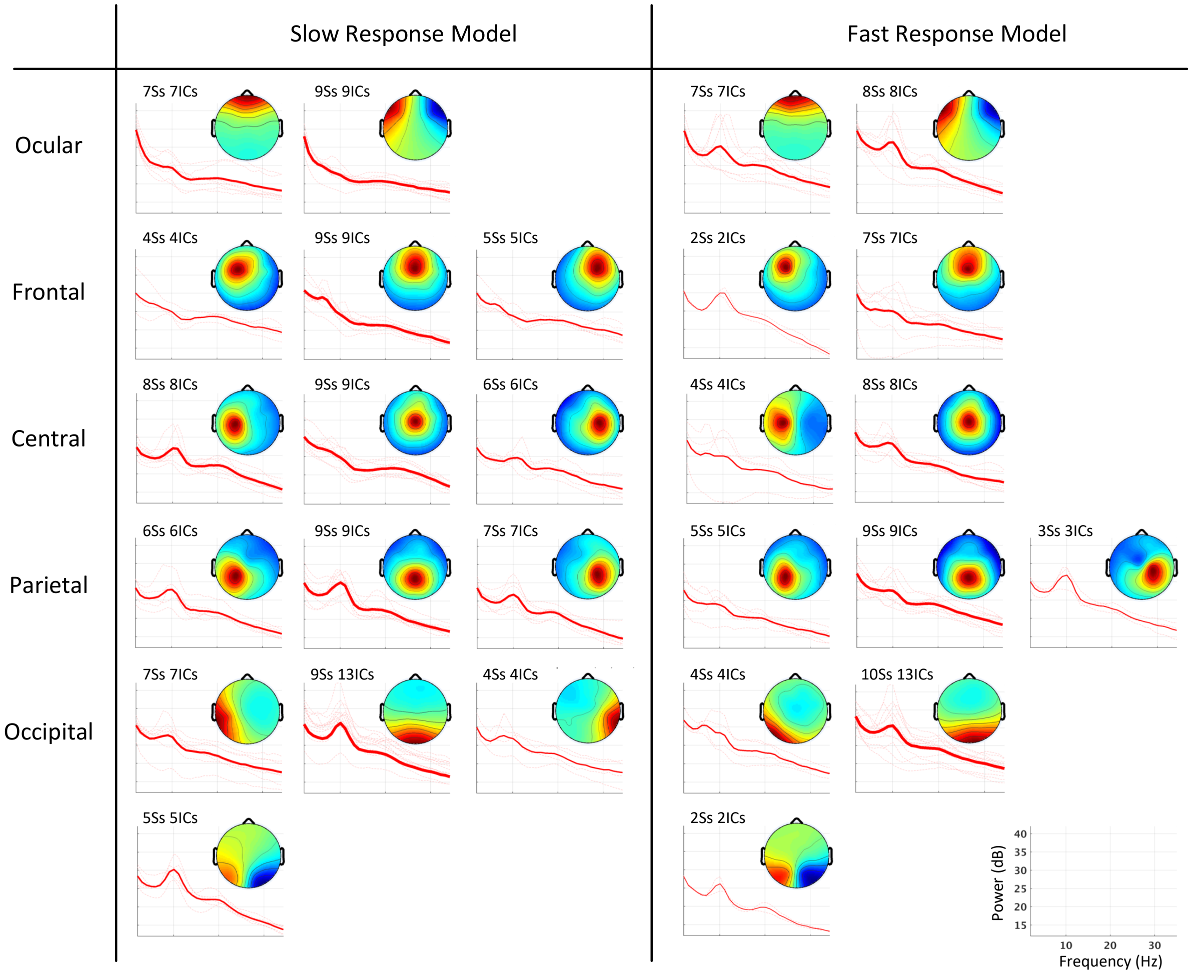


Figure 7.11: Average scalp maps and power spectra of independent component (IC) clusters in slow-response models versus those from clusters in fast-response models from separate 3-model AMICA decompositions of data for each subject. The power spectrum of each IC (thin line) was calculated over 5-second EEG data segments occurring prior to driving challenges in which the respective model had the highest probability. The number of subjects and ICs contributing to each cluster are specified and are also indicated by the width of the power spectral traces.

7.5 Discussion

7.5.1 Unsupervised learning of brain dynamics by modeling source nonstationarity

This study aims to demonstrate the utility of AMICA as a general, unsupervised approach to assessing nonstationary dynamics of cortical dynamic states from nonstationary multichannel EEG signals. Our underlying hypothesis is that the ever-changing formation and dissolution of locally synchronous (or near-synchronous) cortical effective source activities and the network interactions they reflect and support give rise to the fluidity of cognition and behavior. Our results show that these nonstationary dynamics in cortical and cognitive state may be effectively modeled using an ICA mixture model and, specifically, by multi-model AMICA decomposition. Here we applied multi-model AMICA decomposition to one simulated and two actual EEG data sets to evaluate the efficacy of AMICA to estimate abrupt and continuous state changes, to classify multiple sleep stages, and to reveal moment-to-moment cortical (and likely cognitive state) dynamics supporting performance in a simulated driving task. In so doing, we tested the capability of AMICA to estimate continuous state changes, to return consistent model sets across subjects, and to return models suitable for biological interpretation. We also tested the effects of the number of ICA models used. The following subsections discuss these topics in more detail.

7.5.2 Classification of multiple brain states

Our results demonstrate the capability of AMICA decomposition, applied to low channel-count (6- to 13-channel) sleep data, to separate six recognized sleep stages with high classification accuracy based only on changes in the likelihoods of the models AMICA learned from the data. Although the relationship between sleep stages and dominant ICA models was not a one-to-one mapping, the ICA models each captured different source dynamics that jointly characterized differences in EEG activities during the six sleep stages. Hence, in the feature space of model probabilities shown in Figure 7.6, EEG activities from different sleep stages could be clearly separated. Applying a simple Gaussian Bayes classifier to quantitatively assess state separability, we found that based on multi-model AMICA decomposition and using only four to eight data features, we could achieve an average cross-validation accuracy of 75%, significantly higher than chance (17% for a general six-class problem, 38% taking into account the unbalanced numbers of class labels). This sensitivity was higher for REM and N4 stages and lower for stage N1, in alignment with clinical expectations.

Furthermore, classification errors occurred more frequently near sleep stage transitions, and particularly between more strongly related stages (Figure 7.7AB). This may in part reflect the relatively coarse grain (30-sec) of the manual sleep staging, and possible lower inter-scorer consistency in distinguishing strongly related stages. Figure 7.4 shows that during stage changes in model probabilities and thus in EEG activities were not discreet or regular but were continuous and irregular. In particular, in REM or between progressive stages N1 to N4, changes in model probabilities were distributed continuously (Figure 7.6). Thus, the AMICA results suggest that transitions between sleep stages were more continuous, across both time and AMICA "feature space", than as measured by standard sleep stage scoring.

7.5.3 Estimation of rapid state changes

While AMICA assumes and learns discrete ICA models, the relative probabilities of each model measure the "fitness" or likelihood of each model at each data point or group of neighboring data points, that can be effective estimators of moment-to-moment cognitive and behavioral state changes. Applied to the drowsy driving dataset, AMICA automatically and consistently learned fast-response and slow-response trial models for each subject whose model probability changes across time were positively and negatively correlated, respectively, with drowsiness level as indexed by driver speed in reacting behaviorally to occasional lane-deviation driving challenges. These strong and opposite correlations signified that higher likelihood for the fast-response model predicted higher reaction speed, while higher likelihood for the slow-response model predicted lower reaction speed in response to an immediately upcoming driving challenge. Further, rapid (sub-second scale) patterns of shifts between most probable models were consistent with interpretations that appearance of driving challenges induced brief changes from less alert to more alert EEG dynamics, and that during less alert (slow-response model) periods, EEG dynamics typically shifted back to less alert model a second or less after the offset of the driver's behavioral response. Further, these brief transitions between less alert and more alert state often involved momentary transitions through a third ("intermediate") AMICA model. The models returned by AMICA decompositions exhibited these close relationships to the behavioral data record despite not using any direct information about the nature or timing of experimental events and behavioral responses.

Our previous studies have employed other measures to quantify EEG state changes during simulated driving and sleep, including a nonstationary index [Hsu and Jung, 2017] and relative likelihoods [McKeown et al., 1998] of separately-trained ICA models. Compared with these studies, AMICA here learned multiple ICA models that proved able to better characterize the EEG dynamics and could be generalized to follow both irregular and transient shifts between more than two brain states. Instead of training multiple ICA models on separate sets of data segregated by behavior [Hsu and Jung, 2017], AMICA, an unsupervised learning approach, here automatically learned distinctions between EEG activities occurring in different brain/behavioral states. More importantly, as shown in Figure 7.9, unsupervised multi-model AMICA had comparable performance with the supervised ICA approach in estimating drowsiness levels, even showing weak tendency ($p < 0.10$) of improved performance when 3- or 4-model AMICA was used. This weak tendency might become significant when more subjects are included in the analysis.

By examining switching between dominant models within single trials with sub-second temporal resolution, we found a consistent sequence and timing of brain state changes immediately before and after driver responses to experimental driving challenges. When drivers were drowsy, i.e., exhibiting EEG best fit by their "slow-response" model, they were slow in detecting lane-departure events. In many trials, drivers began their behavioral response to these challenges within about a second (0.9 to 1.2 secs) after their EEG exhibited a very brief transition to "intermediate" model dynamics, their motor response appearing about half a second after their faster-response model then became dominant. Following the end of these motor responses, drivers relapsed into the slow-response model dynamics after only about a second. These results demonstrate capability of multi-model AMICA decomposition to track cortical dynamic state changes on the sub-second time scale.

7.5.4 Consistency across subjects

Although AMICA, as an unsupervised learning approach, need not give learned ICA models that are similar across subjects, applied to actual experiment data AMICA here produced results that were surprisingly consistent across subjects in three senses:

(1) Consistent relations between ICA models and brain states were clearly observed in both applications (sleep and driving challenges). In the sleep dataset, Figure 7.5 shows that ICA models with similar probability distributions over sleep stages were found across all subjects. In other words, for each subject some ICA models were dominant during specific sleep stages (e.g. group B model during stage N4). Similarly, Figure 7.9 shows that slow-response and fast-response models were consistently learned for all subjects in the drowsy driving experiment.

(2) Results included consistent differences in AMICA model probabilities across subjects. As shown in Figure 7.6, although the model probabilities were here based on subject-specific ICA models, their values could be directly summarized across all subjects without normalization. These results provide strong evidence that model probabilities, intrinsically bounded from zero to one, can be global indices that generalize across subjects.

(3) Differences between independent component (IC) clusters (here based on IC scalp maps) for different model classes appeared and are discussed in the next subsection.

7.5.5 Biological interpretation of AMICA models

Besides its unsupervised segmentation of nonstationary data into putative brain dynamic and function states, another benefit of the AMICA approach is that it learns a generative model that characterizes a complete set of active, statistically maximally independent components (ICs) in each state, plus a set of time series giving the probability of each model at each time point based on a probability density function (PDF) learned for each model IC from the data. During iterative training, each model becomes adapted to time points at which it is most probable. We validated this characterization by applying multi-model AMICA decomposition to simulated quasi-stationary data, showing that multi-model AMICA can accurately learn the ground-truth source IC scalp projection patterns, activities, and PDFs. A growing amount of evidence suggests an association between many ICs and localized biological and functional processes in cortex [Makeig et al., 2002, Onton et al., 2006]. By constructing an individualized subject electrical forward model from an MR head image, a subset of (brain source) ICs can be further localized using either single or dual equivalent current dipole or distributed cortical patch models [Acar and Makeig, 2010, Gwin and Ferris, 2012, Acar et al., 2016].

Applied to the drowsy driving dataset, IC processes learned by AMICA were generally consistent across subjects. ICs compatible with a compact cortical source area or eye movement artifact could be clustered into similar fast-response and slow-response model source clusters based on scalp map correlations. The identified IC clusters, including clusters mainly projecting to the frontal regions with high theta or alpha power and the occipital and parietal clusters with high alpha power, were consistent with previous studies applying a single-model ICA decomposition to these data [Chuang et al., 2014, Hsu and Jung, 2017]. There, differences in the dynamics of similar ICs were shown to be associated with alert and drowsy states respectively. Interestingly, more dipolar sources (see Section 7.3.3 for details) were found by AMICA in the slow-response models than the fast-response models. This result may be related to the fact that the brain activities, especially alpha waves, spread through larger cortical areas during drowsiness as reported in Santamaria and Chiappa [1987] and Lal and Craig [2002]. In our results, stronger alpha activities appear in frontal and prefrontal (indicated as ocular) fast-response cluster ICs. These clusters may be driven by anterior cingulate activity [Jones and Harrison, 2001]. AMICA also identified a larger number of dipolar ICs for both fast- and slow-response models than the ICA-based approach, Hsu and Jung [2017] (Fig. 3), suggesting that unsupervised multi-model AMICA might be a more effective approach to learning state-related ICA models than their supervised multi-model ICA approach in which the models were trained on manually selected data segments

We could not study the cortical origins of the ICs learned from these sleep data as the CAP sleep database consists of only low-density sleep EEG data recorded using bipolar channels. The application of AMICA decomposition to high-density sleep EEG data could be of interest to sleep research exploring changes in effective EEG sources and source network activities in each sleep stage.

7.5.6 Choosing the model order

One of the most important parameters required to apply AMICA is the number of ICA models, i.e., H in Equation 7.1. Since the ground-truth model order of the data is typically unknown, the present work focuses on examining the effects of assumed model order on AMICA performance. Applied to simulated 3-segment quasi-stationary data, complete 3-model AMICA decomposition and over-complete (4- to 6-model) AMICA decomposition all successfully segmented the data and accurately learned the ground-truth sources, suggesting that in many applications choice of model order might not crucially affect the validity of AMICA results in particular when a complete (ground-truth) number of models, or at most only a few excess models are learned. Typically, excess models only account for a small portion of data not well modeled by the other ICA models, e.g. data points at which many sources are unusually co-activated (in the presence of adventitious artifact, for example). When applied to the simulated driving data, AMICA decomposition using two, three, or four models consistently returned “fast-response” and “slow-response” models accounting for the EEG data in alert and drowsy behavioral conditions, respectively. A third (“intermediate”) model (M3 in Figure 7.8) accounted for EEG activities not well fit by the two dominant models, e.g. during brief transitions between the two dominant EEG states.

The above results provide evidence that choosing a precise number of models is not critical to the information value of AMICA decomposition (including model probabilities and brain source characteristics). For example, applying 2-model through 10-model AMICA decomposition to the sleep data from a single subject, we found that that adding or eliminating one model typically returned models with almost identical model probability dynamics. As with other clustering analyses, increasing the model order may produce a new model accounting for lower-probability data points of one or two existing “parent” models while leaving other existing models intact.

Several approaches have been proposed to help select the number of nonstationary data models. For example, one may compare the marginal likelihood for different candidate models by adding a penalty on model complexity, for example the Akaike information criterion (AIC) [Akaike, 1974] or Bayesian information criterion (BIC) [Schwarz et al., 1978]. Some adaptive approaches including variational Bayesian learning [Chan et al., 2002] and online adaptive learning [Lin et al., 2005] have also been proposed. However, these methods are computationally expensive and also require heuristic setting of thresholds for splitting or merging source clusters.

7.5.7 Alternative approaches

Although this study focuses on AMICA decomposition, the results might be able to generalize to other ICAMM approaches that may have other desirable properties. For example, different approximations of source probability density functions (PDFs) can be used to better match the underlying source activity in the data, such as a generalized exponential model [Roberts and Penny, 2001], a mixture of Gaussians [Chan et al., 2002], and a nonparametric model [Salazar et al., 2010b].

Hidden Markov Models (HMM) form another family of generative models with a rigorous temporal structure for unsupervised brain state monitoring. Previous studies, often applied to source-space MEG signals, have demonstrated that the HMM-based approaches could characterize transient brain states in rest and task [Baker et al., 2014, Vidaurre et al., 2016, 2017, Nielsen et al., 2017]. The generative assumptions between HMMs and AMICA differ, as HMMs generally use a variation of Gaussian distributions parametrized by the states while AMICA assumes an ICA mixture model. All such HMM methods seem to be applied in source-space to explicitly model functional connectivity between sources; AMICA instead operates in sensor-space and learns collections of sources which are likely to be active simultaneously during some time periods in the data. Even so, HMM and ICA are not mutually exclusive as evidenced in the proposal for Hidden Markov ICA [Penny et al., 2000] and sequential ICAMM [Salazar et al., 2010a] where HMMs govern transitions in multi-model ICA decompositions. AMICA might be generalized in a similar way and may help in situations when state transitions are likely structured and continuous over time, such as during sleep.

7.5.8 Limitations and open questions

Given that multi-model AMICA must learn parameters at each of its three layers (Figure 7.1), the issue of identifiability – whether varying sets of model parameters across the three layers may equally well account for the decomposed data – is legitimate.

Like most unsupervised-learning and data-driven approaches, successful AMICA decomposition has data and computation requirements. Source-level analyses such as ICA require relatively high-density EEG data to achieve meaningful source separation. They also implicitly assume that the number of data channels is at least as large as the number of substantial effective sources. AMICA relaxes these assumptions by learning multiple ICA models and allowing source dependence between the different models. How much this relaxation of the ICA assumptions can improve AMICA's performance in applications to low-density EEG data and in identifying and interpreting dependent sources is still unclear and worth studying. For example, applied to the sleep dataset, AMICA achieved an average accuracy of 75% in six-class classification using EEG data with only 6–13 channels, but this accuracy dropped to 68% for subjects with only five EEG channels available.

Another requirement for successful AMICA decomposition is a reasonable number of data samples. Learning H ICA models, each with N stable sources, requires approximately $k \cdot H \cdot N^2$ samples, where empirically $k \geq$ 25 [Onton and Makeig, 2006]. For $H = 6$, $N = 16$, $k = 25$, and a 250-Hz sampling rate, this corresponds to $\sim$2.5-min of data; hence here we generated 3-minute stationary segments in the simulated data. Lastly, AMICA decomposition requires significant computation time to run on a personal computer. For 13-channel sleep EEG data from 9-hour recordings with a 512-Hz sampling rate, multi-model AMICA decomposition required 13–15 hours on a 2.40-GHz CPU. However, AMICA computation time can be significantly reduced through parallelization, as featured in the AMICA code made available (https://sccn.ucsd.edu/~jason/amica_web.html) by its author, Jason Palmer, and interested users might explore use of the Neuroscience Gateway (www.nsgportal.org) to run AMICA decompositions on larger data sets.

Results of this study support the use of multi-model AMICA decomposition for assessing brain state changes by validating its performance on sleep stage classification and alert versus drowsy performance estimation. These results provide evidence to support the application of multi-model AMICA decomposition as a general unsupervised-learning approach to study the continuous, endogenous, and nonstationary brain dynamics in either EEG, MEG [Iversen and Makeig, 2014], or electroencephalographic (ECoG) data [Whitmer et al., 2012]. For example, AMICA decomposition might be applied to multichannel brain electrical signals to explore brain dynamics during rest, movie watching, or hypnotherapy, to identify the nonstationary, task-irrelevant brain source activity changes during performance of a complex cognitive task, or even to study mental strategy or emotional shifts using a brain-computer interface.

7.6 Conclusions

Here we have demonstrated that AMICA decomposition provides a general unsupervised approach to mining changes in effective source dynamics in nonstationary multichannel EEG signals. The underlying hypothesis here is that different brain states may involve different active effective sources (each typically compatible with an emergent area of locally-synchronous cortical field activity), and that the locations and source-level probability density functions (PDFs) of these state-specific effective source activities can be well modeled by transitions between ICA data models.

We showed that, applied to simulated quasi-stationary data, AMICA decomposition could accurately learn the ground truth sources and source activities, either when directed to return complete or (mildly) over-complete model sets. Applied to some sleep EEG data, multi-model AMICA decompositions could be used to meaningfully characterize sleep dynamics, giving consistent results across subjects and allowing 75% cross-validation accuracy in classifying data from six sleep stages validated by expert sleep scoring.

Applied to EEG datasets recorded during simulated driving, AMICA automatically identified two models accounting for EEG activity in slow- and fast-response trials respectively. The corresponding model probability differences could be used as an effective estimator of reaction speed in single trials and appeared to track brain dynamic state changes on the sub-second scale. In addition, AMICA decomposition also learned physiologically interpretable results including the spatial distribution and temporal activity pattern of the effective brain sources in each ICA model.

Thus multi-model AMICA decomposition can be applied to continuous and unlabeled EEG (or other electrophysiological) data to study, for example, nonstationarities in brain dynamics during resting states, accompanying mental strategy changes, or through different states of emotion, fatigue, and arousal.

7.7 Acknowledgements

Chapter 7, in part, is a reprint of the material as it appears in Sheng-Hsiou Hsu, Luca Pion-Tonachini, Jason Palmer, Makoto Miyakoshi, Scott Makeig, and Tzyy-Ping Jung. Modeling brain dynamic state changes with adaptive mixture independent component analysis. *NeuroImage*, 183:47–61, 2018. The dissertation author was the secondary investigator of this paper. This work was supported in part by a gift fund (KreutzKamp TMS RES F-2467), the National Science Foundation under grant number IIP-1719130), and the Army Research Laboratory under grant number W911NF-10-2-0022. Dr. Makeig's participation was funded by a grant from the National Institutes of Health under grant number R01 NS047293-13A1 and by a gift from The Swartz Foundation (Old Field, NY).

Chapter 8

Conclusion

As electroencephalogram (EEG) source analyses, and EEG analyses in general, progresses towards automated pipelines with minimal manual intervention, the potential impacts of EEG-based methods grow with them. Not only can the scale at which they are applied expand without the need for individual expert involvement and intervention, but even methods for applications that are less important could spread if the “barrier to entry” for using EEG is mitigated sufficiently.

To this end, we have identified the stages of EEG source analysis which have continued to necessitate manual intervention and either developed tools to automate them or extensively quantified the capacity of promising methods which were previously largely uncharacterized. We developed the ICLabel classifier for automatic EEG component classification in Chapter 3. In Chapter 4 we quantified the efficacy of artifact removal using artifact subspace reconstruction (ASR) and its effects on subsequent independent component analysis (ICA) decompositions of the processed EEG data. In Chapter 7 we explored the utility of adaptive mixture ICA (AMICA) for automatic segmentation of EEG time series and nonstationarity analysis.

As most automatic methods are often, at least initially, met with skepticism in the scientific community, we do not expect this work to immediately transform the way EEG data are analyzed. A key element to automation is trust, without which automation becomes mere suggestion. We have measured and evaluated the effectiveness of both novel and existing components that allow for automated EEG source analysis in the hope that these analyses will begin to cultivate the trust necessary for widespread adoption of automated EEG processing pipelines. To further this goal, in Chapters 5 and 6 we have combined many of these methods, along with others, in the real-time EEG source-mapping toolbox (REST) to *explicitly* show what is happening at each stage of the EEG processing pipeline. Furthermore, the novel tools we developed, such as the ICLabel classifier and REST, have publicly-available implementations to ease their adoption. We also do not claim that any of these methods are the best possible means to accomplish their respective goals. They have been developed, analyzed, and presented here with the full expectation, hope even, that they will be superseded by better methods in time. It is for this reason that we made the ICLabel dataset easily available for download, along with many other pieces of the work covered in previous chapters.

With all of these advancements, EEG practitioners can more easily develop and apply new EEG methods and, possibly, utilize them in real-time and mobile applications.

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/. Software available from tensorflow.org.

Zeynep Akalin Acar and Scott Makeig. Neuroelectromagnetic forward head modeling toolbox. *Journal of Neuroscience Methods*, 190(2):258–270, 2010.

Zeynep Akalin Acar, Can E. Acar, and Scott Makeig. Simultaneous head tissue conductivity and EEG source location estimation. *NeuroImage*, 124:168–180, 2016.

Geoffray Adde, Maureen Clerc, Olivier Faugeras, Renaud Keriven, Jan Kybic, and Théodore Papadopoulo. Symmetric BEM formulation for the M/EEG forward problem. In Chris Taylor and J. Alison Noble, editors, *Information Processing in Medical Imaging*, pages 524–535, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

Deepak Agarwal and Bee-Chung Chen. fLDA: matrix factorization through latent Dirichlet allocation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pages 91–100. ACM, 2010.

Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

Muhammad Tahir Akhtar, Tzyy-Ping Jung, Scott Makeig, and Gert Cauwenberghs. Recursive independent component analysis for online blind source separation. In *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, pages 2813–2816. IEEE, 2012.

Shun-ichi Amari, Noboru Murata, Klaus-Robert Müller, Michael Finke, and Howard Hua Yang. Asymptotic statistical theory of overtraining and cross-validation. *IEEE Transactions on Neural Networks*, 8(5):985–996, Sept 1997. ISSN 1045-9227. doi: 10.1109/72.623200.

Fiorenzo Artoni, Chiara Fanciullacci, Federica Bertolucci, Alessandro Panarese, Scott Makeig, Silverstro Micera, and Carmelo Chisari. Unidirectional brain to muscle connectivity reveals motor cortex control of leg muscles during stereotyped walking. *NeuroImage*, 159:403–416, 2017.

Adam P. Baker, Matthew J. Brookes, Iead A. Rezek, Stephen M. Smith, Timothy Behrens, Penny J. Probert Smith, and Mark Woolrich. Fast transient networks in spontaneous human brain activity. *eLife*, 3, 2014.

Claudia Beleites, Reiner Salzer, and Valter Sergo. Validation of soft classification models using partial class memberships: An extended concept of sensitivity & co. applied to grading of astrocytoma tissues. *Chemometrics and Intelligent Laboratory Systems*, 122:12–22, 2013. ISSN 0169-7439. doi: https://doi.org/10.1016/j.chemolab.2012.12.003. URL http://www.sciencedirect.com/science/article/pii/S0169743912002419.

Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995. doi: 10.1162/neco.1995.7.6.1129. URL https://doi.org/10.1162/neco.1995.7.6.1129.

Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, pages 1165–1188, 2001.

Richard F. Betzel, Molly A. Erickson, Malene Abell, Brian F. O'Donnell, William P. Hetrick, and Olaf Sporns. Synchronization dynamics and evidence for a repertoire of network states in resting EEG. *Frontiers in Computational Neuroscience*, 6, 2012.

Nima Bigdely-Shamlo, Ken Kreutz-Delgado, Christian Kothe, and Scott Makeig. EyeCatch: Data-mining over half a million EEG independent components to construct a fully-automated eye-component detector. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 5845–5848. IEEE, 2013.

Nima Bigdely-Shamlo, Tim Mullen, Christian Kothe, Kyung-Min Su, and Kay A. Robbins. The PREP pipeline: standardized preprocessing for large-scale EEG analysis. *Frontiers in Neuroinformatics*, 9, 2015.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

Mary A.B. Brazier. A study of the electrical fields at the surface of the head. *American Journal of EEG Technology*, 6(4):114–128, 1966. doi: 10.1080/00029238.1966.11080676. URL https://doi.org/10.1080/00029238.1966.11080676.

Chris Buckley, Matthew Lease, and Mark D. Smucker. Overview of the TREC 2010 Relevance Feedback Track (Notebook). In *The Nineteenth Text Retrieval Conference (TREC) Notebook*, 2010. URL ../papers/trec-notebook-2010.pdf.

Kevin Robert Canini, Lei Shi, and Thomas L. Griffiths. Online inference of topics with latent Dirichlet allocation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 65–72, 2009.

Kwokleung Chan, Te-Won Lee, and Terrence J. Sejnowski. Variational learning of clusters of undercomplete nonsymmetric independent components. *Journal of Machine Learning Research*, 3(Aug):99–114, 2002. ISSN ISSN 1533-7928.

Chi-Yuan Chang, Sheng-Hsiou Hsu, Luca Pion-Tonachini, and Tzyy-Ping Jung. Evaluation of artifact subspace reconstruction for automatic EEG artifact removal. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1242–1245. IEEE, 2018.

Chi-Yuan Chang, Sheng-Hsiou Hsu, Luca Pion-Tonachini, and Tzyy-Ping Jung. Evaluation of artifact subspace reconstruction for automatic EEG artifact component removal. *Transactions on Biomedical Engineering (under review)*, 2019.

Maximilien Chaumon, Dorothy V.M. Bishop, and Niko A. Busch. A practical guide to the selection of independent components of the electroencephalogram for artifact correction. *Journal of Neuroscience Methods*, 250:47–63, 2015.

Catherine J. Chu, Mark A. Kramer, Jay Pathmanathan, Matt T. Bianchi, M. Brandon Westover, Lauren Wizon, and Sydney S. Cash. Emergence of stable functional networks in long-term human electroencephalography. *Journal of Neuroscience*, 32(8):2703–2713, 2012.

Chun-Hsiang Chuang, Li-Wei Ko, Yuan-Pin Lin, Tzyy-Ping Jung, and Chin-Teng Lin. Independent component ensemble of EEG for brain–computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(2):230–238, 2014.

Zihang Dai, Zhilin Yang, Fan Yang, William W. Cohen, and Ruslan R. Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*, pages 6510–6520, 2017.

Alexander Philip Dawid and Allan M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28:20–28, 1979.

Alain de Cheveigné. Sparse time artifact removal. *Journal of Neuroscience Methods*, 262:14–20, 2016.

Nicolás Della Penna and Mark D. Reid. Crowd & prejudice: An impossibility theorem for crowd labelling without a gold standard. *arXiv preprint arXiv:1204.3511*, 2012.

Arnaud Delorme and Scott Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1):9–21, 2004.

Arnaud Delorme, Tim Mullen, Christian Kothe, Zeynep Akalin Acar, Nima Bigdely-Shamlo, Andrey Vankov, and Scott Makeig. EEGLAB, SIFT, NFT, BCILAB, and ERICA: new tools for advanced EEG processing. *Computational Intelligence and Neuroscience*, 2011:10, 2011.

Arnaud Delorme, Jason Palmer, Julie Onton, Robert Oostenveld, and Scott Makeig. Independent EEG sources are dipolar. *PloS One*, 7(2):e30135, 2012.

Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st International Conference on World Wide Web*, pages 469–478. ACM, 2012.

Joseph Dien. Issues in the application of the average reference: Review, critiques, and recommendations. *Behavior Research Methods, Instruments, & Computers*, 30(1):34–43, 1998.

Georg E. Fabiani, Dennis J. McFarland, Jonathan R. Wolpaw, and Gert Pfurtscheller. Conversion of EEG activity into cursor movement by a brain-computer interface (BCI). *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12(3):331–338, 2004.

Laura Frølich, Tobias S. Andersen, and Morten Mørup. Classification of independent components of EEG into multiple artifact classes. *Psychophysiology*, 52(1):32–45, 2015. doi: 10.1111/psyp.12290. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/psyp.12290.

Laurel J. Gabard-Durnam, Adriana S. Mendez Leal, Carol L. Wilkinson, and April R. Levin. The harvard automated processing pipeline for electroencephalography (HAPPE): standardized processing software for developmental and high-artifact data. *Frontiers in Neuroscience*, 12: 97, 2018.

Junfeng Gao, Chongxun Zheng, and Pei Wang. Online removal of muscle artifact from electroencephalogram signals based on canonical correlation analysis. *Clinical EEG and Neuroscience*, 41(1):53–59, 2010.

Alan S. Gevins, Charles L. Yeager, Gerald M. Zeitlin, Sonia Ancoli, and Mark F. Dedon. On-line computer rejection of EEG artifact. *Electroencephalography and Clinical Neurophysiology*, 42 (2):267–274, 1977.

Ary L. Goldberger, Luis A.N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23), 2000.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Alexandre Gramfort, Théodore Papadopoulo, Emmanuel Olivi, Maureen Clerc, et al. OpenMEEG: opensource software for quasistatic bioelectromagnetics. *Biomedical Engineering Online*, 9 (1):45, 2010.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

Tyler S. Grummett, Sean P. Fitzgibbon, Trent W. Lewis, Dylan DeLosAngeles, Emma M. Whitham, Kenneth J. Pope, and John O. Willoughby. Constitutive spectral EEG peaks in the gamma range: suppressed by sleep, reduced by mental activity and resistant to sensory stimulation. *Frontiers in Human Neuroscience*, 8:927, 2014.

Roberto Guarnieri, Marco Marino, Federico Barban, Marco Ganzetti, and Dante Mantini. Online EEG artifact removal for BCI applications by adaptive spatial filtering. *Journal of Neural Engineering*, 15(5):056009, 2018.

Joseph T. Gwin and Daniel P. Ferris. An EEG-based study of discrete isometric and isotonic human lower limb muscle contractions. *Journal of Neuroengineering and Rehabilitation*, 9(1): 35, 2012.

Frank R. Hampel, Elvezio M. Ronchetti, Peter J. Rousseeuw, and Werner A. Stahel. *Robust statistics: the approach based on influence functions*, volume 196. John Wiley & Sons, 2011.

C.J Henderson, Stuart R. Butler, and Alan Glass. The localization of equivalent dipoles of eeg sources by the application of electrical field theory. *Electroencephalography and Clinical Neurophysiology*, 39(2):117–130, 1975. ISSN 0013-4694. doi: https://doi.org/10.1016/0013-4694(75)90002-4. URL http://www.sciencedirect.com/science/article/pii/0013469475900024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Matthew Hoffman, Francis R. Bach, and David M. Blei. Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 856–864, 2010.

Sheng-Hsiou Hsu and Tzyy-Ping Jung. Monitoring alert and drowsy states by modeling EEG source nonstationarity. *Journal of Neural Engineering*, 14(5):056012, 2017.

Sheng-Hsiou Hsu, Tim Mullen, Tzyy-Ping Jung, and Gert Cauwenberghs. Online recursive independent component analysis for real-time source separation of high-density EEG. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 3845–3848. IEEE, 2014.

Sheng-Hsiou Hsu, Luca Pion-Tonachini, Tzyy-Ping Jung, and Gert Cauwenberghs. Tracking non-stationary EEG sources using adaptive online recursive independent component analysis. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 4106–4109. IEEE, 2015.

Sheng-Hsiou Hsu, Tim R. Mullen, Tzyy-Ping Jung, and Gert Cauwenberghs. Real-time adaptive EEG source separation using online recursive independent component analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(3):309–319, 2016.

Sheng-Hsiou Hsu, Luca Pion-Tonachini, Jason Palmer, Makoto Miyakoshi, Scott Makeig, and Tzyy-Ping Jung. Modeling brain dynamic state changes with adaptive mixture independent component analysis. *NeuroImage*, 183:47–61, 2018.

Jing Hu, Chun-sheng Wang, Min Wu, Yu-xiao Du, Yong He, and Jinhua She. Removal of EOG and EMG artifacts from EEG using combination of functional link neural network and adaptive neural fuzzy inference system. *Neurocomputing*, 151:278–287, 2015.

Ruey-Song Huang, Tzyy-Ping Jung, and Scott Makeig. Tonic changes in EEG power spectra during simulated driving. *Foundations of Augmented Cognition: Neuroergonomics and Operational Neuroscience*, pages 394–403, 2009.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 64–67. ACM, 2010.

Md Kafiul Islam, Amir Rastegarnia, and Zhi Yang. Methods for artifact detection and removal from scalp EEG: a review. *Neurophysiologie Clinique/Clinical Neurophysiology*, 46(4-5): 287–305, 2016.

John R. Iversen and Scott Makeig. MEG/EEG data analysis using EEGLAB. In *Magnetoencephalography*, pages 199–212. Springer, 2014.

Mainak Jas, Denis Engemann, Yousra Bekhti, Federico Raimondo, and Alexandre Gramfort. Autoreject: Automated artifact rejection for MEG and EEG data. *NeuroImage*, 159:417–429, 2017.

Kay Jones and Yvonne Harrison. Frontal lobe function, sleep loss and fragmented sleep. *Sleep Medicine Reviews*, 5(6):463–475, 2001.

Carrie A. Joyce, Irina F. Gorodnitsky, and Marta Kutas. Automatic removal of eye movement and blink artifacts from EEG data using blind component separation. *Psychophysiology*, 41(2): 313–325, 2004.

Tzyy-Ping Jung, Colin Humphries, Te-Won Lee, Scott Makeig, Martin J. McKeown, Vicente Iragui, and Terrence J. Sejnowski. Extended ICA removes artifacts from electroencephalographic recordings. In *Advances in neural information processing systems*, pages 894–900, 1998.

Tzyy-Ping Jung, Scott Makeig, Colin Humphries, Te-Won Lee, Martin J. McKeown, Vicente Iragui, and Terrence J. Sejnowski. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37(2):163–178, 2000a.

Tzyy-Ping Jung, Scott Makeig, Te-Won Lee, Martin J. McKeown, Glen Brown, Anthony J. Bell, and Terrence J. Sejnowski. Independent component analysis of biomedical signals. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation*, pages 633–644. Citeseer, 2000b.

Christian Jutten and Jeanny Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

Alexander Kaplan, Joachim Röschke, Boris Darkhovsky, and Juergen Fell. Macrostructural EEG characterization based on nonparametric change point segmentation: application to sleep analysis. *Journal of Neuroscience Methods*, 106(1):81–90, 2001.

Arjun Khanna, Alvaro Pascual-Leone, Christoph M. Michel, and Faranak Farzan. Microstates in resting-state EEG: current status and future directions. *Neuroscience & Biobehavioral Reviews*, 49:105–113, 2015.

Hyun-Chul Kim and Zoubin Ghahramani. Bayesian classifier combination. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 619–627, 2012.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

George H. Klem, Hans Otto Lüders, H.H. Jasper, and C. Elger. The ten-twenty electrode system of the international federation. *Electroencephalographic Clinical Neurophysiology*, 52(3):3–6, 1999.

Christian Kothe. Lab streaming layer (LSL). *https://github.com/sccn/labstreaminglayer*, 2014.

Christian Kothe and Tzyy-Ping Jung. Artifact removal techniques with signal reconstruction. *U.S. Patent Application No. 14/895,440*, 2014.

Christian Kothe and Scott Makeig. BCILAB: a platform for brain-computer interface development. *Journal of Neural Engineering*, 10(5):056014, 2013.

Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent Dirichlet allocation for tag recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*, pages 61–68. ACM, 2009.

Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 03 1951. doi: 10.1214/aoms/1177729694. URL http://dx.doi.org/10.1214/aoms/1177729694.

Saroj K.L. Lal and Ashley Craig. Driver fatigue: electroencephalography and psychological assessment. *Psychophysiology*, 39(3):313–321, 2002.

Clement Lee and Scott Makeig. get_chanlocs: Compute 3-D electrode positions from a 3-D head image. https://sccn.ucsd.edu/wiki/Get_chanlocs, 2018. Accessed: 2019-01-30.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryiAv2xAZ.

Te-Won Lee, Mark Girolami, and Terrence J. Sejnowski. Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural Computation*, 11(2):417–441, 1999.

Te-Won Lee, Michael S. Lewicki, and Terrance J. Sejnowski. ICA mixture models for unsupervised classification of non-gaussian classes and automatic context switching in blind signal separation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10): 1078–1089, 2000. ISSN 01628828. doi: 10.1109/34.879789.

Dietrich Lehmann, Hisaki Ozaki, and Ivan Pál. EEG alpha map series: brain micro-states by space-oriented adaptive segmentation. *Electroencephalography and Clinical Neurophysiology*, 67(3):271–288, 1987.

Marie Lienou, Henri Maitre, and Mihai Datcu. Semantic annotation of satellite images using latent Dirichlet allocation. *IEEE Geoscience and Remote Sensing Letters*, 7(1):28–32, Jan 2010. ISSN 1545-598X. doi: 10.1109/LGRS.2009.2023536.

Chin-Teng Lin, Wen-Chang Cheng, and Sheng-Fu Liang. An on-line ICA-mixture-model-based self-constructing fuzzy neural network. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(1):207–221, 2005.

Chin-Teng Lin, Kuan-Chih Huang, Chih-Feng Chao, Jian-Ann Chen, Tzai-Wen Chiu, Li-Wei Ko, and Tzyy-Ping Jung. Tonic and phasic EEG and behavioral changes induced by arousing feedback. *NeuroImage*, 52(2):633–642, 2010.

Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

Scott Makeig and Mark Inlow. Lapse in alertness: coherence of fluctuations in performance and spectrum. *Electroencephalography and Clinical Neurophysiology*, 86(1):23–35, 1993.

Scott Makeig, Anthony J. Bell, Tzyy-Ping Jung, Terrence J. Sejnowski, et al. Independent component analysis of electroencephalographic data. *Advances in Neural Information Processing Systems*, pages 145–151, 1996.

Scott Makeig, Sigurd Enghoff, Tzyy-Ping Jung, and Terrence J. Sejnowski. A natural basis for efficient brain-actuated control. *Rehabilitation Engineering, IEEE Transactions on*, 8(2): 208–211, 2000.

Scott Makeig, Marissa Westerfield, Tzyy-Ping Jung, Sigurd Enghoff, Jeanne Townsend, Eric Courchesne, and Terrance J. Sejnowski. Dynamic brain sources of visual evoked responses. *Science*, 295(5555):690–694, 2002.

Scott Makeig, Stefan Debener, Julie Onton, and Arnaud Delorme. Mining event-related brain dynamics. *Trends in Cognitive Sciences*, 8(5):204–210, 2004.

Jaakko Malmivuo and Robert Plonsey. *Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields*. Oxford University Press, 1995. ISBN 9780195058239. URL https://books.google.com/books?id=H9CFM0TqWwsC.

Nadia Mammone, Fabio La Foresta, and Francesco Carlo Morabito. Automatic artifact rejection from multichannel scalp EEG by wavelet ICA. *IEEE Sensors Journal*, 12(3):533–542, 2012.

Josep Marco-Pallares, Carles Grau, and Giulio Ruffini. Combined ICA-LORETA analysis of mismatch negativity. *NeuroImage*, 25(2):471–477, 2005.

Peter J. Marshall, Yair Bar-Haim, and Nathan A. Fox. Development of the EEG from 5 months to 4 years of age. *Clinical Neurophysiology*, 113(8):1199–1208, 2002. ISSN 1388-2457. doi: https://doi.org/10.1016/S1388-2457(02)00163-3. URL http://www.sciencedirect.com/science/article/pii/S1388245702001633.

Hengameh Marzbani, Hamid Reza Marateb, and Marjan Mansourian. Neurofeedback: a comprehensive review on system design, methodology and clinical applications. *Basic and Clinical Neuroscience*, 7(2):143, 2016.

Martin McKeown, Colin Humphries, Peter Achermann, Alexander Borbély, and Terrence Sejnowsk. A new method for detecting state changes in the EEG: exploratory application to sleep data. *Journal of Sleep Research*, 7(S1):48–56, 1998.

Jesus Minguillon, M. Angel Lopez-Gordo, and Francisco Pelayo. Trends in EEG-BCI for daily-life: Requirements for artifact removal. *Biomedical Signal Processing and Control*, 31: 407–418, 2017.

Thomas Minka. Estimating a Dirichlet distribution. Technical report, 2000.

Andrea Mognon, Jorge Jovicich, Lorenzo Bruzzone, and Marco Buiatti. ADJUST: An automatic EEG artifact detector based on the joint use of spatial and temporal features. *Psychophysiology*, 48(2):229–240, 2011.

Pablo G. Moreno, Yee Whye Teh, Fernando Perez-Cruz, and Antonio Artés-Rodríguez. Bayesian nonparametric crowdsourcing. *arXiv preprint arXiv:1407.5017*, 2014.

Barzan Mozafari, Purnamrita Sarkar, Michael J. Franklin, Michael I. Jordan, and Samuel Madden. Active learning for crowd-sourced databases. *arXiv preprint arXiv:1209.3686*, 2012.

Jafar Muhammadi, Hamid Reza Rabiee, and Abbas Hosseini. Crowd labeling: a survey. *arXiv preprint arXiv:1301.2774*, 2013.

Tim Mullen, Christian Kothe, Yu Mike Chi, Alejandro Ojeda, Trevor Kerth, Scott Makeig, Gert Cauwenberghs, and Tzyy-Ping Jung. Real-time modeling and 3D visualization of source dynamics and connectivity using wearable EEG. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2184–2187, July 2013. doi: 10.1109/EMBC.2013.6609968.

Tim R. Mullen, Christian Kothe, Yu Mike Chi, Alejandro Ojeda, Trevor Kerth, Scott Makeig, Tzyy-Ping Jung, and Gert Cauwenberghs. Real-time neuroimaging and cognitive monitoring using wearable dry EEG. *IEEE Transactions on Biomedical Engineering*, 62:2553–2567, 2015.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress. ISBN 978-1-60558-907-7. URL http://dl.acm.org/citation.cfm?id=3104322.3104425.

Kannathal Natarajan, Rajendra Acharya, Fadhilah Alias, Thelma Tiboleng, and Sadasivan K. Puthusserypady. Nonlinear analysis of EEG signals at different mental states. *BioMedical Engineering OnLine*, 3(1):7, 2004.

Søren F.V. Nielsen, Mikkel N. Schmidt, Kristoffer H. Madsen, and Morten Mørup. Predictive assessment of models for dynamic functional connectivity. *NeuroImage*, 2017.

Hugh Nolan, Robert Whelan, and R.B. Reilly. FASTER: fully automated statistical thresholding for EEG artifact rejection. *Journal of Neuroscience Methods*, 192(1):152–162, 2010.

Borna Noureddin, Peter D. Lawrence, and Gary E. Birch. Online removal of eye movement and blink EEG artifacts using a high-speed eye tracker. *IEEE Transactions on Biomedical Engineering*, 59(8):2103–2110, 2012.

Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

Alejandro Ojeda, Nima Bigdely-Shamlo, and Scott Makeig. MoBILAB: an open source toolbox for analysis and visualization of mobile brain/body imaging data. *Frontiers in Human Neuroscience*, 8, 2014.

Julie Onton and Scott Makeig. Information-based modeling of event-related brain dynamics. *Progress in Brain Research*, 159:99–120, 2006.

Julie Onton, Marissa Westerfield, Jeanne Townsend, and Scott Makeig. Imaging human EEG dynamics using independent component analysis. *Neuroscience & Biobehavioral Reviews*, 30 (6):808–822, 2006.

Robert Oostenveld, Pascal Fries, Eric Maris, and Jan-Mathijs Schoffelen. FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Computational Intelligence and Neuroscience*, 2011:1:1–1:9, January 2011. ISSN 1687-5265. doi: 10.1155/2011/156869. URL http://dx.doi.org/10.1155/2011/156869.

Dave Orr. 50,000 lessons on how to read: a relation extraction corpus, April 2013. URL https://research.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html.

Jason A. Palmer, Kenneth Kreutz-Delgado, Scott Makeig, et al. Super-gaussian mixture source model for ICA. In *ICA*, pages 854–861. Springer, 2006.

Jason A. Palmer, Scott Makeig, Kenneth Kreutz-Delgado, and Bhaskar D. Rao. Newton method for the ICA mixture model. In *ICASSP*, pages 1805–1808, 2008.

Roberto D. Pascual-Marqui, Dietrich Lehmann, Thomas Koenig, Kieko Kochi, Marco C.G. Merlo, Daniel Hell, and Martha Koukkou. Low resolution brain electromagnetic tomography (LORETA) functional imaging in acute, neuroleptic-naive, first-episode, productive schizophrenia. *Psychiatry Research: Neuroimaging*, 90(3):169–179, 1999.

William D. Penny, Richard M. Everson, and Stephen J. Roberts. Hidden Markov independent component analysis. In *Advances in independent component analysis*, pages 3–22. Springer, 2000.

William D. Penny, Karl J. Friston, John T. Ashburner, Stefan J. Kiebel, and Thomas E. Nichols. *Statistical parametric mapping: the analysis of functional brain images*. Elsevier, 2011.

Luca Pion-Tonachini, Sheng-Hsiou Hsu, Scott Makeig, Tzyy-Ping Jung, and Gert Cauwenberghs. Real-time EEG source-mapping toolbox (REST): Online ICA and source localization. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 4114–4117. IEEE, 2015.

Luca Pion-Tonachini, Scott Makeig, and Ken Kreutz-Delgado. Crowd labeling latent Dirichlet allocation. *Knowledge and Information Systems*, 53(3):749–765, 2017.

Luca Pion-Tonachini, Sheng-Hsiou Hsu, Chi-Yuan Chang, Tzyy-Ping Jung, and Scott Makeig. Online automatic artifact rejection using the real-time EEG source-mapping toolbox (REST). In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 106–109. IEEE, 2018.

Luca Pion-Tonachini, Ken Kreutz-Delgado, and Scott Makeig. ICLabel: An automated electroencephalographic independent component classifier, dataset, and website. *NeuroImage (under review)*, 2019a.

Luca Pion-Tonachini, Ken Kreutz-Delgado, and Scott Makeig. The ICLabel Dataset of electroencephalographic (EEG) independent component (IC) features. *Data in Brief (under review)*, 2019b.

Lutz Prechelt. *Early Stopping — But When?*, pages 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_5. URL https://doi.org/10.1007/978-3-642-35289-8_5.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Thea Radüntz, Jon Scouten, Olaf Hochmuth, and Beate Meffert. Automated EEG artifact elimination by applying machine learning algorithms to ICA-based features. *Journal of Neural Engineering*, 14(4):046004, 2017.

Stephen J. Roberts and William D. Penny. Mixtures of independent component analysers. In *International Conference on Artificial Neural Networks*, pages 527–534. Springer, 2001.

Gonzalo Safont, Addisson Salazar, Luis Vergara, Enriqueta Gomez, and Vicente Villanueva. Probabilistic distance for mixtures of independent component analyzers. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2017. ISSN 2162-237X. doi: 10.1109/TNNLS.2017.2663843.

Addisson Salazar, Luis Vergara, and Ramón Miralles. On including sequential dependence in ICA mixture models. *Signal Processing*, 90(7):2314–2318, jul 2010a. ISSN 01651684. doi: 10.1016/j.sigpro.2010.02.010.

Addisson Salazar, Luis Vergara, Arturo Serrano, and Jorge Igual. A general procedure for learning mixtures of independent component analyzers. *Pattern Recognition*, 43(1):69–85, jan 2010b. ISSN 00313203. doi: 10.1016/j.patcog.2009.05.013.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

Joan Santamaria and Keith H. Chiappa. The EEG of drowsiness in normal adults. *Journal of Clinical Neurophysiology*, 4(4):327–382, 1987.

Issei Sato, Hisashi Kashima, and Hiroshi Nakagawa. Latent confusion analysis by normalized gamma construction. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1116–1124, 2014.

Gideon Schwarz et al. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464, 1978.

Aashish Sheshadri. A collaborative approach to IR evaluation. Master's thesis, The University of Texas at Austin, 2014.

Aashish Sheshadri and Matthew Lease. SQUARE: A benchmark for research on computing crowd consensus. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008.

Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised MAP inference for image super-resolution. *CoRR*, abs/1610.04490, 2016. URL http://arxiv.org/abs/1610.04490.

Akash Srivastava, Lazar Valkoz, Chris Russell, Michael U. Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318, 2017.

Tatiana A. Stroganova, Elena V. Orekhova, and Irina N. Posikera. EEG alpha rhythm in infants. *Clinical Neurophysiology*, 110(6):997–1012, 1999. ISSN 1388-2457. doi: https://doi.org/10.1016/S1388-2457(98)00009-1. URL http://www.sciencedirect.com/science/article/pii/S1388245798000091.

Becky Su, Shouhei Shirafuji, Tomomichi Oya, Yousuke Ogata, Tetsuro Funato, Natsue Yoshimura, Luca Pion-Tonachini, Scott Makeig, Kazuhiko Seki, and Jun Ota. Source separation and localization of individual superficial forearm extensor muscles using high-density surface electromyography. In *Micro-NanoMechatronics and Human Science (MHS), 2016 International Symposium on*, pages 1–7. IEEE, 2016.

Gabriella Tamburro, Patrique Fiedler, David Stone, Jens Haueisen, and Silvia Comani. A new ICA-based fingerprint method for the automatic removal of physiological artifacts from EEG recordings. *PeerJ*, 6:e4380, 2018.

Wei Tang and Matthew Lease. Semi-supervised consensus labeling for crowdsourcing. In *ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR)*, pages 36–41, 2011. URL http://www.ischool.utexas.edu/~ml/papers/tang-cir11.pdf.

Mario Giovanni Terzano, Liborio Parrino, Arianna Smerieri, Ronald Chervin, Sudhansu Chokroverty, Christian Guilleminault, Max Hirshkowitz, Mark Mahowald, Harvey Moldofsky, Agostino Rosa, Robert Thomas, and Arthur Walters. Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep. *Sleep medicine*, 3(2):187–99, mar 2002. doi: 10.1016/S1389-9457(02)00003-5.

Nelson J. Trujillo-Barreto, Eduardo Aubert-Vázquez, and Pedro A. Valdés-Sosa. Bayesian model averaging in EEG/MEG imaging. *NeuroImage*, 21(4):1300–1319, 2004.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

Jose Antonio Urigüen and Begoña Garcia-Zapirain. EEG artifact removal—state-of-the-art and guidelines. *Journal of Neural Engineering*, 12(3), 2015.

Dimitri Van de Ville, Juliane Britz, and Christoph M. Michel. EEG microstate sequences in healthy humans at rest reveal scale-free dynamics. *Proceedings of the National Academy of Sciences*, 107(42):18179–18184, 2010.

Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 689–692, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2807412. URL http://doi.acm.org/10.1145/2733373.2807412.

Diego Vidaurre, Andrew J. Quinn, Adam P. Baker, David Dupret, Alvaro Tejero-Cantero, and Mark W. Woolrich. Spectrally resolved fast transient brain states in electrophysiological data. *NeuroImage*, 126:81–95, 2016.

Diego Vidaurre, Romesh Abeysuriya, Robert Becker, Andrew J. Quinn, Fidel Alfaro-Almagro, Stephen M. Smith, and Mark W. Woolrich. Discovering dynamic brain networks from big data in rest and task. *NeuroImage*, 2017.

Filipa Campos Viola, Jeremy Thorne, Barrie Edmonds, Till Schneider, Tom Eichele, and Stefan Debener. Semi-automatic identification of independent components representing EEG artifact. *Clinical Neurophysiology*, 120(5):868–877, 2009.

Hanna M. Wallach, David M. Mimno, and Andrew McCallum. Rethinking LDA: Why priors matter. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3854-rethinking-lda-why-priors-matter.pdf.

Hanna Megan Wallach. *Structured topic models for language*. PhD thesis, University of Cambridge, 2008.

Xiaogang Wang and Eric Grimson. Spatial latent Dirichlet allocation. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1577–1584. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3278-spatial-latent-dirichlet-allocation.pdf.

Yi Wang, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y. Chang. *PLDA: Parallel Latent Dirichlet Allocation for Large-Scale Applications*, pages 301–314. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-02158-9. doi: 10.1007/978-3-642-02158-9_26. URL http://dx.doi.org/10.1007/978-3-642-02158-9_26.

Peter Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, June 1967. ISSN 0018-9278. doi: 10.1109/TAU.1967.1161901.

Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The multidimensional wisdom of crowds. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, NIPS'10, pages 2424–2432, USA, 2010. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2997046.2997166.

Jacob Whitehill, Ting fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In

Advances in Neural Information Processing Systems, volume 22, pages 2035–2043. Curran Associates, Inc., 2009.

Diane Whitmer, Camille De Solages, Bruce C. Hill, Hong Yu, Jaimie M. Henderson, and Helen Bronte-Stewart. High frequency deep brain stimulation attenuates subthalamic and cortical rhythms in parkinson's disease. *Frontiers in Human Neuroscience*, 6:155, 2012.

Andrew T. Wilson and Peter A. Chew. Term weighting schemes for latent Dirichlet allocation. In *Human Language Technologies: The 2010 annual conference of the North American Chapter of the Association for Computational Linguistics*, pages 465–473. Association for Computational Linguistics, 2010.

Irene Winkler, Stefan Haufe, and Michael Tangermann. Automatic classification of artifactual ICA-components for artifact removal in EEG signals. *Behavioral and Brain Functions*, 7(1): 30, 2011.

Irene Winkler, Stephanie Brandl, Franziska Horn, Eric Waldburger, Carsten Allefeld, and Michael Tangermann. Robust artifactual independent component classification for BCI practitioners. *Journal of Neural Engineering*, 11(3):035013, 2014.

Feng Yan, Ningyi Xu, and Yuan Qi. Parallel inference for latent Dirichlet allocation on graphics processing units. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 2134–2142. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3788-parallel-inference-for-latent-dirichlet-allocation-on-graphics-processing-units.pdf.

Chi Zhang, Li Tong, Ying Zeng, Jingfang Jiang, Haibing Bu, Bin Yan, and Jianxin Li. Automatic artifact removal from electroencephalogram data based on a priori artifact information. *BioMed Research International*, 2015, 2015.