LEARNING AND EXPLOITING RECURRENT PATTERNS IN NEURAL DATA

By

AUSTIN J. BROCKMEIER

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2014

To all my teachers and coaches, including those who helped me on their own time

ACKNOWLEDGMENTS

I thank my advisor José C. Príncipe for guiding and challenging me, and allowing me the time and freedom for my research to mature. I thank my committee members for their oversight and perspectives. I want to thank Justin C. Sanchez and Joseph T. Francis, and their laboratories, for collaboration and data sharing.

I thank my research mentors Memming, Sohan, and Luis for their guidance and insight on many of the topics in this dissertation. I thank my teammates Lin, Jihye, Matthew, and Kan; my collaborators who collected and explained their datasets to me John, Adi, Chelly, Brandi, Mulugeta, Pratik, Noe, Eric, and Babak; and my other coauthors Evan, Stefan, Bilal, Mehrnaz, and Eder. I want to thank Rakesh and Rosha for accompanying me in the dissertation preparation process.

I want to thank the whole of CNEL—those who were visitors, those who graduated before me, and the current members—for all of the memories, fun, and inspiration. I also want to acknowledge the friends I have met in Gainesville and from Gator Wesley. I would like to give a heartfelt thanks to my family and friends as a source of cheer over the phone or during visits. I thank my wife for her joyful heart and encouragement.

I thank the National Science Foundation and the Japan Society for the Promotion of Science for providing funding and travel support for my research stay in Japan. I thank Andrzej Cichocki for hosting me in his laboratory; Anh Huy Phan, Honjo Wakako, Tanvir, Jarek, and the rest of the laboratory members, and also the other BSI laboratories. I thank Takao Utashiro for his hospitality.

I thank the state of Florida and the University of Florida for providing financial support through the Graduate School Fellowship, and the U.S. taxpayer for supporting this work through DARPA Contract N66001-10-C-2008. I also want to thank the administration and staff of the Department of Electrical and Computer Engineering. Finally, I thank all the anonymous reviewers who improved, rejected, complimented, or simply read my work.

TABLE OF CONTENTS

page

ACKNOWLEDGMENTS 4

LIST OF TABLES 9

LIST OF FIGURES 10

ABSTRACT 13

CHAPTER

1 INTRODUCTION 15

1.1 Neural Signals 16
1.1.1 Neural Spiking Activity 16
1.1.1.1 Electrophysiological recordings 17
1.1.1.2 Spike train binning 18
1.1.2 Neural Electrical Potentials 18
1.2 Challenges of Neural Decoding 20
1.2.1 Neural Signal Variability 21
1.2.2 Disparity in the Information Rate 22
1.3 Decoding Neural Data 22
1.4 Literature Review on Learning Representations for Neural Data 24
1.4.1 Low-dimensional Representations Are More Informative 26
1.4.2 Unsupervised Dimensionality Reduction 27
1.4.3 Dynamical Modeling 28
1.4.4 Supervised Dimensionality Reduction 28
1.4.5 Decompositions of Spatiotemporal Signals 29
1.4.6 Representations for Spike Trains 29
1.5 Aims 31
1.5.1 Unsupervised Learning for Neural Firing Rate Patterns 32
1.5.2 Linear Models of Neural Electric Potentials 33
1.5.3 Optimized Representations 34
1.5.4 Matching the Neural Complexity 35

2 UNSUPERVISED ANALYSIS OF POPULATION FIRING RATES 36

2.1 Learning without Supervision 37
2.1.1 Clustering 38
2.1.1.1 Vector quantization 39
2.1.1.2 Clustering via a soft assignment function 40
2.1.1.3 Graph-theoretic clustering 41
2.1.2 Non-linear Dimensionality Reduction 42
2.1.2.1 Stochastic neighborhood embedding 43
2.2 Clustering for State Estimation in the Nucleus Accumbens 45

2.2.1 Nucleus Accumbens Data . . . 45
2.2.2 Data Representation . . . 46
2.2.3 Clustering . . . 46
2.2.4 Results . . . 47
2.2.5 Discussion . . . 47
2.3 Nonlinear Dimensionality Reduction for Motor Cortex Representations during Reaching Tasks . . . 49
2.3.1 Data Collection . . . 49
2.3.2 Results . . . 50
2.3.3 Discussion . . . 51
2.4 Summary . . . 52

3 LEARNING RECURRENT PATTERNS IN TIME SERIES . . . 53

3.1 Modeling Systems Excited by Sparse Signals . . . 57
3.1.1 Analysis: Estimating the Sources . . . 58
3.1.2 Discrete Time Synthesis and Analysis . . . 59
3.2 Matrix-based Deconvolution and Demixing . . . 59
3.2.1 Deconvolution . . . 60
3.2.2 Multiple Source, Matrix-based Formulation . . . 61
3.2.2.1 Deconvolution and demixing . . . 62
3.3 Iterative Deconvolution and Demixing . . . 63
3.4 System Identification . . . 67
3.5 Methods for Blind MISO System Identification . . . 68
3.5.1 Independent Component Analysis . . . 69
3.5.1.1 FastICA . . . 69
3.5.1.2 Multiple source case . . . 72
3.5.1.3 Filter subset selection . . . 73
3.5.2 Matching Pursuit with K-SVD . . . 74
3.5.2.1 Block-based approximation . . . 76
3.5.2.2 Greedy approach . . . 77
3.6 Synthetic Experiments . . . 78
3.6.1 Single Source, Blind System Identification . . . 79
3.6.2 Multiple Source, Blind System Identification . . . 80
3.6.3 Discussion . . . 83
3.7 Decomposing Local Field Potentials . . . 84
3.7.1 Single Channel Decomposition . . . 84
3.7.2 Model Complexity . . . 88
3.7.3 Multichannel Decomposition . . . 89
3.8 Summary . . . 91

4 TRIAL-WISE DECOMPOSITIONS FOR NEURAL POTENTIALS . . . 96

4.1 Previous Work . . . 98
4.2 Mathematical Modeling Framework . . . 102
4.2.1 Tensor Representation . . . 102

4.3 Models with Variable Temporal Alignment . . . 106
4.3.1 Windowed Tensor Representation . . . 107
4.3.2 First Model . . . 108
4.3.3 Second Model . . . 108
4.4 Spatial Covariance-based Models . . . 109
4.4.1 Single Source . . . 109
4.4.2 Spatial Subspace . . . 109
4.5 Fitting the Spatiotemporal Models . . . 110
4.5.1 Updating the Temporal Factors . . . 110
4.5.2 Updating the Spatial Factors . . . 111
4.5.3 An Alternating Optimization Algorithm . . . 112
4.6 Model Selection . . . 112
4.7 Using the Models for Classification . . . 113
4.8 Reward Representation in Striatal LFPs during a Reach Task . . . 114
4.8.1 Data Collection and Experimental Setup . . . 114
4.8.2 Model Design . . . 115
4.8.3 Results . . . 116
4.9 Reward Expectation in the Motor Cortex during an Observation Task . . . 120
4.9.1 Data Collection . . . 121
4.9.2 Spatiotemporal Model . . . 123
4.9.3 Peristimulus Time Histograms Aligned to LFP Events . . . 124
4.9.4 Discussion . . . 127
4.10 Model Selection . . . 127
4.11 Summary . . . 131

5 NEURAL DECODING WITH KERNEL-BASED METRIC LEARNING . . . 135

5.1 Metrics and Similarity Functions . . . 138
5.1.1 Neural Data Representation and Metrics . . . 138
5.1.2 Kernels . . . 140
5.1.2.1 Tensor-product kernel . . . 141
5.1.2.2 Infinitely divisible kernels . . . 142
5.1.2.3 Weighted product kernels . . . 143
5.1.2.4 Multivariate Gaussian kernel . . . 143
5.1.2.5 Sum of product kernels . . . 143
5.1.2.6 Kernel matrices . . . 144
5.1.3 Neural Metrics . . . 144
5.2 Kernel-based Metric Learing . . . 145
5.2.1 A Kernel-based Measures of Dependence . . . 145
5.2.1.1 Correntropy coefficient . . . 148
5.2.1.2 Empirical estimation . . . 148
5.2.2 Metric Learning Optimization Using Centered Alignment as an Objective . . . 149
5.3 Benchmark Comparison . . . 151
5.4 Decoding Forepaw Touch Location from Rat Somatosensory Cortex . . . 152

5.4.1 Data Collection . . . 152
5.4.2 Results . . . 154
5.4.2.1 Learning multi-unit spike train metrics . . . 155
5.4.2.2 Learning local field potential metrics . . . 156
5.4.2.3 Discussion . . . 158
5.5 Decoding Reach Target from Monkey Premotor Cortex . . . 160
5.5.1 Results . . . 161
5.5.1.1 Classification across windows of each trial . . . 161
5.5.1.2 Visualization using metric-based embedding . . . 161
5.5.1.3 Effect of training set cardinality on performance of metric learning . . . 162
5.5.1.4 Analysis of spike train unit weights . . . 163
5.6 Metric Learning for Neural Encoding . . . 166
5.7 Summary . . . 168

6 CONCLUSION . . . 170

6.1 Applications to Electroencephalography . . . 170
6.2 Sparse Decompositions of Long-term Recordings . . . 170
6.3 Extensions of the Spatiotemporal Models . . . 171
6.4 Kernel-based Metric Learning . . . 173
6.5 One-stage Does Not Fit All . . . 174

REFERENCES . . . 176

BIOGRAPHICAL SKETCH . . . 196

LIST OF TABLES

Table page

2-1 Classifying rewarding versus non-rewarding trials using cluster labels. 48

2-2 Performance of reach target decoding for latent and original space across time points. 52

3-1 Single-channel approximation performance as proportion of variance explained 86

3-2 Computation time for single-channel filter estimation and approximation on 60 s of data sampled at 666.67 Hz 86

3-3 Multichannel approximation performance as proportion of variance explained on a select channel and across all channels. 93

4-1 Classification performance for decoding object type across features and models 121

4-2 Number of units per condition with statistically better fits for each model 126

4-3 Model performance using 2 components for each dataset. 132

4-4 Model performance using 8 components for each dataset. 133

5-1 Benchmark comparison across UCI datasets using different feature weightings 152

5-2 Comparison of touch site classification accuracy using multi-unit spike train metrics 156

5-3 Comparison of touch site classification accuracy using binned spike trains and Euclidean or Mahalanobis-based metrics 157

5-4 Comparison of touch site classification accuracy using LFPs 157

5-5 Comparison of multi-unit metrics for reach target decoding 162

LIST OF FIGURES

Figure page

1-1 The preprocessing steps and signal flow for extracting spike trains and LFPs . 18

1-2 Diagram of classification and dimensionality reduction methods organized by their complexity and their ability to use information among dimensions 24

1-3 Diagram of learning new representations for neural data 32

1-4 Diagram of four methods for the analysis of spatiotemporal data 33

2-1 Clustering results for the reward expectation experiment 48

2-2 Latent space embedding of samples during the movement portion of the center-out task 51

2-3 The movement trajectories colored by the corresponding neural state's location in the latent space 51

3-1 Source-excitation amplitude distribution 78

3-2 Example of a single-source signal 80

3-3 Single-source blind waveform estimation performance 81

3-4 Multiple waveform, single-channel blind estimation performance 82

3-5 Filters estimated from a single-channel of motor cortex LFP 85

3-6 Analysis of single-channel decomposition of motor cortex LFP 87

3-7 Atomic decomposition of a motor cortex LFP 88

3-8 Proportion of variance explained versus number of atoms for various number of filters 89

3-9 Proportion of variance explained across approximation iterations versus the number of filters 90

3-10 Temporal and spatial filters estimated from multi-channel LFP recording of motor cortex 91

3-11 Analysis of a select channel of the multichannel decomposition of motor cortex LFPs 92

4-1 Diagrams of third-order tensor models for sets of evoked potentials 105

4-2 The range of alignments between the temporal waveform and the signal 107

4-3 Diagram of the optimization process consisting of temporal alignment and tensor decomposition . . . 112

4-4 LFPs are projected to a one-dimensional time series and the root-mean-squared, RMS, power is computed for each trial . . . 116

4-5 Spatial and temporal filters of the spatiotemporal decompositions with different ranks objectives . . . 117

4-6 Multichannel LFPs from the straitum and rank-1, single-atom model approximations of four example trials, two from each reward condition . . . 118

4-7 Illustration of the processing stages for the model using a temporal subspace . 119

4-8 Timing and magnitude scatter plots for atomic decompositions . . . 120

4-9 Nearest-neighbor classifier performance using different features while varying training set size . . . 121

4-10 Spatiotemporal models trained during reward expectation . . . 123

4-11 Multichannel LFPs from the motor cortex and rank-1, single-atom approximations of 4 example trials, 2 from each reward condition . . . 124

4-12 Parameter distribution during reward expectation . . . 124

4-13 Spike trains and histograms for spiking units for which the LFP-event aligned provides a significantly better fit . . . 126

4-14 A unit that fit significantly better to the cue-time aligned model for rewarding conditions and the LFP-event aligned model for non-rewarding trials . . . 127

4-15 Model performance across local field potential datasets . . . 130

4-16 Model performance on P300 EEG data on two subjects . . . 131

5-1 Experimental setup showing motorized lever touching digit 1 on the forepaw . . 153

5-2 Comparison of touch site classification between binned spike count metrics with varying bin sizes versus multi-unit spike train metrics . . . 156

5-3 Comparison of metric-based dimensionality reduction before and after using centered alignment metric learning . . . 158

5-4 Learned weights for spiking unit and temporal precision pairs for the optimized Victor-Purpura spike-train metric . . . 159

5-5 Learned temporal weighting of the optimized local field potential metric for decoding touch site across all datasets. . . . 159

5-6 Metric-based dimensionality reduction on the premotor cortex data: before and after using centered alignment metric learning 163

5-7 Reach target decoding performance across different sizes of training sets . . . 164

5-8 Regression analysis between the optimized weights and different independent variables derived from the spike trains 165

5-9 Using metric learning to estimate the filter for a simple cell model 167

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

LEARNING AND EXPLOITING RECURRENT PATTERNS IN NEURAL DATA

By

Austin J. Brockmeier

May 2014

Chair: José C. Príncipe
Major: Electrical and Computer Engineering

Micro-electrode arrays implanted into the brain record the electrical potentials corresponding to the activity of neurons and neural populations. These recordings can be used to understand how a subject's brain represents different conditions such as external stimuli or movement intention. After learning this association, a subsequent condition can be decoded solely from the neural signals, enabling the brain to directly operate computers or machines, thereby creating a brain-machine interface.

Brain-machine interfaces have the potential to improve as new technology enables concurrent recordings from an increasing number of signals throughout the brain. Neural signals are recorded at multiple scales: the action potentials, or spikes, from individual neurons, and the local field potentials corresponding to neural populations. On these diverse and high-dimensional signals, it is a challenge to pinpoint the indicators of different conditions. In addition, the neural responses have natural variability even for the same condition. Furthermore, it is likely that a portion, or even a majority, of the neural signals may pertain to other cognitive processes. To a naive decoder, this background activity appears as inexplicable noise.

In this study, these challenges are addressed by proposing a set of methods that learn new representations of the neural data. These representations are adapted to both recurrent patterns in the neural signals and the decoding task. These methods include clustering and dimensionality reduction, which label or group reoccurring spatiotemporal

patterns without supervision. Similarly, generative models are used to parsimoniously explain both the spatial and temporal patterns in neural potentials. In particular, models are explored that can account for variability in amplitude, waveform shape, and timing, and exploit spatial filters to separate different conditions. Finally, a new approach for optimizing the distance metrics for population activity is used to exploit information jointly represented across space and time and to highlight the most informative dimensions.

Throughout the study, these tools were applied to neural recordings of both spike trains and local field potentials in different brain regions of animal models. The proposed approaches improve data visualization and decoding performance, aiding researchers in their quest to understand the brain from increasingly complex neural recordings.

CHAPTER 1
INTRODUCTION

Systematic analysis of neural activity is used to investigate the brain's representation of different conditions such as stimuli, actions, intentions, cognitive states, or affective states. The fundamental questions of the analysis is "How is this information represented in the neural response?" and "How can this information be extracted from the neural response?" These complementary questions correspond to the problems of neural encoding and neural decoding.

Both of these questions are relevant to researchers designing brain-machine or brain-computer interfaces (BMIs). Motor brain-machine interfaces attempt to decode upper-limb movements from neural data (Carmena et al., 2003; Lebedev & Nicolelis, 2006; Wessberg et al., 2000). Novel applications of BMI include delivering lost sensation back to the brain (Brockmeier et al., 2012a; Choi et al., 2012; Liu et al., 2010; O'Doherty et al., 2011, 2012; Romo et al., 2000; Schmidt et al., 1996) or deriving the training signal for motor decoders directly from the brain's representation of reward (Mahmoudi & Sanchez, 2011; Pohlmeyer et al., 2014). In these cases, the fundamental role of a BMI is to decode motor intention, location of touch, or cognitive features such as anticipation of reward solely from the recored neural signals.

Electrophysiological recordings use multiple electrodes and high-sampling rates to accurately record neural signals as they vary across time and space. The neural responses may be recorded either invasively or non-invasively. Surgically implanted micro-electrode arrays allow invasive recordings to capture both the timing of action potentials (spike trains) across many neurons, and local field potentials (LFPs) across many electrodes. Only the action potential waveforms of neurons in the close vicinity of the electrode are captured, providing a minute fraction of the neurons contributing in the implanted region.

Estimating the decoding models required by BMIs is difficult on this diverse, high-dimensional data. In addition, there are multiple modalities by which the signal characteristics differ between conditions. These include the energy at certain frequencies, the spatial or temporal patterns of evoked potentials following a stimulus presentation, and the rate or pattern of individual neuronal spiking patterns.

Instead of treating the representation of the neural response as given and attempting to solve a very difficult classification or regression task, it is necessary to explicitly optimize an alternative, possibly low-dimensional, representation of the neural signals. Ideally, with this new representation it will be easier to perform the subsequent classification or regression problem, investigate the important dimensions of the neural response, or gauge the information content relevant to the experimental condition. In this dissertation, a number of algorithms are proposed that learn useful representations of neural signals based on their underlying statistics among responses from the same condition and between responses from different conditions. These approaches are specifically tailored to handle both the complexity and diversity of neural signals and the high-dimensionality found in real neural recordings.

The rest of the introductory chapter is organized as follows: first, some general but intrinsic characteristics of the neural signals of interest are introduced, then specific challenges of decoding neural signals are covered, next a review of approaches for learning alternative representations of neural signals is made, and the chapter is concluded by the discussing the aims and contributions of the dissertation.

1.1 Neural Signals

In this section, some background on the electrophysiological recordings of neuronal activity and neural electrical potentials is presented.

1.1.1 Neural Spiking Activity

Interneuronal communication is often characterized by series of action potentials, or spikes. In very general terms, a relatively large and rapid change in a neuron's potential

is generated by the dynamic activation of multiple types of ion channels (Hodgkin & Huxley, 1952). Action potentials in the peripheral nerve carry information back to the central nervous system via their timing and rate of generation (Adrian, 1926; Liddell & Sherrington, 1924). In the central nervous system, the initial impulse is generated near the neuron's cell body, and the impulse travels along its axon toward where other neurons are synaptically connected. As the relative amplitude of the spike carries less information than the timing (Rieke, 1999), the amplitude of spikes is discarded, and only the spike timing is recorded. Statistically, spike trains are modeled as a point process in time.

1.1.1.1 Electrophysiological recordings

Micro-electrode arrays capture the time-varying extracellular potential at each recording location. Single-unit action potential waveforms, referred to as spikes, are relatively short and impulse-like, thus their contribution is spread over the spectrum; however, spikes can be distinguished by their contribution to the high portion of the spectrum (300Hz to 6 KHz). Spikes are isolated from the high frequency portion of the voltage trace in two steps. Initially, the potential spike times are found by identifying the times when the signal crosses a threshold. Then the spikes from each specific neuron are discriminated based on the shape of the action potential wave-form. This second process is called spike sorting. Sorting involves matching the waveforms surrounding a threshold crossing to a set of shape templates. Defining the shape templates for the different units can be done as either a supervised or unsupervised process. Since noise can also cause a spurious threshold crossing, waveforms that do not sufficiently match any of the templates are discarded. Whether sorted to separate the different units or not, spiking activity no longer contains amplitude information and is encoded solely in the sequence of times called a spike train.

In the last couple decades, it has become possible to sample from large numbers of neurons. This is largely due to the development of multi-electrode arrays, better

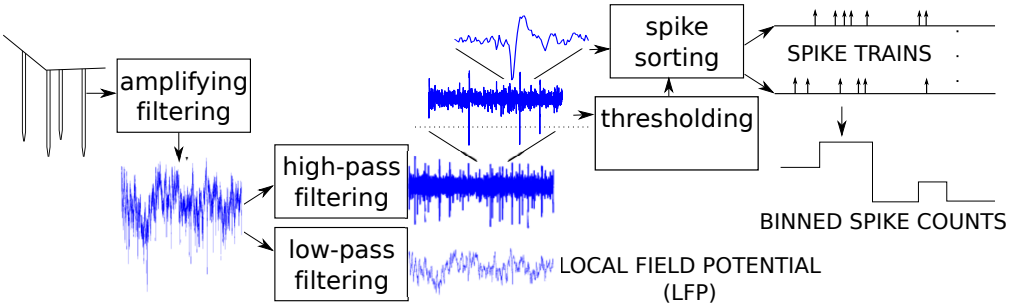


Figure 1-1. The preprocessing steps and signal flow for extracting spike trains and LFPs. After initial amplifying, filtering, and sampling, the signal is low-pass filtered if destined for LFP analysis and high-pass filtered for spike detection. To resolve a spike train, the signal is thresholded and sorted. Using a histogram in time to bin the spike train is an optional final step.

instrumentation, and faster computing that allows the simultaneous recording and processing of multiple neurons. A neuron can spike multiple times in a second, so recording from a large population can yield hundreds of thousands of action potential in the course of 20 minute experiment. This amount of data requires intensive computing for spike identification and spike sorting.

1.1.1.2 Spike train binning

A spike train is a set of real-valued times, but as the number of times may vary, it cannot be directly used in many algorithms that assume fixed length vectors. To use spike times as real-valued vectors, they are first transformed into a discrete-time firing rate function by either smoothing, similar to kernel density estimation, or binning, using a time histogram of non-overlapping windows with a pre-specified bin width, see Figure 1-1. A bin width or kernel width in the hundreds of milliseconds is often used for a smooth estimate of the firing rate.

1.1.2 Neural Electrical Potentials

Brain waves are measurements of the time-varying electrical currents occurring in the brain. Since their discovery in humans (Berger, 1929), brain waves have intrigued researchers as a window into the computation of the brain (Adrian & Matthews, 1934; Buzsáki & Draguhn, 2004; Ciganek, 1961; Farwell & Donchin, 1988; Freeman, 2004;

Sejnowski & Paulsen, 2006). The electrical potential produced by the combined electric fields of neurons in the central nervous system can be measured at varying locations and scales: the electroencephalography (EEG), the electrocorticography (ECoG), and the local field potential (LFP). EEG measures the potentials across the scalp, and can be used to analyze cortical processing. ECoG measures intracranial potentials, and LFPs are a measure of the extracellular potential recorded from penetrating electrodes implanted within brain tissue. These neural potentials are a combination of both local and distributed neural activity. The meaningful frequency ranges for analysis are less than one cycle per second up to hundreds of cycles per second (0.5 to 300Hz). Classically, EEG has been studied in terms of event-locked evoked potentials or in terms of power spectral density of the oscillations across the famous alphabetic frequency bands (alpha, beta, gamma, delta, mu). Neural potentials offer a unique window into the operation of the neural circuits of the brain (Buzsáki et al., 2012).

At each of these scales and recording locations, the electrical potential measurement at each electrode is the superposition of electrical potential from multitudes of individual current sources. For the case of EEG, the measurements are also subject to noise from muscle activity and movement artifacts. To discern the underlying brain activity it is necessary to first separate these sources. Consequently, analysis of the electrical potentials relies heavily on signal processing to resolve the raw signals into usable forms.

Freeman (1975) developed a theory of how masses of neurons generate oscillations in the electric potential, using a series of hierarchical dynamical system models. Essentially, the waveforms in evoked potentials are characterized by the synchronous activity of large numbers of neurons with both excitatory and inhibitory connections having different time constants.

Nunez & Srinivasan (2006) provide a comprehensive view of the physics of the electric fields in the brain, the techniques for recording of electrical potentials, and the

limitations of the technology. In principle, the electric fields of the brain can be described by three-dimensional vector fields. However, recordings are limited to the placement of electrodes and only provide measurement of the scalar electric potential. Nonetheless, understanding the physics behind the generation of the recorded electric potential from the ion currents is essential (Nunez & Srinivasan, 2006).

Due to the relatively slow speed of ion transport by neurons and through diffusion, the magnetic and electric waves in the brain are decoupled. Also, electric fields passively propagating through biological media (with possibly heterogeneous conductance) will only experience linear filtering—that is, they will not experience any non-linear distortion. Consequently, the power of a signal fades as it is recorded farther from its source, with high-frequency experiencing more attenuation (Buzsáki et al., 2012). The power at any frequency cannot increase, nor can it be distributed to other frequencies, and as the distance between the sensors is fixed there is no chance of Doppler effects.

The LFP provides a measure of the electric potential in a localized area of the brain. The LFP is closer to the origin of neural oscillations than either ECoG or EEG measurements. The placement of ECoG electrodes above the cortex and EEG electrodes on the scalp limit which signal sources are measured; the orientation of the electric field means that the signals from some areas of the cortex are poorly captured by these recordings. Often, LFPs are recorded internally with micro-electrode arrays with headstage preamplifiers; consequently, the signals are less prone to movement artifacts and external noise sources. Overall, LFPs offer a unique opportunity to study neural electric potentials nearer their origins.

1.2 Challenges of Neural Decoding

Neural data analysis has many challenges associated with the size, complexity, and unobservable nature of the central nervous system.

- Heterogeneity: The central nervous system in vertebrates is a complex system composed of a richly structured and connected network of neurons and other cells that are specialized in form, function, and location. Even within a specific

cortical region, such as the motor or somatosensory cortex, it is difficult to predict *a priori* how exactly a given neuron may behave to different stimuli or conditions: information can be carried by the precise timing of some neurons' action potentials, and for others, information is carried by their instantaneous firing rate (Rieke, 1999).

- High-dimensionality: Multichannel neural recordings with high-sampling rates yield high-dimensional datasets; yet, they correspond to an extreme subsampling of the implanted area.

- Diversity of representations: The activity of a population of neurons can be represented by a set of timings, locations, and shapes of action potentials; whereas electrical potentials, such as local field potentials, are discretely-sampled, multivariate time-series.

- Network complexity: Neural data analysis has inherent dependencies at different scales and between different scales—e.g., the dependency between individual neurons, between neurons and the local field potential, between the field potentials in different brain regions, etc. There may exist many distinct neural ensembles whose activity is associated with processing similar information; however, the assignment of neurons to ensembles, or other dependencies, have to be inferred solely from their activity without known connectivity.

- Variability: Unable to account for the activity of unobserved neural populations nor the distributed nature of neural circuits, the neural response to repeated conditions appears noisy and variable (de Ruyter van Steveninck et al., 1997). The relative contribution of the background activity can mask the activity relevant to the condition of interest. Mathematically, background activity can only be modeled if it is stationary in some sense, but in the brain this activity may be non-stationary. In addition, long term variability arises from the fact the central nervous system is itself plastic, and adapts over time. This neural plasticity may occur rapidly over the course of seconds or minutes (Fritz et al., 2003).

1.2.1 Neural Signal Variability

When analyzing neural data when there is known external information, such as the timings of stimuli, it may be possible to identify a reliable response in the time-locked average, but the neural response to repeated trials is never wholly consistent. Consequently, computing the average response is the most basic offline analysis approach. This average is referred to as a peristimulus or peri-event time average, or in the case of binned spike trains, the peristimulus time histogram (PSTH).

The average of the time-locked neural response discards much of the underlying response. If there was high-frequency content in the signal that was not phase-locked, or temporal patterns with various temporal alignments, then these patterns will not be present in the average.

The signal itself may be the result of multiple unobserved sources impinging on each recording sensor. Averaging does not provide a means to separate the contribution of the other sources, and cannot be used to separate multiple sources for an individual trial.

1.2.2 Disparity in the Information Rate

Another challenge is the disparity between the information rate of the external information and the sampling rate of the data. The exact timing window in which the neural response represents the condition of interest may be short compared to the length of the recording. Additionally, the lag between stimulus and response may vary (Woody, 1967). It is helpful to identify which time points in the response are important, or the single-trial response time.

For example, in a binary classification task a single bit of external information per trial is available; is it possible to learn the patterns that characterize the signal corresponding to each class? Assume each trial consists of a 4 second window of multichannel recordings sampled at 500Hz. Trying to classify each time point independently is a naive approach. Alternatively, treating the whole segment as a single observation provides only a few training examples in a sparse, high-dimensional space. Thus, there is a need for methods that can extract information somewhere in between these two extremes.

1.3 Decoding Neural Data

This dissertation is motivated by the particular challenges faced when training BMIs (Brockmeier & Príncipe, 2013). As the details of neural signals differ between subject and recording location, a general 'one size fits all' BMI system cannot succeed. Portions

of the complete system may use predefined signal processing blocks, but a subset of system parameters need to be adjusted, through training, for a specific subject. Often this training uses data from prior sessions where the desired responses are known. The system parameters are trained such that the output matches as closely as possible to this desired response. This form of adaptive training is known as supervised learning. The system may remain adaptive with the system parameters adjusted so that performance will be maintained as the underlying neural signals may change over time.

Neural decoders can be used to assess the amount of information the recorded activity carries about the condition of interest. However, it is difficult to distinguish whether poor performance is the result of lack of neural modulation or an inefficient decoder. If the performance is high then it is clear that the neural response carries information about the condition, but in the other case the poor performance may be indicative of poor decoder selection. It is desirable to be able to assess the neural response in a decoder-independent manner in order to understand the degree to which the neural response varies among similar or repeated stimuli on a trial-to-trial basis.

With all of the previously mentioned challenges, it would appear that brains are an enigma, and neural decoding should be impossible. Yet, brains are physical systems with real constraints, redundancies, and dependency between dimensions. In particular, brains are limited in terms of their power output, energy dissipation, and rate of change. Additionally, neural signals must obey the laws of physics—unlike the quantities involved with stock and energy markets. The choice of modeling techniques or processing algorithms should be grounded by these considerations.

In addition, techniques should be chosen based on two considerations: first, the goal of the analysis, whether it is distilling the information to a more interpretable form, or extracting the relevant signals from a set or mixture; and second, the complexity of the signals based on an understanding of the underlying neurophysiology and generating process.

1.4 Literature Review on Learning Representations for Neural Data

The challenges of neural data analysis and neural decoding necessitates well-chosen methods. Ideally methods should do the following: qualitatively capture the intrinsic relationship between the neural modulation and the variable to be decoded, highlight important features or dimensions of the neural response, and improve the performance of subsequent decoding.

While neural data has certain unique characteristics, the analysis is often based on general pattern recognition methods. Figure 1-2 presents a number of pattern recognition methods for classification, clustering, dimensionality reduction, and modeling. Methods are oriented along two axes, organized along the horizontal axis by the underlying complexity in terms of number of parameters and computation time and along the vertical by the methods ability of algorithms to weight or separate specific dimensions or features. Methods with higher decomposition ability are able to attenuate irrelevant dimensions or demix the superpositions of multiple sources.

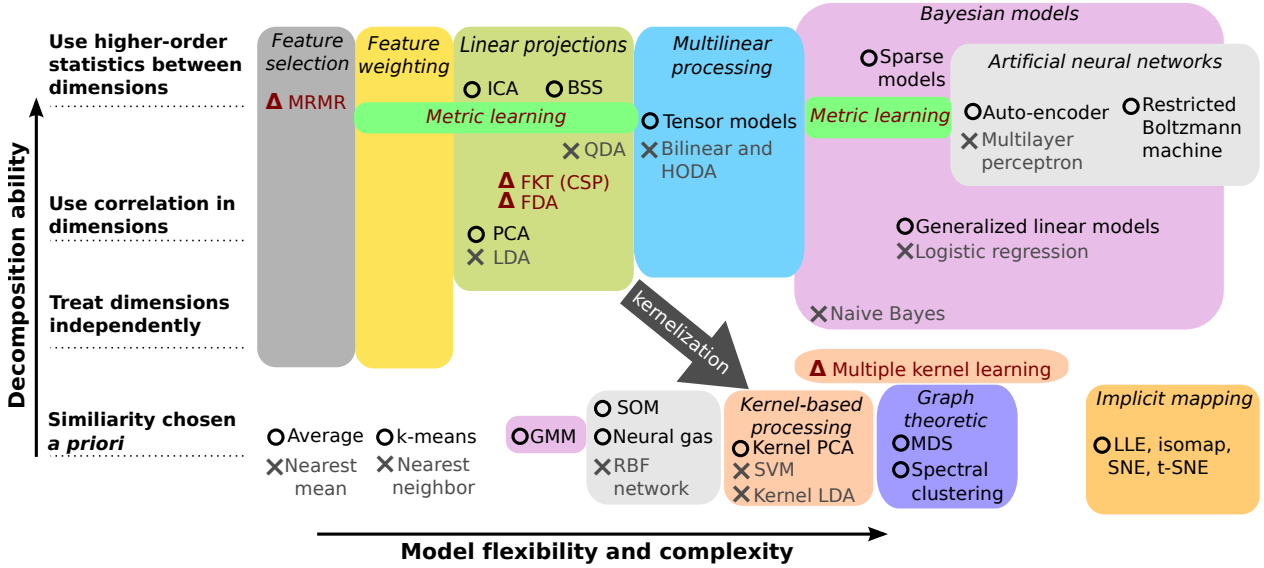


Figure 1-2. Diagram of classification and dimensionality reduction methods organized by their complexity and their ability to use information among dimensions. ◯ indicates an unsupervised method, × indicates a classification method, and Δ indicates a supervised dimensionality reduction approach.

Moving from left to right, the complexity or number of parameters in the algorithms increases. Methods on the right allow non-linear classification decision boundaries or functional mappings. In addition, methods on the right typically require longer computation. Moving from the bottom to the top, methods are able to isolate components or dimensions relevant to a task using supervised information or show statistical structure without supervision (Huber, 1985; Kruskal, 1969, 1972).

A variety of methods consider the formation of features from the dimensions. Starting on the far left, feature selection is the simplest approach as it consists of an inclusion or exclusion decision for each dimension. Although the relative importance of a feature is assessed, this information is only used to select the features. On the contrary, feature weighting explicitly optimizes the relative contribution of the individual dimensions. In addition, feature weighting is applicable whenever there is a underlying distance metric, as is discussed in Chapter 5.

When applicable, linear projections can be used to find linear combinations of the dimensions. More generally, the correlation or dependence between dimensions can also be used as features. Beyond linear projections, techniques for multilinear processing exploit the intrinsic organization of dimensions along multiple modes. For instance, features can be derived via bilinear projections for windows of multivariate time-series (Christoforou et al., 2010; Dyrholm et al., 2007). Bilinear projections use a linear transformation along time and another one across space.

Residing at the bottom of the diagram are methods that require a given similarity measure. This similarity could be defined as inversely proportional to a intrinsic distance function, such as Euclidean distance. Alternatively, the user may provide the distance function, but in any case the methods do not attempt to modify this measure. Among these methods, are ways to summarize data such as averaging and clustering. Correspondingly, nearest-mean and nearest neighbor classifiers use the locations of the samples in the training set without any optimization.

The radial basis function (RBF) network is another method that learns a decision boundary function without changing the underlying measure (Park & Sandberg, 1991). Similarly, Kernel machines (Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002) are learning algorithms that rely on positive definite, bivariate similarity measure. Usually, the user must choose this function along with any of its hyper-parameters, e.g., the variance parameter of the Gaussian kernel.

Above and to the right of kernel machines, multiple kernel learning (Cortes et al., 2012; Lanckriet et al., 2004; Yamada et al., 2013) provides a way to circumvent this difficulty by automatically learning the optimal convex combination of kernels for a given task. Even more flexibility can be achieved using kernel-based metric learning (Brockmeier et al., 2013a; Fukumizu et al., 2004), which allows the kernel functions themselves to be adapted.

In the top right quadrant are the most versatile and powerful methods, such as artificial neural networks. Auto-encoders are one instance of artificial neural networks that are designed to solve non-linear dimensionality reduction problems (Hinton & Zemel, 1993). Although powerful, these methods require *a priori* selection of the architecture. These models typically require extensive computation time to adapt parameters and hyper-parameters using non-convex optimization techniques. In a Bayesian modeling framework, certain models can be formulated that only require convex optimization, particularly with certain choices in generalized linear models (Pillow et al., 2011).

1.4.1 Low-dimensional Representations Are More Informative

Although neural recordings may be very high-dimensional, often stimuli are applied or the behavior is performed in 2–D or 3–D space. This is especially true for motor and tactile experiments. The similarity among the conditions may correspond to similarity among behaviors or stimuli, such as spatial organization of the targets in a reaching task or the location of touches in a somatosensory task. In these cases, it may be

possible to find a low-dimensional representation of the neural responses. If this representation preserves the relationships among the conditions, then there should be natural correspondence between the low-dimensional representation and the conditions. The representation can be used to assess the trial-to-trial variability (Churchland et al., 2010) and the similarity between neural responses to similar conditions—without explicit modeling. The neural response representation may be optimized in either a supervised or unsupervised manner.

1.4.2 Unsupervised Dimensionality Reduction

A number of unsupervised methods have been used for the exploratory analysis of neural data (Broome et al., 2006; Park et al., 2012; Stopfer et al., 2003). Principal component analysis (PCA) can reveal patterns in neural data (Chapin & Nicolelis, 1999), but the results from PCA on neural data may be unsatisfactory (Cowley et al., 2012), as the directions of largest variance may not contain any useful information. Also in the linear case, independent component analysis (ICA) (Comon, 1994) optimizes a linear projection so the resulting vector has maximally independent elements. The activity projected along each of these dimensions may be informative, but unlike the case for PCA, there is not a natural ordering for independent components, and the user is left to assess which components are meaningful.

Other non-linear approaches include distance embeddings (Sammon Jr, 1969), kernel-based extension to PCA (Schölkopf et al., 1998), and manifold learning algorithms that try to preserve the similarity structure between samples in a low-dimensional representation. Such methods tend to concentrate on preserving either local (Roweis & Saul, 2000) or structural-based (Tenenbaum et al., 2000) similarities. For any of these objective functions, novel samples can be mapped to the representation space via explicit mappings (Bunte et al., 2012).

1.4.3 Dynamical Modeling

State-space models with low-dimensional states can be used to model the temporal evolution and dependencies in neural responses. In a generative model, the low-dimensional state corresponds to the latent process generating the high-dimensional neural activity. After training, the state-space models provide a means to explore the temporal evolution and variability of neural responses during single trials. The low-dimensional or discrete state variables, as in hidden Markov models (HMMs), can be visually depicted to track the dynamics of the neural response (Kemere et al., 2008; Radons et al., 1994; Seidemann et al., 1996; Yu et al., 2009, July 2009). Petreska et al. (2011) have shown how a combination of both temporal dynamics and a discrete state can efficiently capture the dynamics in a neural response. Ideally, the estimated states contain information regarding the experimental conditions. For instance, they may be indicative of the start of movement or intention (Shenoy et al., 2003). However, these state-space models are trained in an unsupervised manner and are not guaranteed to capture the aspects of the neural data related to the experimental conditions.

1.4.4 Supervised Dimensionality Reduction

In the supervised case, Fisher discriminant analysis (FDA) and extensions (Baudat & Anouar, 2000; Fukunaga, 1990) use sample covariances from each class to form discriminative projections. The optimal projection is a solution to a generalized eigenvalue problem that maximizes the spread between the means in different classes while minimizing the spread of samples within the same class. Local estimates of the class-covariance can also be used for multimodal distributions (De la Torre & Kanade, 2005; Sugiyama, 2007).

The dimensionality of the neural response can be reduced by feature selection (Kira & Rendell, 1992). The simplest approach is to find how informative each feature is for a given task and then select a set of informative, but not redundant features (Guyon

& Elisseeff, 2003; Peng et al., 2005; Yamada et al., 2013), or a set of features may be obtained by backward or forward-selection algorithms (Song et al., 2007, 2012).

1.4.5 Decompositions of Spatiotemporal Signals

Models of stimulus evoked potentials can be used to separate the signal of interest from background activity (de Munck et al., 2004; Jaskowski & Verleger, 1999; Karjalainen et al., 1999; Li et al., 2009; Pham et al., 1987; Rivet et al., 2009; Souloumiac & Rivet, 2013; Truccolo et al., 2003; Weeda et al., 2012; Woody, 1967). Alternatively, the tools developed for blind-source separation can identify and separate spatially distinct sources (Delorme et al., 2012; Jung et al., 2000). Sources can also be identified by their temporal patterns (Brockmeier et al., 2011b; Douglas et al., 2007; Jmail et al., 2011; Koldovskỳ & Tichavskỳ, 2011; Mijović and et al., 2010). Furthermore, multiway or tensor models of the neural potentials allow for both spatial and temporal factors (Cichocki et al., 2008; Miwakeichi et al., 2004; Mørup et al., 2006).

1.4.6 Representations for Spike Trains

Spikes present a challenge to most processing algorithms that are designed for fixed-length vectors, whereas spike trains are variable length sets. Probabilistically a spike train is a realization of a temporal point process (Brown et al., 2002; Kass & Ventura, 2001). Point process models are incredibly useful for building encoding models of how stimuli affect neural spiking activity, and for decoding stimuli directly from the spiking activity (Brown et al., 1998). Pillow et al. (2011) provide an excellent review and new efficient algorithms for optimizing point process models.

Alternatively, decoding algorithms can be built directly off the geometric structure of the spike trains, without explicit probabilistic modeling, by exploiting measures of similarity and dissimilarity provided by spike train kernels and metrics, respectively. The Victor-Purpura metric (Dubbs et al., 2010; Victor, 2005; Victor & Purpura, 1996) is an edit distance between two spike trains. A key feature of the distance is its temporal precision parameter that adjusts the cost associated with moving spikes in time to

align them, versus simply adding or deleting spikes. The van Rossum distance is the $\mathcal{L}_2$ distance between continuous rate functions (van Rossum, 2001), where the rate functions are estimated by convolving the spike trains with an impulse function consisting of a one-sided exponential decay. Population, also referred to as multi-unit, versions of these single neuron metrics have also been proposed (Aronov, 2003; Houghton & Sen, 2008). The computational neuroscience community is still developing new metrics for spike trains (Rusu & Florian, 2013).

Following the success of kernel machines for machine learning (Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002), and the ability to define kernels on general spaces, such as graphs (Gärtner et al., 2003), researchers have explored spike train kernels (Paiva et al., 2009, 2010; Park et al., 2013, 2012). In the reproducing kernel Hilbert space framework (Aronszajn, 1950), linear algorithms in the Hilbert-space can implement non-linear processing in the input space (Principe, 2010). This is especially important for spike trains where the input space does not permit linear operations. However, after binning, spike trains are indeed vectors in Euclidean space and general kernels or simple linear operations can be applied. Specialized kernels for binned spike trains have also been developed: the alignment-based kernels (Eichhorn et al., 2004) inspired by bioinformatics research on gene alignment and the spikernel (Shpigelman et al., 2005) that efficiently uses population activity.

Although these metrics and kernels exploit the structure in individual neurons, when it comes to population activity the multi-unit approaches are unsatisfactory. Joint measures—such as the tensor product spike train kernel (Li et al., 2012; Park et al., 2013)—use the contribution of all units equally. If only a few of the units are meaningful, then their activity is diluted in the joint kernel. Thus, there is a need of methods which can adapt joint measures for a specific task. Although suggested by Park et al. (2013), there has not been an attempt to learn weighted combinations of spike-train kernels.

1.5 Aims

The aim of this dissertation is to develop methodologies that can learn meaningful representations from raw neural data. The underlying idea is that the raw recorded signals, even with data-independent processing using filtering or other transformations, are insufficient to understand neural activity and to perform neural decoding. Instead the neural signals have to be processed in a data-dependent manner—exploiting the inherent structure of the signals. These ideas are summarized by the following two hypotheses:

The first hypothesis is that recurrent patterns in observed neural signals indicate repeated instances of the same neural processing. By identifying the recurrent patterns, their occurrence can be used for later decoding. There are two paradigms for pattern recognition: summarizing the signal by a discrete or lower dimensional variable, and extracting a component under the assumption that the signal is a linear combination of components.

The second hypothesis is that, for multi-electrode recordings, only a subset of the neural dimensions are relevant for a given decoding task. The relevant dimensions need to be identified and combined in a manner that to maximizes the information relevant to the task. The goal is to learn these new representations based on the statistics of the data and possibly the decoding task. The new representation serves as an intermediate descriptor that is better suited for neural decoding than the raw signals. Specifically, intermediate descriptors should efficiently capture the characteristics of the signals or be estimates of latent, unobserved sources that are combined in the observed signals. A diagram depicting this processing is shown in Figure 1-3.

These hypotheses are matched to the type and complexity of the signals. For populations of spiking neurons, clustering and dimensionality reduction techniques are used to summarize and capture recurring spatiotemporal firing rate patterns of populations of neurons (Brockmeier et al., 2011a, 2010). For local field potentials and

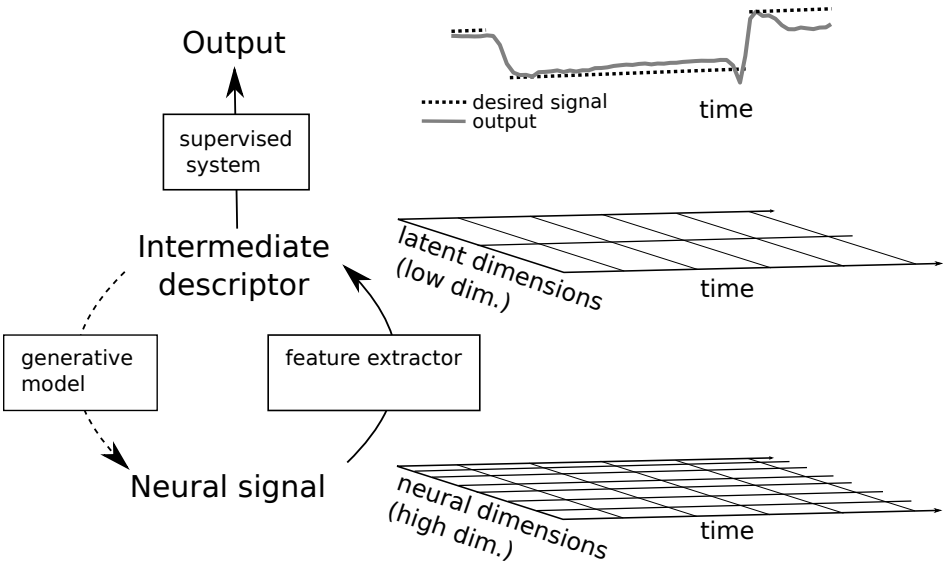


Figure 1-3. Diagram of learning new representations for neural data. An intermediate descriptor is extracted from the signals prior to the classification task.

EEG data, the signals are modeled as an additive combination of background activity and relevant reoccurring patterns (Brockmeier et al., 2012b, 2011b). In the case of spike trains and local field potentials, the important spatial and temporal dimensions are identified (Brockmeier et al., 2014). Graphical representations of these approaches are shown in Figure 1-4.

1.5.1 Unsupervised Learning for Neural Firing Rate Patterns

The first aim is to develop means to summarize multichannel neuron spike rates with a low-dimensional or discrete variable. This representation should allow the single-trial variability and cross-trial reliability to be assessed. Clustering and dimensionality reduction-techniques are the obvious approaches towards this aim. Clustering assigns labels to recurrent patterns during and across multiple task trials. The sequence of labels is used to identify cognitive processing, specifically reward expectation (Brockmeier et al., 2010). Using dimensionality reduction, the trajectories of the high-dimensional neural data are visualized in two- or three-dimensional representations—elucidating whether the neural data has consistent representations

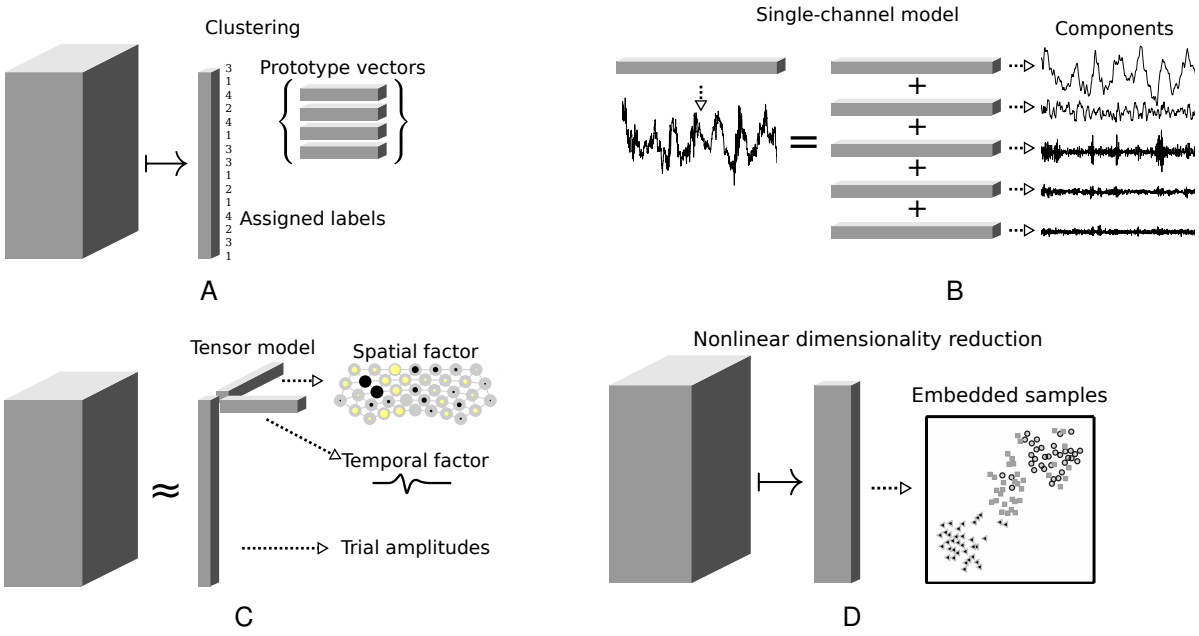


Figure 1-4. Diagram of four methods for the analysis of spatiotemporal data. A) Clustering using spatiotemporal metrics. B) Single channel decomposition of potentials. C)Tensor model for evoked potentials. D) Nonlinear dimensionality reduction using optimized metrics.

across multiple trials of a behavioral task (Brockmeier et al., 2011a). The methods and results for this aim are covered in Chapter 2.

1.5.2 Linear Models of Neural Electric Potentials

The second aim is to extract reoccurring spatiotemporal patterns from neural electric potentials. This is accomplished by using linear synthesis models for the multichannel time-varying signals. Two signal processing schemes are explored for extracting information from local field potential data: the first finds multiple recurrent patterns in continuous recordings, and the second assumes a single pattern per trial.

In the first scheme, the recurrent patterns in single-channel neural potential signals are learned in a completely unsupervised manner on continuous segments. This approach can be used to summarize the neural activity over time. The recurrent waveforms are automatically tuned to characteristics of the particular signals. Specifically, independent component analysis and methods for learning dictionaries for sparse

coding are proposed to blindly estimate the recurrent waveforms of the underlying sources. This expands on preliminary results (Brockmeier et al., 2011b), which used only independent component analysis. Chapter 3 discusses the mathematical framework, synthetic experiments, and results on single-channel LFPs and proposes a simple extension for the multichannel case. Furthermore, the timing and amplitude of the sources of these recurrent patterns is proposed as an natural representation space that may prove useful for analysis of long-term recordings.

In the second scheme, novel models are proposed to explain the spatiotemporal waveforms evoked after stimuli presentations. The techniques proposed in Chapter 4 naturally capture the patterns occurring at random time points and leverage the inherent spatiotemporal aspects of the signals. The timing and amplitude of these waveforms can be used to predict cognitive states. For these models, it is necessary to strike a balance between using simple models to summarize the neural responses and maintaining the diversity in the signal features. To find this balance model selection criteria (Schwarz, 1978; Stoica & Selen, 2004) are applied. The methods build from preliminary results (Brockmeier et al., 2012b), which explored using predefined temporal waveforms and a greedy approach to identify spatial amplitude patterns in EEG on a single-trial basis.

The second goal is more scientific: to use these models to identify coupling across scales—i.e. between neuronal data (spike trains) and electrical potential data. This hypothesis is motivated by the recent results linking the phase of LFP and spiking rate of certain neuron populations Canolty et al. (2010). Instead of using the phase, the relative temporal location of recurrent patterns in the LFP is shown to be more predictive then the task timing for some neurons.

1.5.3 Optimized Representations

The last aim is to develop a general framework for learning better representations of neural data. Metric learning (Fukumizu et al., 2004; Lowe, 1995; Xing et al., 2003) is proposed as a general framework suitable for this task. Within this framework, new

approaches for learning joint kernels are pursued that rely on the well-known connection between metrics and reproducing kernel Hilbert spaces (Schoenberg, 1938). The kernel framework is used not only for kernel machine implementations—to perform non-linear classification and regression—but for its connection with dependency measures (Bach & Jordan, 2003; Cristianini et al., 2002; Gretton et al., 2005) and information-theoretic quantities (Principe, 2010). Recent work (Sanchez Giraldo & Principe, 2013; Sanchez Giraldo et al., 2012) has shown how entropy and dependency measures could be empirically evaluated from kernel matrices, without the need of explicit kernel density estimation. This approach allows Mahalanobis metrics on real-valued vectors to be optimized (Sanchez Giraldo & Principe, 2013). Preliminary work using this method were applied to binned spike trains and LFPs (Brockmeier et al., 2013c). The approach was generalized to learning weighted product kernels (Brockmeier et al., 2013a). Finally, in Chapter 5, a computationally faster dependency measure (Cortes et al., 2012) is used as the objective function to optimize the parameters of multi-unit spike train metrics (Brockmeier et al., 2014). This allows and improves neural decoding algorithms that can directly use spike trains from multiple neurons.

1.5.4 Matching the Neural Complexity

Throughout this dissertation the analysis methods are chosen such that the complexity of the model is matched to the complexity of the neural signals. For instance, if only a subset of neural signals are assumed to be important then a feature selection approach is taken; alternatively, if the temporal alignment is variable between trials then a shift-tolerant model should be employed. In certain cases, the modeling and post-hoc model selection itself can be used to understand the complexity of the data. These last discussion points are meant to emphasize the overarching goal of the dissertation—to develop processing tools that extract information from and better understand neural signals.

CHAPTER 2
UNSUPERVISED ANALYSIS OF POPULATION FIRING RATES

In the analysis of concurrent recordings of multiple neurons, there is a need to identify reoccurring patterns relevant to the behavior, stimulus, or cognitive state. Visualizing the collective modulation of multiple neurons is useful for this exploratory analysis, but visualizing the activity of a large number of neurons at once is challenging. In general, it is difficult to represent the joint neural activity in a single trial basis, that is both understandable and informative.

A common analysis tool, in the case of multiple stereotyped trials, is the peristimulus time histogram (PSTH) (Gerstein & Kiang, 1960), but it only provides the average for a discrete number of time-locked conditions. In addition, each neuron is treated separately, providing little intuition on any dependence between neurons, and it is unable to capture any variability that exists on a trial-to-trial basis (Radons et al., 1994).

One alternative approach is to estimate a low-dimensional latent state variable that corresponds to the high dimensional data. Estimating a latent state from multi-electrode neural data can be a general tool for characterizing the evolution of the population neural activity through time. This can be used to pinpoint the times of reoccurring patterns.

Bayesian modeling is a natural approach for latent state stimation (Yu et al., 2006, July 2009), which can handle both variability and high dimensionality. Other work on state-estimation for multi-electrode recordings studies is based on dynamical models such as hidden Markov models (HMMs), Kalman filters, or other dynamical models (Gat

---

Portions of this Chapter are published for in the following manuscripts: Brockmeier, A.J., Park, I., Mahmoudi, B., Sanchez, J.C., and Principe, J.C. (2010). Spatio-temporal clustering of firing rates for neural state estimation. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 6023-6026; Brockmeier, A.J., Kriminger, E.G., Sanchez, J.C., and Principe, J.C. (2011). Latent state visualization of neural firing rates. In *Neural Engineering (NER), 2011 5th International IEEE/EMBS Conference on*, pages 144–147.

et al., 1997; Kemere et al., 2008; Petreska et al., 2011; Radons et al., 1994; Seidemann et al., 1996; Wu et al., 2004; Xydas et al., 2011; Yu et al., 2009, July 2009). State estimation for brain-machine interfaces can be used to decode movement type (Shenoy et al., 2003) or plan timing (Achtman et al., 2007). A review of methods is taken by Churchland et al. (2007) and Paninski et al. (2010).

These approaches require explicit model formulation and estimation. Instead, we propose to use data-driven approaches for discrete state estimation or continuous state estimation where high-dimensional data are statically projected to a low-dimensional space wherein trial-to-trial variability can be assessed visually.

First, we propose to perform neural state estimation via clustering of the neural firing rates. We use a spatiotemporal representation of the neural firing rates: each sample corresponds to a vector containing all of the sampled neurons' firing rates at the current time and some previous times. However, simply vectorizing the observations and applying a Euclidean distance metric ignores the relative contributions of dimensions at points in space (electrodes) or in time. We investigate a weighted distance metric for the spatiotemporal space.

Next, we further investigate using static dimensionality reduction techniques on neural firing rate data. The dimensionality reduction techniques are used to find a two-dimensional embedding that can be visualized and preserves aspects of the data in its original, high-dimensional space. The method is applied to neural data recorded during a manual center-out reaching task. Meaningful visualization confirm the underlying structure in data, and the lower-dimensional representation is shown to be just as useful as the original data in predicting the reach target. This technique is a straightforward way to extract a useful visualization of the dynamics in neural recordings.

2.1 Learning without Supervision

In this section we consider methods for statistical learning in the unsupervised case where no external information is given to system besides the input. The goal of

unsupervised learning is to learn a function whose output preserves or captures the underlying distribution of the original data. For it to be considered learning, the function should be dependent on the data. By this definition, preprocessing such as Fourier or wavelet transforms are not learning.

Often observations are assumed to be drawn independently and identically distributed from an underlying generative model. In reality this is not the case as observations are taken from one continuous signals—e.g., windows drawn from time-series.

Typical forms of unsupervised learning include clustering, auto-encoders, dimensionality reduction, and sparse/non-negative/independent component analysis. In a generative sense, unsupervised learning identifies latent variables that describe the structure of observed data while conforming to a set of a priori constraints. Here we explore clustering and non-linear dimensionality reduction.

2.1.1 Clustering

Clustering can be described as the unsupervised partitioning of the indices of the observation vectors into a finite number of classes such that some form of similarity is maximized among vectors assigned to a given class and/or the dissimilarity between vectors of different classes is maximized.

For the case of real valued vectors, a single model vector can be used to approximate all vectors assigned to the class. The similarity function used to assign a observation to a class varies, but typically, a shift-invariant measure such as Euclidean distance is used. Alternatively, for vector spaces a gain-invariant measure such as the angle between vectors may be used. In the gain-invariant case the magnitude is discarded. This is useful for waveforms, but is not as useful for neural firing rate vectors where the magnitude may carry the information.

At its simplest, clustering amounts to learning a partition function such that each observation is more similar to those assigned to the same class than those observations

assigned to other classes. In the clustering literature, this similarity is assessed via a linkage function, which can be thought of as a generalized distance function that is defined between two sets of clusters, including singleton sets that correspond to individual samples. For example, centroid linkage assesses the similarity between the observation vector and the prototype vector, that is the average of all vectors already assigned to that class. An alternative is the average linkage, which is the average distance between the observation vector and all the observations assigned to that class. In the case of Euclidean distance these two linkages are equivalent.

Let the assignments of samples to clusters be defined by S, a matrix with a single 1 per column:

$$s_{p,n} = \begin{cases} 1 & f(n) = p \\ 0 & f(n) \neq p \end{cases} \tag{2–1}$$

Alternatively, let $f : [N] \rightarrow [P]$[1] denote the partition function that maps the index of an observation vector to a class index.

2.1.1.1 Vector quantization

In the case of real valued vector, all samples assigned to the same class can be represented by a prototype vector. This is a way to compress the data. Let the set of observation vectors be denoted $X = \{\mathbf{x}_n\}_{n=1}^{N}$ and the set of prototype vectors be denoted $A = \{\mathbf{a}_p\}_{p=1}^{P}$, $P \ll N$. The objective is to maximize the similarity between the observations assigned to each class and their corresponding prototype vector.

The average distance, which for the Euclidean distance is equivalent to mean squared error (MSE) over the observation set, can be used as a cost function:

$$J_{\text{MSE}}(A, f) = \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{a}_{f(n)}\|_2^2 = \text{tr}((X - AS)(X - AS)^{\text{T}}). \tag{2–2}$$

[1] For compactness, the subset of the positive integers $\{1, \dots, n\}$ is denoted as $[n]$.

The prototype vectors that minimize the mean squared error cost function are simply the class centroids—i.e., the averages of the vectors assigned to that class $\mathcal{S}_p = \{n \in [N] : f(n) = p\} = \{n \in [N] : s_{p,n} = 1\}$

$$\mathbf{a}_p = \sum_{n \in \mathcal{S}_p} \frac{1}{|\mathcal{S}_p|} \mathbf{x}_n \quad p \in [P] \tag{2–3}$$

$$\mathbf{a}_p = \frac{\sum_{n \in [N]} s_{p,n} \mathbf{x}_n}{\sum_{n \in [N]} s_{p,n}} \quad p \in [P] \tag{2–4}$$

All that remains is to choose the partition function f, but for a given number of clusters there are N^P choices—an exhaustive search is out of the question. Once candidate prototype vectors are available, the simplest partition function is to return the index of the nearest prototype vector.

$$f(n) = \text{argmin}_{p \in P} \|\mathbf{x}_n - \mathbf{a}_p\| \quad n \in [N] \tag{2–5}$$

Assignment and prototype vector update are the two underlying steps in the k-means algorithm (Hartigan & Wong, 1979; Lloyd, 1982; MacQueen, 1967). The algorithm alternates between adjusting the partition function by the current prototype vectors and updating the prototype vectors based on the current partition function, and runs until convergence. An initial set of prototype vectors can be chosen as a subset of the observation vectors, and multiple initial sets can be tried and the best clustering, in terms of the cost function, can be used.

2.1.1.2 Clustering via a soft assignment function

The use of a hard assignment function $f : [N] \mapsto [P]$ is not required for computing the prototype vectors. For instance, probabilistic modeling assigns the value of $s_{p,n}$ as the posterior probability that observation n is from the pth class. A fuzzy or soft assignment provides another way of computing the prototypes. The prototypes are computed by weighing the contribution of all observation vectors, with vectors nearby the prototype receiving the largest weights. This avoids an explicit cluster assignment.

Let $h : \mathbb{R}^L \mapsto \mathbb{R}^+$ be a bivariate, positive definite function. If there is some function g such that $h(x_i, x_j) = g(x_i - x_j)$, then h is also a shift-invariant function of two variables. Then the assignment matrix can be redefined as

$$s_{p,n} = g(a_p - x_n). \tag{2–6}$$

Gaussian Mean-shift (Cheng, 1995) algorithm is the special case where $g_\sigma(\cdot) = \exp(-\|\cdot\|_2^2/(2\sigma))$, the initial prototype vectors are some subset of the data vectors $A \subseteq X$ (often $A = X$), and Equation (2–4) with Equation (2–6) is used to iteratively update the prototype vectors. Interestingly, mean shift is a localized version of algorithms designed to estimate the geometric median of a set of points (Chandrasekaran & Tamir, 1989).

As mentioned, in a probabilistic setting such as a Gaussian Mixture Model (Duda et al., 2001), $s_{p,n}$ is the posterior probability of x_n being in class c_p—i.e.,

$$s_{p,n} = p(c_p|x_n, A, \theta) = \frac{p(c_p|A, \theta)p(x_n|c_p, A, \theta)}{p(x_n|A, \theta)} \tag{2–7}$$

where the prototype vectors A are the means and the remaining parameters, e.g., covariances and priors, are stored in θ. The Expectation Maximization algorithm for Gaussian Mixture Model consists of alternating the update in Equation (2–7) with the computation of Equation (2–4) for the means along side another update for the covariances and priors $p(c_p|A, \theta) = N^{-1} \sum_{n \in [N]} p(c_p|x_n, A, \theta)$.

2.1.1.3 Graph-theoretic clustering

All three of the previously detailed algorithms rely on multiple iterations to identify prototype vectors that minimize a similarity cost function. As opposed to the aforementioned iterative methods, an analytic approach can be had by posing clustering as equivalent to dividing a weighted graph into disconnected components. In this setting, the edge weight corresponds to a pairwise measure of similarity between sample points. An affinity matrix is a symmetric matrix with entries corresponding to all of the pairwise similarities between sample points. An optimal solution is one that maximizes the sum

of the edges in each separate component, while minimizing the weight of the cut edges. Spectral clustering is a graph-theoretic approach that has yielded success in a variety of domains (Spielman & Teng, 2007).

Any affinity matrix can be formed into a positive definite matrix by means of the graph Laplacian. The affinity matrix is already positive definite if a positive definite kernel $g(\cdot, \cdot)$ is used as the similarity measure. Letting $A = X, P = N$, this affinity matrix corresponds to the matrix S: $s_{p,n} = g(x_n, a_p) \quad p, n \in [P] = [N]$, where $g(x_n, a_p)$ denotes a measure of similarity between x_n and a_p.

As the matrix S defines the weights of a graph (possibly disconnected), the aim of spectral clustering is to cut the graph into k subgraphs such that each subgraph is maximally connected (minimum edge distances). Finding the best cut is the same as finding the best partition function f. Spectral clustering finds a cut, which is suboptimal, but it is guaranteed to be an approximation of the optimal partition (Spielman & Teng, 2007). Essentially, spectral clustering finds a projection of the affinity matrix such that the points representing vectors within the same cluster are closely packed whereas those from distance clusters are separated. Running k-means in this space with $P = k$ will return the partition function f. Here the algorithm by Ng et al. (2002) is used.

2.1.2 Non-linear Dimensionality Reduction

Dimensionality reduction techniques are a class of unsupervised learning algorithms that attempt to find a low dimensional embedding of high dimensional data that preserves aspects of the structure of the original data such as clusters or manifolds. The specific characteristics preserved vary by method such as local distances, local neighborhoods, pseudo-geodesic, or global distances.

Here the goal is to preserve the relative location of sets of observation vectors in the high-dimensional space by their location on a low-dimensional latent space. The mapping from high-dimensional space to low-dimensional space can be an explicit

function or it can implicitly defined. In either case, this mapping should be based on the statistics of data, without any supervision.

Often the goal is to reduce the dimensionality such that the data can be visualized in two or three dimensions. In practice, a variety of cost and objective functions have been proposed that seek to quantify how well the high-dimensional topology is represented in the low-dimensional projection. One of the earliest approaches is Sammon projections (Sammon Jr, 1969). Kohonen's self-organizing maps (SOMS) are artificial neural network implementations that find non-linear mappings between units on a two dimensional lattice and the original space (Kohonen, 1990). More recent static dimensionality reduction techniques include local linear embedding (LLE) (Roweis & Saul, 2000) and t-distributed Stochastic Neighborhood Embedding (t-SNE) algorithm (van der Maaten & Hinton, 2008). The t-SNE algorithm has proven useful for visualizing the aspects of high-dimensional datasets such as clusters and manifolds.

2.1.2.1 Stochastic neighborhood embedding

The t-SNE algorithm uses a probabilistic formulation with the Kullback-Leibler divergence as the cost function. Specifically, all of the pairwise Euclidean distances in both spaces, original and latent, are transformed to densities that represent the probability of points i and j being in the same neighborhood.[2]

In the original space, the joint density is formed as the symmetric combination of the conditional probability of finding j in the neighborhood of i and vice versa (2–9). The conditional density is considered a Gaussian density centered around point i, (2–8), where the scale parameter σ_i is automatically chosen such that the conditional density has a user-defined perplexity, where the logarithm of the perplexity is the Shannon entropy. The perplexity corresponds to a smoothed estimate of the number of neighbors for each point in the original space (van der Maaten & Hinton, 2008).

[2] The approach has been extended to other divergences by Bunte et al. (2012).

In the low-dimensional latent space, the density function centered at each point is the Student's t-distribution with one degree of freedom, i.e., the Cauchy distribution (2–10). Unlike the original space, the choice of scale is arbitrary, but the Cauchy distribution provides a much larger tail than that of the Gaussian. This avoids a "crowding" problem (van der Maaten & Hinton, 2008). In a high dimensional space, many points can exist at the same distance and density, but in the low dimensional space, the same number of points would have to crowd together to be at the same density with a Gaussian distribution. However, with the Cauchy distribution the density falls off much slower with increasing distance, thereby increasing the range of distances that are within a given density range. The increase in range of distances allows the points to spread out resulting in more useful visualizations.

Let $\{\mathbf{x}_i\}_{i=1}^N$ be the points in the original space and $\{\mathbf{a}_i\}_{i=1}^N$ be the points in the embedded space. The original conditional densities are represented by

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_j - \mathbf{x}_i\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_k - \mathbf{x}_i\|^2/2\sigma_i^2)}. \tag{2–8}$$

Then the joint density is

$$p_{ij} = (p_{i|j} + p_{j|i})/2N. \tag{2–9}$$

The embedded space has joint density

$$q_{ij} = \frac{(1 + \|\mathbf{a}_i - \mathbf{a}_j\|^2)^{-1}}{\sum_l \sum_{k\neq l}(1 + \|\mathbf{a}_k - \mathbf{a}_l\|^2)^{-1}}. \tag{2–10}$$

The cost function is the Kullback-Leibler Divergence,

$$C = D_{KL}(P||Q) = \sum_i \sum_{j\neq i} p_{ij}\log(p_{ij}/q_{ij}). \tag{2–11}$$

The algorithm initializes the latent space with a PCA projection, and proceeds to minimize the KL divergence by gradient descent with a momentum term.[3]

2.2 Clustering for State Estimation in the Nucleus Accumbens

In this section, windows of spatiotemporal neural firing rats are clustered during a reward deliver task. The sequence of estimated labels is used to investigate the neural representation preceding reward delivery, and the differences between rewarding and non-rewarding trials are compared.

2.2.1 Nucleus Accumbens Data

This data was collected by Babak Mahmoudi and Justin Sanchez (Mahmoudi & Sanchez, 2011). A microwire electrode array recorded signals from the nucleus accumbens of a rat. Before implantation, the rat was trained in a two-lever choice task. By pressing the target lever, cued by LEDs, the rat received a water reward. Each trial was initiated by the rat with a nose poke.

Here the rat is simply watching a robotic arm move toward one of two levers—the lever on each trial is chosen pseudo-randomly; if the robotic arm moves toward the target lever—indicated by a LED—the rat will receive the water reward, but if the robotic arm moves to the wrong target no reward is given and the rat receives a negative tone. The rat can learn to identify the upcoming reward based on the robot movement.

The data used in the analysis was from a single day's recording where the same target lever was always cued. We analyzed 102 trials; on 43 trials the robot moved to the correct target lever, and the rat received reward; on 59 trials the robot moved to the wrong lever, and the rat received the negative tone. After spike sorting, 43 neurons were isolated. For each trial, we used 15 s of data surrounding the instance the robot stopped moving when it reached the lever. A bin size of 100 ms was selected, which yielded 150 bins per trial.

[3] The code is publicly available at http://homepage.tudelft.nl/19j49/t-SNE.html.

2.2.2 Data Representation

Time embedding is used to define a spatiotemporal space for the clustering. At each time step a vector is composed of the spike counts of all neurons over a window of time

$$\mathbf{x}_i = [r_1(i), r_2(i), \dots, r_n(i), r_1(i-1), r_2(i-1), \dots,$$
$$r_n(i-1), r_n(i-2), \dots, r_n(i-L+1)] \in \mathbb{Z}_+^{n \cdot L}$$

where $r_v(i)$ is bin count for neuron v at time step i, n is the number of neurons, and $L-1$ is the maximum lag of the time embedding. We analyze a set $X = \{\mathbf{x}_1, \dots \mathbf{x}_N\}$ of N such vectors.

The goal is to assign a label corresponding to the state or cluster to the vector at every time step. In order to cluster, we need a distance measure which works across different neurons at different time steps. The simplest approach is to treat all dimensions equally using the Euclidean distance. Another approach is to use smaller weights for the dimensions of the temporal embedding with larger time lags. The intuition behind this approach is that it will put emphasis on the recent data instead of treating all time lags in the window equally, while still providing more information than instantaneous firing rate. Using this idea, a weighted Euclidean distance is defined as

$$d_{\mathrm{w}}(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=0}^{L-1} \mathbf{w}_k \sum_{v=1}^{n} \left(r_v(i-k) - r_v(j-k) \right)^2 \right)^{\frac{1}{2}}, \tag{2–12}$$

where the weights for successive lags are decayed exponentially $w_k = e^{\frac{-k}{L-1}}$ and the time constant is set to the maximum lag of $L-1$.

2.2.3 Clustering

K-means and spectral clustering (Ng et al., 2002) are compared for various embedding dimensions and number of clusters. For spectral clustering, the affinity matrix is formed using a Gaussian kernel $K_\tau(d) = 1/(\sqrt{2\pi}\tau)\exp(d^2/(2\tau^2))$. The

value of τ is found by doing a parameter sweep and use the value with the 'tightest' clusters (Ng et al., 2002) in terms of mean-square error. Both methods of clustering are computationally straightforward, but still require a user-defined number of clusters.

To speed computation, only samples between the start cue and when the robot stopped moving were used in clustering. The remaining samples are assigned to the state corresponding to the nearest cluster center for visualization purposes. In summary, the user-defined parameters for the clustering approach are the time embedding depth L, number of clusters, the time constant for the weighting (here taken as $L - 1$), and the distance measure (2–12).

2.2.4 Results

The state estimation consistently finds recurrent sequences associated with the task timing Figure 2-1. These clusters can be used for single trial classification. Specifically, a trial was classified by the distribution of the cluster labels during the last 2.4 s of robot movement, as the robotic arm reached for the lever. For every trial, a histogram of the labels was computed. Each bin in the histograms was normalized by the number of times that cluster was assigned across all trials and time indexes. Linear discriminant analysis was chosen as an off-the-shelf classification method. Across 1000 Monte Carlo runs, the trials were divided into a training set with two-thirds the trials and the remaining one-third were used for testing. The averages are reported in Table 2-1. The best classification rate is obtained with 8 clusters and an embedding length of 5. In addition, spectral clustering appears to perform better than k-means for all parameter choices.

2.2.5 Discussion

The results presented demonstrate that the discrete state estimation can capture trends in neural data using unsupervised clustering. Since spectral clustering operates only on the affinity matrix, and does not require linear operations such as averaging, it can be applied to data types (such as spike trains) that cannot be clustered with

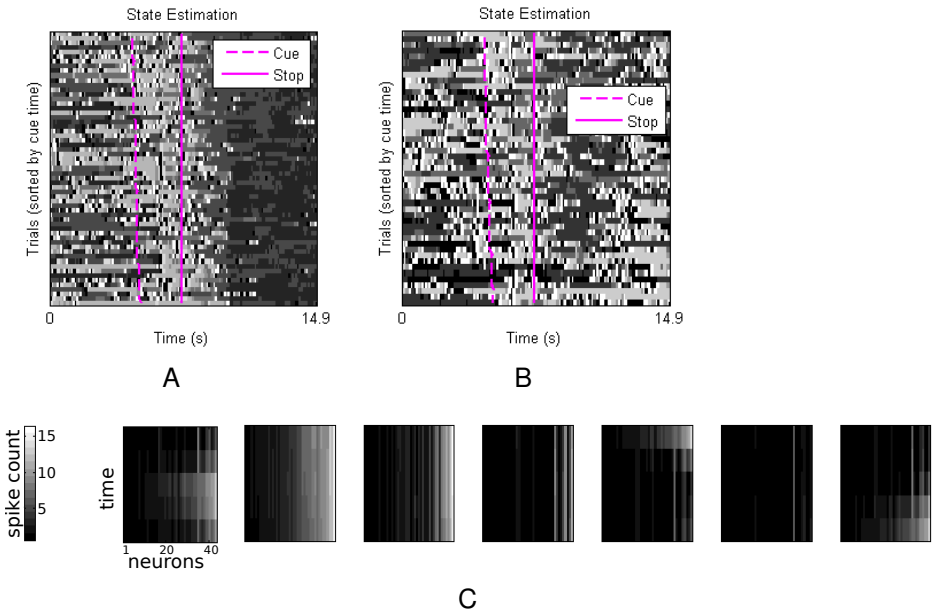


Figure 2-1. Clustering results for the reward expectation experiment. Results are for spectral clustering with 8 classes and a time embedding of 5 samples. Cluster labels are assigned for each time step across all trials and the grayscale color corresponds to an arbitrary state label. A) Cluster assignments during non-rewarding trials. B) Cluster assignments during rewarding trials. C) The spatiotemporal mean for each cluster.

Table 2-1. Classifying rewarding versus non-rewarding trials using cluster labels.

	Spectral clustering					k-means				
L	$k=4$	$k=6$	$k=8$	$k=10$	$k=16$	$k=4$	$k=6$	$k=8$	$k=10$	$k=16$
1	44	52	47	45	47	45	48	47	44	50
5	44	57	58	57	52	45	51	57	51	48
10	50	46	51	52	52	45	50	52	48	45
20	44	48	51	55	53	42	49	45	49	48

Entries indicate the percent of trials correctly classified, averaged across Monte Carlo runs. The maximum standard deviation across the runs is 7.8%. The time embedding dimension is denoted L, which varies across the rows. The number of clusters is denoted k and varies across the columns.

traditional methods. In addition, it is able to capture trends in data with a small number of states. A predefined temporal weighting was used as for the distance metric, but another consideration is how to weigh the contribution of each neurons, as currently they are taken to be independent and equally important. Learning this weighting is explored in Chapter 5.

2.3 Nonlinear Dimensionality Reduction for Motor Cortex Representations during Reaching Tasks

In this section, we use t-SNE to produce a two-dimensional visualization of a subject's neural data while the subject performed a manual reaching task. As the underlying movement is in three-dimensions, it is a reasonable assumption that any relevant dynamics can be preserved in a visualization with equal or smaller dimension. We analyze how well the latent space preserves known external dynamics. Although visualization is inherently qualitative, it is straightforward to quantify how well the two-dimensional embedding can be used to predict the movement.

2.3.1 Data Collection

The data was provided by Pratik Chhatbar and Joseph Francis at SUNY Downstate Medical Center and used with their permission. Brandi Marsh and Shaohua Xu helped record the data. A female bonnet macaque was trained for a center-out reaching task with its right arm in a Kinarm exoskeleton and visual feedback of hand position and virtual targets provided by a computer display. The reaches to 8 virtual targets were constrained to be in a two-dimensional. Every trial started at the center point, and the distance from the center to each of the equally-spaced target was 4 cm. Each reach trial consists of the following steps: center target presentation, subject returns to center target and must hold for 300 ms; reach target presentation, the subject must remain on center target for another 300 ms; center target vanishes, subject reaches to the target; and liquid reward delivery. Successful reaches consist of a maximum duration with a hold period on the target. The vanishing center target serves as the go cue.

After the subject attained about an 80% success rate, micro-electrode arrays were implanted in motor cortex (M1), dorsal premotor (PMd), and somatosensory cortex (S1) (Chhatbar et al., 2010). The M1 data was previously analyzed using other decoding methods (Bae et al., 2011). Animal surgery was performed under the Institutional

Animal Care and Use Committee (IACUC) regulations and assisted by the Division of Laboratory Animal Resources (DLAR) at SUNY Downstate Medical Center.

The data from each spiking unit is a sequence action potential timings. Instead of using the exact time of each action potential, we use a time histogram and count only the spikes in contiguous non-overlapping fixed-width bins with a bin width in the tens to hundreds of milliseconds. At each time step a vector is composed of the spike counts of all neurons $\mathbf{x}_i = [r_1(i), r_2(i), \ldots, r_n(i)] \in \mathbb{R}^n$ where $r_j(i)$ is the count for the jth neuron at the ith time step, and n is the number of neurons.

One of the difficulties of using the time histogram is choosing a bin width that captures the dynamics while reducing the variability. This problem is well-known and possible solutions exist when all the trials belong to the same class (Shimazaki & Shinomoto, 2007). For simplicity, we use a single choice of 100 ms bins and use a 3-tap moving-average filter on each individual neural channel. In general, different choices of bin size and filter order need consideration depending on the data.

2.3.2 Results

We use the t-distributed stochastic neighborhood embedding (t-SNE) algorithm, as described in Section 2.1.2.1, for dimensionality reduction. For qualitative analysis we use a 2-dimensional latent space mapped to a color-wheel. Thus, each point's location defines its color, and we color the corresponding point of the external dynamics with the same color for easy visual analysis. The results are shown in Figure 2-2 and Figure 2-3.

For a quantitative analysis, we see how well the latent space representation can be used for neural decoding versus the original data. The results in Table 2-2 show that the highest classification rate is achieved by the new representation of motor cortex data: outperforming the original representation by 6 percentage points. It appears that the latent dimension removes some of the noise in the original signal, but for the other areas, which have significantly lower decoding performance, the latent space embedding lowers the performance further.

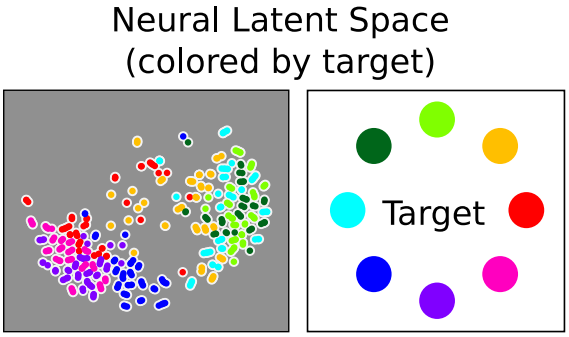


Figure 2-2. Latent space embedding of samples during the movement portion of the center-out task. Each point in the latent space represents a time window during this portion of the task. The points are colored by the reach target. Clear segmentation between left versus right targets and top versus bottom targets is preserved in this mapping.

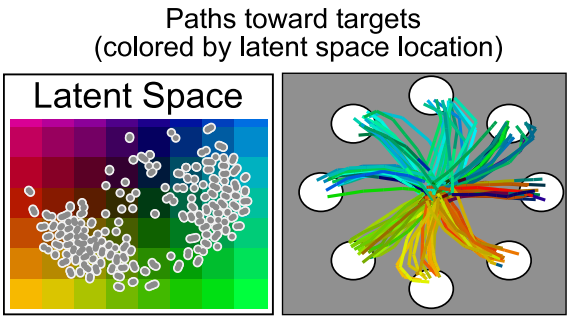


Figure 2-3. The movement trajectories colored by the corresponding neural state's location in the latent space. Similar colors for the same reach target indicate a useful latent space embedding. Position in latent space defines the color of the corresponding movement.

2.3.3 Discussion

The results highlight that similar reach movements have similar neural representation. The visualization also maintains the neighborhoods of the original data, which is useful for prediction. The method requires no explicit model or trial-based averaging. In addition, it can be used with minimal preprocessing, only spike binning and low-pass filtering of the rates. The t-SNE algorithm only requires a single user-defined parameter, the perplexity of the original space. For the movement task, the visualization neatly segmented the different portions of movement. Similar attempts for visualization were made using PCA and ensemble averaging, but these methods were not successful in capturing any distinction between the segments of the movements. Overall, this

Table 2-2. Performance of reach target decoding for latent and original space across time points.

Space	Area	Start cue	Target cue	During reach				Target hold	Reward
Original	M1	9	15	51	54	53	49	44	20
Latent	M1	11	18	31	50	60	52	50	28
Original	PMd	8	11	36	30	25	23	19	7
Latent	PMd	12	11	12	17	17	13	14	15
Original	S1	15	14	23	26	30	27	18	20
Latent	S1	14	12	17	18	22	17	19	18

Entries indicate the accuracy (% correct) for a nearest-neighbor classifier using leave-one-trial-out cross-validation for the 54 trials. The classification rate is for 8 targets, and the chance rate is 12.5%.

technique is useful for exploratory analysis of neural recordings without assumptions or model formulation. However, since the method is stochastic the user is left with no explicit model for the embedding; thus, further modeling that captures the structure seen in the visualization is still required.

2.4 Summary

In this study, we have demonstrated how clustering and non-linear dimensionality reduction—two methods that are fundamental to unsupervised learning—can be used to generate new representation spaces for neural data. The new representation spaces are useful for exploratory data analysis, to generate new hypothesis about structure in the date, and serve as intermediate descriptors for decoding. This approach is in contrast to one-stage decoding where the raw signal is mapped directly to the output. Although the unsupervised learning methods rely on the empirical distribution of the data, they need a predefined and reasonable similarity metric and are unable to automatically determine the relative importance or correlation among the dimensions. Because of this they are equally influenced by all dimensions and cannot separate signals from background noise. Here we have chosen a similarity metric that fits the spatiotemporal nature of the data, but it may not be perfectly tuned. Nonetheless, as fundamental methods they can be used in conjunction with methods that adjust the metric for a specific task.

CHAPTER 3
LEARNING RECURRENT PATTERNS IN TIME SERIES

Sparsely-activated signals are evident in a number of natural signals: neural spike trains (Roux et al., 2009), electrocardiograms (Mailhé et al., 2009), and seismic recordings. Signals such as these have two key distinctions: they consist of the same waveforms appearing repeatedly—but only occasionally—throughout the signal. These time-series signals can be modeled as the sparse excitation of a single filter. A signal of this form is known as shot noise. Mathematically, it corresponds to the convolution of a train of Dirac delta functions, with possibly varying amplitude, with a filter representing the system's impulse response (Papoulis, 1990).

Even more real-world signals are approximated by a combination of shot-noise processes—for example, recordings from an electrode in the brain near multiple neurons. This summation of single-source signals forms a multiple-input single-output (MISO) linear system. In this system, each component has a unique filter, and each source is assumed to be independently and sparsely excited. This model can explain signals such as sound or electromagnetic waves that are emitted in a passive physical environment with a linear medium.

One goal of time-series analysis is to approximate a signal using a small 'dictionary' or basis of interpretable component waveforms, e.g., sinusoids, delta functions, etc. Both Fourier and wavelet analysis are of this form, but they differ by the choice of the basis used in the analysis. Using the correct basis allows both compression and meaningful representations. A dictionary can also be used to decompose a signal into multiple constituent components, which enables denoising and demixing (Bobin et al., 2007; Elad et al., 2005).

The shot-noise model can be seen as a form of time-series analysis where the energy is not only temporally localized and sparsely distributed, but also where the same waveform is recurrent throughout the signal. By only recording the waveform index,

amplitude, and timing—a so-called atomic[1] decomposition (Chen et al., 1998)—this approach provides significant compression. The atomic decomposition is only useful if selected waveforms match the signal such that a good approximation of the whole signal, or a relevant component, can be achieved with a limited number of atoms.

Certain wavelet families are perfectly amenable to the atomic decomposition. However, a small number of coefficients will only be sufficient to approximate the signal if the wavelet family is properly chosen. Instead of leaving the wavelet family as a design choice, the waveforms, essentially the filters of the MISO model, can be learned directly from the data.

There are two problems entwined in this learning problem: learning the dictionary—i.e., the set of filters—to represent the signal, and inferring the index, amplitude, and timing for the train of Dirac deltas, corresponding to the unobserved sources. Together these problems are known as blind deconvolution or blind system identification.

In principle, there are two regimes in which it is possible to identify the inputs to a MISO system from a single output: spectrally disjoint filters (corresponding to sparsity in the frequency domain) or sufficiently sparse input (corresponding to temporally disjoint input). Even without noise and known filters, linear solutions fail in separating spectrally overlapping components. Non-linear analysis techniques that explicitly exploit the sparsity perform better. Matching-pursuit (Mallat & Zhang, 1993) provides an iterative approach for non-linear analysis. Given estimates of the sparse sources, learning the different filters is a system identification problem.

The study's primary contribution is the coverage, comparison, and application of two distinct approaches for blind system identification in the case of sparse sources. The first approach is to assume a generative model for the signal, with constraints in the form of sparse priors for the sources. Using a normal distribution for the noise, the optimal

[1] Each instance of a waveform index, amplitude, and timing is known as an atom.

model is the one that minimizes the least-squares reconstruction cost while complying with the sparsity constraint. Based on prior work in computational neuroscience, this approach is often referred to as sparse coding (Olshausen & Field, 1997). A number of researchers have shown how filters which describe natural signals can be efficiently estimated directly from data (Balcan et al., 2009; Ekanadham et al., 2011; Lewicki, 2002). In particular, using matching-pursuit as a proxy for a maximum a posterior estimate, an efficient algorithm for learning sparse bases (Aharon et al., 2006) has been extended to the time-series case (Mailhé et al., 2008).

The second approach avoids the explicit estimation of the sources and the reconstruction of the signal. Instead the sparse sources' statistical properties are used in the filter estimation (Shalvi & Weinstein, 1990). Essentially, this approach for blind estimation uses a matrix-based projection pursuit, where windows of the time series are treated as vectors in independent component analysis (ICA) (Bell & Sejnowski, 1996). In this way, the filters can be blindly estimated without estimating the sources or using a reconstruction cost function. Other researchers (Davies & James, 2007; Lucena et al., 2011) have demonstrated the ability of FastICA (Hyvarinen, 1999) to efficiently estimate the filters. FastICA is particularly suited for this since its objective implicitly estimates the sources through a non-linear function of the projected signals.

This study has three main aims: first, a systematic analysis of the subproblems involved in blind system identification are covered; second, we make a direct comparison of ICA-based and sparse-coding-based approaches for blind system identification; and third, we apply these algorithms on neural potential data.

After a survey of sparse coding and ICA, we introduce matrix-based algebra used for deconvolution and demixing. We consider blind deconvolution in the case of a single source, and dictionary learning with sparse coding for the multiple source case. Matching pursuit as a solution to sparse coding for time series is reviewed. System identification is covered as a subproblem of system identification given the source

estimates. Alternating optimization algorithms (Blumensath & Davies, 2006; Mailhé et al., 2008; Smith & Lewicki, 2006) are introduced and compared with the ICA approach (Bell & Sejnowski, 1996; Davies & James, 2007; Shalvi & Weinstein, 1990).

We contribute a systematic exploration of blind system identification on synthetic data—considering the limiting case of a single source and single sensor in order to understand the effect of sparsity on both the shift-invariant sparse coding and ICA. The single-source case is an important subproblem in some alternating estimation approaches (Mailhé et al., 2008) wherein the estimation of a single source is performed assuming the contributions from all other sources have been removed. To ease the computation, we propose a fast approximation of matching pursuit for time series. We also introduce a greedy approach that learns each waveform from the residual remaining after removing the previous waveform's component.

Finally, these algorithms are applied to the analysis of neural potential signals, specifically, local field potentials (LFPs) collected from the motor cortex of a non-human primate. Time-frequency analysis, via short-term Fourier analysis or wavelet decompositions, has been the standard tool to analyze neural potential signals. However, linear filtering does not offer the best tradeoff for localizing both the timing and frequency of events. Matching pursuit (Mallat & Zhang, 1993) offers an algorithmic, and thus non-linear, form of analysis. Using a predefined and stochastic basis, matching pursuit has been shown to be an effective, but underutilized tool, for neural signal analysis (Durka & Blinowska, 1995; Durka et al., 2001). Recently Kuś et al. (2013) have made efforts to provide software for the matching-pursuit-based time-frequency analysis. Here we explicitly learn the filters that are then used in the matching pursuit framework for decomposing single-channel neural potentials. In addition, using principal component analysis (PCA) we extend the approaches to multiple channels.

3.1 Modeling Systems Excited by Sparse Signals

A sparse signal stands in stark contrast to both Wiener processes and their discretely-sampled counterpart described by Gaussian distributions.[2] Stochastically, the distribution of source activation times is a point process—i.e., a realization of a point process is a train of Dirac delta function. A train of delta functions with varying amplitudes is a realization from a marked point process. The marked point process is described by both the joint distribution of the timing and amplitude of the impulses.

Let $y(t)$ be a time series created by the sparse excitation of a single filter. The signal is formed by convolving a weighted train of delta functions $s(t) = \sum_i \alpha_i \delta(t - \tau_i)$ with a filter $a(t)$.

$$y(t) = \int_{-\infty}^{\infty} s(t-u)a(u)du \tag{3–1}$$

Let $x(t)$ be a combination of these component signals $\{y_p(t)\}_p$, $p \in \{1, \dots, P\}$ observed in the presence of noise. This signal is created by a multiple-input single-output (MISO) linear system with sparse inputs. Each component, $y_p(t)$, has a unique filter $a_p(t)$:

$$x(t) = e(t) + \hat{x}(t) = e(t) + \sum_{p=1}^{P} y_p(t) \tag{3–2}$$

$$y_p(t) = \int_{-\infty}^{\infty} s_p(t-u)a_p(u)du \tag{3–3}$$

$$s_p(t) = \sum_i \alpha_{p,i}\delta(t - \tau_{p,i}) \quad p = 1, \dots, P. \tag{3–4}$$

The combination of the components is a noise-free model $\hat{x}(t)$.

The atomic representation of the model signal, $\hat{x}(t)$, consists of a set of source indices, amplitudes, and timings $\{(p_i, \alpha_i, \tau_i)\}_i$. Using this set, and the model signal can

[2] Indeed, this difference is exploited in later sections where blind estimation approaches are introduced that rely on this non-Gaussianity as a projection pursuit criterion (Hyvarinen, 1999).

rewritten as:

$$\hat{x}(t) = \sum_i \int_{-\infty}^{\infty} \alpha_i \delta(t - \tau_i - u) a_{p_i}(u) du. \tag{3–5}$$

Similarly, each component signals can be described by the impulse response of the filter $a_p(t)$ and the set of excitation times and amplitudes $\{(\alpha_j, \tau_j)\}_{j \in \mathcal{I}_p}$ where $\mathcal{I}_p = \{i : p_i = p\}$.

Given the atomic representation $\{(p_i, \alpha_i, \tau_i)\}_i$ or the sparse inputs $\{s_p(t)\}, p \in \{1, \dots, P\}$, learning the filters is a system identification problem; however, estimating the sources is the primary challenge.

3.1.1 Analysis: Estimating the Sources

For a single source with a known filter, estimating the source signal is called deconvolution. The problem of deconvolution with sparse sources arises in many physical processes, e.g., biomedical time-series analysis and imaging. For sparse sources, the filtering spreads out the signal's energy in time. Recovering the input amounts to localizing this energy in time. For a low-pass filter, the goal of deconvolution is to delineate the exact timing of the source excitations.

The analysis problem for the MISO system is estimating the full set of sources $\{s_p(t)\}_p$. Even without noise and with known filters, estimating the sources is difficult. Strictly linear solutions suffer from cross-talk between sources whose filters have similar spectra. In general, it is not possible to identify the inputs to a MISO system from a single output, but this is not the case when the filters are only sparsely active. Essentially, sparsity allows estimation through disjoint representations (Georgiev et al., 2005). This principle holds for a range of regimes in which it is possible to identify the inputs from a smaller number of outputs (Zibulevsky & Pearlmutter, 2001). At the extremes of this range are spectrally disjoint filters (corresponding to sparsity in the frequency domain) or sufficiently sparse input (corresponding to temporally disjoint inputs).

A linear solution fails because it only considers the second-order statistics of the sources, which does not contain any information on the sparsity. Non-linear analysis

techniques that explicitly exploit the sparsity perform better, but require many more computations. Essentially, recovering the source as a train of Dirac deltas requires an overcomplete basis of the signal: shifted versions of the underlying filters appearing at each time-shift. With an overcomplete basis, linear analysis is not possible, and non-linear optimization or iterative algorithms are necessary. However, these non-linear approaches for MISO deconvolution require the set of filters $\mathcal{A} = \{a_p(t)\}_p$ to be known. Thus, any solution for blind deconvolution must first identify the system, i.e., the set of filters.

3.1.2 Discrete Time Synthesis and Analysis

For practical digital implementation, we consider the case when the time series (3–2) is discretely sampled, and the convolution operators are replaced by finite summations. The basis for the signal is assumed to be a set of filters $\{\mathbf{a}_p\}_{p=1}^{P}$ excited by the sparse sources $\{\mathbf{s}_p\}_{p=1}^{P}$. Let $\mathbf{x} = [x_1, \dots, x_T]^\mathsf{T}$ denote the observed signal formed by the moving average model

$$\mathbf{x} = \sum_{p=1}^{P} \mathbf{s}_p * \mathbf{a}_p + \mathbf{e}. \tag{3–6}$$

3.2 Matrix-based Deconvolution and Demixing

Consider a component signal $\mathbf{y} = \mathbf{s} * \mathbf{a}$ formed from a single convolution. Assuming $\mathbf{a}$ has a finite impulse response (FIR) with a length less than M, this convolution can be written as

$$y_t = \sum_{\tau=1}^{M} s_{t-\tau+1} a_\tau = \sum_{\tau=0}^{t} s_\tau a_{t-\tau+1} \quad t = 1, \dots, N. \tag{3–7}$$

Let $Y \in \mathbb{R}^{N \times M}$ denote the Toeplitz matrix formed from the time-series vector $\mathbf{y}$, where M is the window size and N is the number of samples.

$$Y = \begin{bmatrix} y_M & y_{M-1} & \cdots & y_1 \\ y_{M+1} & y_M & \cdots & y_2 \\ \vdots & & & \\ y_N & y_{N-1} & \cdots & y_{N-M+1} \end{bmatrix} \tag{3–8}$$

Then the convolution can be expressed in matrix notation as

$$Y = SA = \begin{bmatrix} s_M & s_{M-1} & \cdots & 0 & \cdots & 0 \\ s_{M+1} & s_M & \cdots & s_1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \\ s_{2M-1} & s_{2M-2} & \cdots & s_{M-1} & \cdots & s_1 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ s_N & s_{N-1} & \cdots & s_{N-M} & \cdots & s_{N-2M+2} \end{bmatrix} \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & a_M & \cdots & a_2 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & a_M \end{bmatrix}, \tag{3–9}$$

where $Y \in \mathbb{R}^{N \times M}$, $S \in \mathbb{R}^{N \times (2M-1)}$, and $A \in \mathbb{R}^{(2M-1) \times M}$. This representation is redundant as the $N \times M$ matrix Y is no more than an arrangement of the N distinct values in $\mathbf{y}$.

The rows of A form the support for the rows of Y. Let M' be the actual extent of the filter $\mathbf{a}$. If $M' = 1$ then the filter is a Kronecker delta and A is a complete basis for Y. For $M' > 1$ then A will have $M + M' - 1$ rows that are pairwise linearly independent. For $M' = M$, A is an overcomplete basis for the rows of Y with support along $2M - 1$ distinct vectors.

3.2.1 Deconvolution

Only a single column of S needs to be estimated to determine a time-lagged version of $\mathbf{s}$. Let $\mathbf{w}_\tau$ denote a solution to the deconvolution problem such that

$$Y\mathbf{w}_\tau = SA\mathbf{w}_\tau \approx S\mathbf{e}_\tau = \mathbf{s}(t - \tau + 1). \tag{3–10}$$

where $\mathbf{e}_\tau$ is the τth column of the $(2M-1)\times(2M-1)$ identity matrix, i.e., $\mathbf{e}_\tau$ is the standard basis vector such that $\mathbf{e}_\tau$ is zero except having a 1 for the τth element.

The set of deconvolution FIR filters $W = [\mathbf{w}_1, \ldots, \mathbf{w}_{2M-1}]$ is found as the pseudoinverse of A, $W = A^\dagger$. Let B be a matrix with linearly independent columns then $B^\dagger = (B^\mathsf{T}B)^{-1}B^\mathsf{T}$ is its pseudo-inverse. Each column of W resolves the source at a different lag. The best lag in terms of least squares can be found as $\arg\min_\tau \|A\mathbf{w}_\tau - \mathbf{e}_\tau\|_2$.

3.2.2 Multiple Source, Matrix-based Formulation

Consider the signal $\mathbf{x} = \sum_{p=1}^{P} \mathbf{y}_p = \sum_{p=1}^{P} \mathbf{s}_p * \mathbf{a}_p$ formed via a combination of sources convolved with different filters. Let $X \in \mathbb{R}^{M\times T}$ denote the transpose of the Toeplitz matrix formed from the time-series vector $\mathbf{x}$, where M is the window size and N is the number of samples. (The transposed version is used to match conventions used later on.)

$$X = \begin{bmatrix} x_M & x_{M+1} & \cdots & x_N \\ x_{M-1} & x_M & \cdots & x_{N-1} \\ \vdots & & & \\ x_1 & x_2 & \cdots & x_{N-M+1} \end{bmatrix} \tag{3–11}$$

Assuming that the impulse response of all the filters is less than M, then the convolution in the synthesis of $\mathbf{x}$ can be expressed as

$$X^\mathsf{T} = \sum_{p=1}^{P} Y_p = \sum_{p=1}^{P} S_p A_p = [S_1, S_2, \ldots, S_P] \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_P \end{bmatrix}, \tag{3–12}$$

where $S_p \in \mathbb{R}^{N\times(2M-1)}$, $A_p \in \mathbb{R}^{(2M-1)\times M}$ are Toeplitz matrices formed from $\mathbf{s}_1, \ldots, \mathbf{s}_P$ and $\mathbf{a}_1, \ldots, \mathbf{a}_P$ as in Equation (3–9). This synthesis equation can be compactly written as $X^\mathsf{T} = \bar{S}\bar{A}$ where $\bar{A} = [A_1^\mathsf{T}|A_2^\mathsf{T}|\cdots|A_P^\mathsf{T}]^\mathsf{T}$ and $\bar{S} = [S_1|S_2|\cdots|S_P]$.

3.2.2.1 Deconvolution and demixing

In general, it is not possible to resolve the sources from a single channel of observation even if all the filters are known. First due to the FIR synthesis, FIR analysis is only an approximation. Second, there will be cross-talk between sources unless the filters have disjoint spectral support, i.e., $\forall t \quad (\mathbf{a}_p * \mathbf{a}_q)(t) = 0$, but with FIR synthesis the filters can only have approximately disjoint spectra.

Every block of $2M-1$ columns in $\bar{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_{P(2M-1)}]$ corresponds to lagged versions of one source, $s_k(t) = s_p(t-\tau+1) \quad k = \tau + (2M-1)\cdot(p-1)$. Let $\mathbf{w}_k$ be an approximate solution to the demixing/demodulation problem that extracts the kth column of $\bar{S}$, i.e.,

$$X^{\mathsf{T}}\mathbf{w}_k = \bar{S}\bar{A}\mathbf{w}_k \approx \bar{S}\mathbf{e}_k = \mathbf{s}_k. \tag{3–13}$$

The matrix of FIR filters $W = [\mathbf{w}_1, \ldots, \mathbf{w}_{P(2M-1)}]$ is found as the pseudoinverse of $\bar{A}$. The matrix W is partitioned into 2M-1 blocks corresponding to each source. In each block of W, there is one column such that the corresponding column of $\bar{A}W$ best approximates the corresponding standard basis vector. Choosing each of these columns from all the blocks is the best linear solution—in the least-squares sense—to the multiple source deconvolution problem (3–13).

Alternatively, component demixing problem can be performed without deconvolution. Demixing the components attempts to separate the portions of the signal corresponding to unique sources. Its success is dependent on the cross-correlation between the source filters; if the filters have disjoint spectral support then perfect separation can be achieved (but the FIR filters can never have completely disjoint spectral support).

Let $\bar{Y} = [Y_1|Y_2|\cdots|Y_P] = [\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_{P(2M-1)}]$ for compactness. Every block of $2M-1$ columns in $\bar{Y}$ corresponds to lagged versions of one filtered source, i.e.,

$$\bar{y}_l(t) = (s_p * a_p)(t-\tau+1) = y_p(t-\tau+1) \quad l = \tau + M(p-1). \tag{3–14}$$

Let $\acute{\mathbf{w}}_k$ be an approximate solution to the demixing problem that extracts the kth column of $\bar{Y}$, i.e.,

$$X^{\mathsf{T}}\acute{\mathbf{w}}_k = \bar{S}\bar{A}\acute{\mathbf{w}}_k \approx \bar{S}\bar{\mathbf{a}}_k = \bar{\mathbf{y}}_k \tag{3–15}$$

The full demixing matrix is $\acute{W} = W\acute{A}$ where $\acute{A}$ is a block diagonal matrix with the Toeplitz representations of the filters on the diagonal $\acute{A} = \text{blkdiag}(A_1, A_2, \ldots, A_P) \in \mathbb{R}^{(2M-1)\cdot P \times MP}$. The best lag for the pth source can be chosen as

$$\arg\min_{\tau} \left\| \bar{A}\acute{\mathbf{w}}_{\tau+M(p-1)} - \acute{\mathbf{a}}_{\tau+M(p-1)} \right\|_2 . \tag{3–16}$$

The quality of the separation is dependent on how close the spectral representations of the filter are. Cross-talk from one source to another occurs when the filters are similar, and thus the separation filters will have similar frequency response. Two filters with minimal cross-talk have $(\mathbf{a}_p * \mathbf{a}_q)(t) \approx 0 \quad \forall t$.

3.3 Iterative Deconvolution and Demixing

Matching pursuit as an iterative approach that can be used to solve deconvolution or demixing problems. In order to introduce the algorithm, we first consider a simple least-squares problem:

$$\min_{\mathbf{s}} \|\mathbf{x} - A\mathbf{s}\|_2^2, \tag{3–17}$$

where $\mathbf{x}$ is the vector to approximate, $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_P]$ is a matrix with possibly more columns than rows, and $\mathbf{s}$ represents the coefficients that minimize the least-squares cost. Matching pursuit is an iterative, and greedy, solution to the least-squares problem (Mallat & Zhang, 1993). Often it is posed as having an explicit constraint on sparsity—in terms of the number of non-zero elemetns in $\mathbf{s}$. The l_0-'norm' of a vector can be considered the number of non-zero elements: $\|\mathbf{s}\|_0 = |\{s_i : |s_i| > 0\}|$. In the simplest

case,[3] matching pursuit is an greedy approach to solve

$$\min_{\mathbf{s}\,:\,\|\mathbf{s}\|_0 = L} \|\mathbf{x} - A\mathbf{s}\|_2^2, \tag{3–18}$$

where L is much less than the number of columns in A. Since the least-squares fit can be solved analytically once the set of L columns of A are selected, the problem can be reframed as the selection of the L column indices $p_1, p_2, \ldots, p_L$ that minimize the least-squares cost:

$$\min_{\tilde{A}\in\{[\mathrm{a}_{p_1},\mathrm{a}_{p_2},\ldots,\mathrm{a}_{p_L}]\}} \|\mathbf{x} - \tilde{A}\tilde{A}^{\dagger}\mathbf{x}\|_2^2. \tag{3–19}$$

Matching pursuit is an iterative approach to select these columns and their corresponding coefficients, $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_L$. The steps are detailed in Algorithm 3-1.

Algorithm 3-1. Matching pursuit (MP).

Input: $\mathbf{x}$, $\{a_p\}_{p=1}^{P}$, L
  $\mathbf{r} \leftarrow \mathbf{x}$
  For $i = 1, \ldots, L$ do
    $c_q \leftarrow \frac{1}{\|\mathbf{a}_q\|}\langle \mathbf{a}_q, r\rangle \quad q = 1, \ldots, P$
    $p_i \leftarrow \arg\max_q = |c_q|$
    $\tilde{s}_i \leftarrow c_{p_i}$
    $\mathbf{r} \leftarrow \mathbf{r} - \tilde{s}_i \mathbf{a}_{p_i}$
  End for
Output: $\{(p_i, \tilde{s}_i)\}_{i=1}^{L}$

Orthogonal matching pursuit (OMP) (Pati et al., 1993) offers a theoretically better solution because it does not rely on coefficients estimated early on, which could have been biased. Thus, the optimal coefficients are re-estimated at each iteration. All that needs to be stored at each iteration are the selected column indices, $p_1, p_2, \ldots, p_L$, as is detailed in Algorithm 3-2.

[3] Alternatively, the constraint on matching pursuit can be posed as threshold on the residual error. However, choosing this threshold depends on the dynamic range of the signal and noise, and in practice it is simpler to predefine L.

Algorithm 3-2. Orthogonal matching pursuit (OMP).

Input: $\mathbf{x}, \{a_p\}_{p=1}^{P}, L$
  $\hat{\mathbf{x}} = 0$
  For $i = 1, \ldots, L$ do
    $\mathbf{r} \leftarrow \mathbf{x} - \hat{\mathbf{x}}$
    $c_q \leftarrow \frac{1}{\|\mathbf{a}_q\|}\langle \mathbf{a}_q, r \rangle \quad q = 1, \ldots, P$
    $p_i \leftarrow \arg\max_q = |c_q|$
    $\tilde{A} \leftarrow \left[\tilde{A}, \mathbf{a}_{p_i}\right]$
    $\tilde{\mathbf{s}} \leftarrow \tilde{A}^{\dagger}\mathbf{x}$
    $\hat{\mathbf{x}} \leftarrow A\tilde{\mathbf{s}}$
  End for
Output: $\{(p_i, \tilde{s}_i)\}_{i=1}^{L}$

Returning to the time-series case, consider the following least-squares problem

$$\min_{\{(p_i,\alpha_i,\tau_i)\}_{i=1}^{L}} \int_{-\infty}^{\infty} \left(x(t) - \sum_{i=1}^{L} \alpha_i \int_{-\infty}^{\infty} \delta(t-\tau_i) a_{p_i}(u) du \right)^2 dt. \tag{3–20}$$

The matching pursuit approach for time series is a greedy solution to this problem. At each iteration the solution is chosen as if only one source is active at only one point in time, and selects a single atom, consisting of the timing, amplitude, and waveform, that explains the most energy remaining in the residual of the signal. This criterion is equivalent to finding the filter with the highest normalized cross-correlation. At the end of each iteration, the residual signal is updated by removing the single-atom reconstruction. This updated residual is used as the input to the next iteration. The steps are detailed in Algorithm 3-3.

Given the atomic decomposition $\{(p_i, \alpha_i, \tau_i)\}_{i=1}^{L}$, either the sources (deconvolution) or the individual components (demixing) can be computed easily via Equation (3–4) or Equation (3–3), respectively.

In the naive implementation, matching pursuit requires the cross-correlation between each filter and the signal to be computed for each iteration. Using the fast Fourier transform on a N-length discretely sampled signal, the computational complexity of time-series MP is $\mathcal{O}(LPN \log N)$. As an approximation, we propose to perform only

Algorithm 3-3. Time-series matching pursuit

Input: $x(t), \{a_p(t)\}_{p=1}^{P}, L$
  $r(t) \leftarrow x(t)$
  For $i = 1, \ldots, L$ do
    $c_q(t) = \frac{1}{\int_{-\infty}^{\infty} a_q^2(u)du}(a_q \star r)(t) \quad q = 1, \ldots, P$
    $p_i \leftarrow \arg\max_q \max_t |c_q(t)|$
    $\tau_i \leftarrow \arg\max_t |c_{p_i}(t)|$
    $\alpha_i \leftarrow c_{p_i}(\tau_i)$
    $r(t) \leftarrow r(t) - \int_{-\infty}^{\infty} \alpha_i \delta(t - \tau_i - u) a_{p_i}(u)du$
  End for
Output: $\{(p_i, \alpha_i, \tau_i)\}_{i=1}^{L}$

a single cross-correlation for each filter, and extract timings for each source excitation that do not overlap with themselves (different filters are allowed to overlap). Given the timings and filter indices, a matrix of individual excitation waveforms is constructed: $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_L]$ where $v_i(t) = (\delta_{\tau_i} * a_{p_i})(t)$. Then the amplitudes are then solved using a least-squares fit, $[\alpha_1, \alpha_2, \ldots, \alpha_L]^\mathsf{T} = V^\dagger \mathbf{x}$. The steps (for a filter length with length C) are outlined in Algorithm 3-4. To allow the filters to overlap this approximation can

Algorithm 3-4. Approximate time-series matching pursuit.

Input: $x(t), \{a_p(t)\}_{p=1}^{P}, C$
  $z_q(t) \leftarrow 1 \quad \forall q, t$
  $i \leftarrow 1$
  While $i < L$ and $\exists q, t : z_q(t) = 1$ do
    $i \leftarrow i + 1$
    $c_q(t) = \frac{1}{\int_{-\infty}^{\infty} a_q^2(u)du}(a_q \star x)(t) \quad q = 1, \ldots, P$
    $p_i \leftarrow \arg\max_q \max_t |c_q(t) z_q(t)|$
    $\tau_i \leftarrow \arg\max_t |c_{p_i}(t) z_{p_i}(t)|$
    $z_{p_i}(t) \leftarrow 0 \quad t \in (\tau_i - C/2, \tau_i + C/2)$
    $\mathbf{v}_i \leftarrow \delta_{\tau_i} * \mathbf{a}_{p_i}$
  End while
  $L \leftarrow i$
  $[\alpha_1, \alpha_2, \ldots, \alpha_L]^\mathsf{T} \leftarrow [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_L]^\dagger \mathbf{x}$
Output: $\{(p_i, \alpha_1, \tau_i)\}_{i=1}^{L}$

be run multiple times in sequence, using the remaining residual from the previous run. Although this approach requires only P cross-correlation computations per run, the

timings may not be optimal, since they are computed without considering the other filters. Additionally, the size of V may be quite large, requiring extensive computation for the least-squares solution. The computational complexity is $\mathcal{O}\left(L^3\right)$. This is better than standard matching pursuit if $L^2 < PN \log N$.

3.4 System Identification

Assuming some estimates of the sources have been made, identifying the filters is a system identification problem. Assuming the filters are all the same length, there are PM coefficients to be estimated from the N observations in $\mathbf{x}$.

In the sparse setting, the system identification problem poses a unique statistical challenge. At extreme sparsity, the filters do not overlap, which makes estimation easier, but since they appear very rarely any noise can cause large variance in the estimation of the filters. As the rate of the sources increases, there are more realizations of the sources but they overlap with each other more frequently.

Assuming the sources are stationary, the Wiener filter provides the optimal system identification in a least-squares sense. The source estimates are the inputs and the observed signal $x(t)$ is treated as the desired signal.

The simplest solution comes from the extreme sparsity case, wherein the sources are independent Poisson processes of very low rate. In this case the autocorrelation matrix of the input can be approximated as a diagonal matrix. Ignoring the scale of each filter (since it can never be determined unambiguously) the filter can be estimated solely from the cross-correlation

$$\hat{a}_p(\tau) = (s_p \star x)(\tau - 1) = \sum_t s_p(t) x(t + \tau - 1). \tag{3–21}$$

If the sources are correlated in time or space, then ignoring the correlations can lead to a biased estimate of the filters. Indeed by analysis, it easily be shown that the weighted average estimate is a biased version of the Wiener filter. The bias decreases as the

correlation matrix of the sources approaches a diagonal matrix. However, both the Wiener filter and weighted average assume the sources are provided and noise-free.

3.5 Methods for Blind MISO System Identification

In the blind setting, the values of the sources are unknown and cannot be used for system identification. In addition, the filters associated with the sources $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_P$ are also unknown and need to be estimated before any form of deconvolution can provide estimates of the sources. Thus, an unsupervised approach is needed—one that can pull itself up by the bootstraps—to estimate both the unknown sources and the unknown filters.

Estimating the filters corresponds to finding one row of each block of $\bar{A}$ corresponding to each unique filter. In the matrix formulation (3–12), it is clear that the inner dimension of the matrix multiplication $X^\mathsf{T} = \bar{A}\bar{S}$ is greater the number of rows of X, even for one source. Thus using a low-rank approximation, i.e., principal component analysis, will fail to yield a useful channel estimation. Instead, the knowledge that the sources are sparse should be used to guide the estimation of the filters.

When using the matrix notation, finding projections that result in sparse coefficients is equivalent to identifying the filters that form the basis of the signal. One way to identity sparsity is by evaluating the higher-order moments of a random variable; sparse random variable have very different higher-order moments than than those of Gaussian random variables.

As an alternative to the statistics of the sources, sufficiently sparse sources ensure that any observation window can be approximated by a small number of the elements. The optimization problem of minimizing the reconstruction error of the signal using only a few activations of the filter can be used to estimate the underlying filters (Jost et al., 2006; Mailhé et al., 2008). In the rest of the section, these two different estimation paradigms are introduced.

3.5.1 Independent Component Analysis

Using independent component analysis to solve the blind deconvolution problem is motivated by the following reasoning: first, linear filtering induces temporal dependence in the resulting signal; consequently, a deconvolution filter that minimizes this dependence should produce the original signal—up to some indeterminacies. This is especially the case when the source has independent, identically distributed entries. As long as the source is not Gaussian, then higher-order statistics can be used to estimate the dependence; whereas, if the source is Gaussian, then the deconvolution filter can at best whiten the observed signal.

The deconvolution filter is a demixing vector applied to a delay line of the signal. If it exists, the optimal deconvolution filter is the filter inverse or a lagged version of it. Unlike the standard case for ICA, when there is single vector for each source, in the time-series case there are many solutions to the single-channel deconvolution problem corresponding to different shifts of the demixing vector. Each shift corresponds to a minimum of the dependence measure (Shalvi & Weinstein, 1990) and a solution to the deconvolution problem. At which shifts it appears is an ambiguity in most blind deconvolution algorithms.

3.5.1.1 FastICA

In the FastICA algorithm (Hyvarinen, 1999), dependence is evaluated as an approximation of negenetropy (Novey & Adali, 2008). Negentropy gauges the non-Gaussianity of a random variable in terms of its higher-order statistics. Since a Gaussian random variable only has second-order statistics these differences can be quantified by finding the expectation of a non-quadratic function such that the moment expansion will contain higher-order terms. The difference between these higher-order and those of a Gaussian are used to assess the non-Gaussianity.

Approximately, the negentropy of the random variable u is proportional to $E\{G(u)\} - E\{G(\nu)\}$ where $G(\cdot)$ is the contrast function and ν is a Gaussian random variable

with the same mean and covariance as u. For random vectors, maximizing the sum of the negentropies, under the constraint of decorrelation, minimizes the mutual information between the elements. This is basic principle of contrast-function-based ICA approaches.

Recall the signal $\mathbf{x}$ is stored in the $M \times N$ matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ where N corresponds to the number of samples and M the size of the inverse filter, and X^T is Toeplitz. Let $\mathbf{x}$ denote a random vector corresponding to a patch of the signal, then $E\{f(\mathbf{x})\}$, the expected value of a function of this random vector, is estimated as $\frac{1}{N}\sum_t f(\mathbf{x}_t)$, i.e, the columns of X are treated as realizations of this random vector.

ICA algorithms typically require the input signals to be first uncorrelated, before evaluating any further dependence. To enforce this constraint, the estimation is usually performed with whitened data. Eigendecompositions are used to whiten the data. The eigendecomposition of the covariance matrix is $E\{\mathbf{x}\mathbf{x}^\mathsf{T}\} = \frac{1}{T}XX^\mathsf{T} = U\Sigma U^\mathsf{T}$. Then $\Psi = \Sigma^{-1/2}U^\mathsf{T}$ is the whitening matrix such that $E\{(\Psi\mathbf{x})(\Psi\mathbf{x})^\mathsf{T}\} = \frac{1}{N}(\Psi X)(\Psi X)^\mathsf{T} = \mathsf{I}$, where I is the identity matrix. After whitening, FastICA assumes the demixing vectors are orthogonal.

Single-unit ICA uses one demixing vector, $\mathbf{w}$, to estimate a single source, $\hat{s} = \mathbf{w}^\mathsf{T}\mathbf{x}$. With a sign and magnitude ambiguity in ICA, the source is constrained to have unit variance. This constraint is typically written as $E\{(\mathbf{w}^\mathsf{T}\mathbf{x})^2\} = \frac{1}{N}\mathbf{w}^\mathsf{T}XX^\mathsf{T}\mathbf{w} = 1$, but can be written in terms of the whitening matrix as $\|\mathbf{w}^\mathsf{T}\Psi^{-1}\|_2 = 1$ (using $\frac{1}{N}XX^\mathsf{T} = \Psi^{-1}\Psi^{-\mathsf{T}}$). The optimal $\mathbf{w}$ is one such that $\mathbf{w}^\mathsf{T}\mathbf{x}$ has higher-order statistics far from those of a Gaussian random variable of equal mean and variance.

The optimization problem for single-unit FastICA can be written as

$$\underset{\|\mathbf{w}^\mathsf{T}\Psi^{-1}\|_2=1}{\arg\max} \left[E\left\{G(\mathbf{w}^\mathsf{T}\mathbf{x})\right\} - E\left\{G(\nu)\right\}\right]^2, \tag{3–22}$$

where $G(\cdot)$ is a suitably chosen contrast function. For sparse sources, the contrast function is typically a symmetric sub-quadratic function such as $G(u) = \log\cosh(u)$, which has the $\tanh(u)$ as its first derivative.

While the vector $\mathbf{w}$ corresponds to a row of the demixing matrix, it also corresponds to the inverse of the filter such that $\mathbf{w}^\mathsf{T} X = \hat{\mathbf{s}}_l^\mathsf{T}$ is an estimate of the source signal as in Equation (3–13). The corresponding column of the mixing matrix is denoted $\mathbf{v}$ and is related by $\mathbf{v} = \frac{1}{N} X X^\mathsf{T} \mathbf{w} = E\{\mathbf{x}\hat{s}\}$. Clearly, $\mathbf{v}$ is nothing more than the weighted average of windows of the signal where the weighting assigned to each vector corresponds to the estimated source value.

Theoretically, the mixing vector corresponding to the correct demixing vector $\mathbf{w}$ should be equal to one of the rows of $\mathcal{V}$ corresponding to a time-reversed and shifted version of the R element FIR filter $\mathbf{a}$, i.e., $\mathbf{v} = [0, \cdots, 0, a(R), \ldots, a(1), 0, \ldots, 0]^\mathsf{T}$. Note that in terms of filters, $\mathbf{w}$ is not computed as the filter inverse as it was using the Toeplitz form of $\mathbf{a}$ as in Equation (3–10).

In order to find a good demixing vector, an approximate solution to the single-unit ICA problem (3–22) yields an iterative update for $\mathbf{w}$ (Hyvarinen, 1999):

$$\mathbf{w} \leftarrow \Psi^\mathsf{T} \Psi E\left\{\mathbf{x} g\left(\mathbf{w}^\mathsf{T}\mathbf{x}\right)\right\} - E\left\{g'\left(\mathbf{w}^\mathsf{T}\mathbf{x}\right)\right\}\mathbf{w}, \tag{3–23}$$

$$\mathbf{w} \leftarrow \frac{\mathbf{w}}{\sqrt{E\left\{\mathbf{w}^\mathsf{T}\mathbf{x}\mathbf{x}^\mathsf{T}\mathbf{w}\right\}}}, \tag{3–24}$$

$g(u) = \frac{\partial}{\partial u} G(u)$, and $g'(u) = \frac{\partial}{\partial u} g(u)$. For $G(u) = \log\cosh(u)$, $g(u) = \tanh(u)$ is a soft activation function, and $g'(u) = 1 - \tanh^2(u)$ is a symmetric function that peaks at 0. In terms of the mixing vector this update is

$$\mathbf{v} \leftarrow E\left\{\mathbf{x} g\left(\hat{s}\right) - \mathbf{v} g'(\hat{s})\right\} \tag{3–25}$$

where $\mathbf{w}^\mathsf{T} = \mathbf{v}^\mathsf{T}\Psi^\mathsf{T}\Psi$ and $\hat{s} = \mathbf{w}^\mathsf{T}\mathbf{x}$. This matches the interpretation of $\mathbf{v}$ as a weighted average, but gives an interpretation on its update. For a given realization, a large source magnitude $|\hat{s}| > 1$ implies $g(\hat{s}) \approx \mathrm{sign}(\hat{s})$ and $g(\hat{s}) \approx 0$, this yields $\mathbf{v} \approx \mathbf{x}$. When the

source coefficient is smaller such that $g'(\hat{s})$ dominates $g(\hat{s})$, the update moves the vector away from its current direction. Thus, single-unit FastICA uses a weighted average to estimate the filter, wherein the source estimates correspond to source that is maximally non-Gaussian.

For the single-channel case there appears to be no need to estimate the full demixing matrix. In practice, using a multiple-unit approach, with a constraint that the sources are uncorrelated, can perform better than the single-unit approach. The multiple-unit approach, which maximizes the sum of negentropies, estimates multiple demixing vectors at once. Let the demixing matrix with K columns be denoted $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]$. The multiple-unit optimization problem is

$$\underset{E\left\{(W^{\mathsf{T}}\mathrm{x})(W^{\mathsf{T}}\mathrm{x})^{\mathsf{T}}\right\}=\mathrm{I}}{\arg\max} \sum_{k=1}^{K} \left[E\left\{G\left(\mathbf{w}_k^{\mathsf{T}}\mathbf{x}\right)\right\} - E\left\{G(\nu)\right\}\right]^2 \tag{3–26}$$

where I is the identify matrix.

In the time-series setting, the true mixing matrix is overcomplete, and the optimal vector for the single-unit case (3–22) will not necessarily correspond to a column of the matrix obtained for the multi-unit case (3–26). When using the multiple-unit approach for a single-source problem, a single vector should be selected to represent the filter. As ICA does not give a natural order for 'better' components, we resort to using the mean squared error as a cost function and choosing the filter that has the best reconstruction of the signal when using matching pursuit to approximate the original signal.

3.5.1.2 Multiple source case

In the multiple source case, an independent component analysis must resolve multiple demixing vectors simultaneously. The projections that yield sparse values correspond to demixing vectors for each source at each permissible lag, as in Equation (3–13). Estimating a set of demixing vectors at once, to extract all the sources, greatly increases the dimensionality of the solution space. Since each source may have arbitrary lag, there is a even greater number of solutions that maximize the measure of independence.

This can lead to the redundant estimation of the same filter at different lags. The constraint on correlation that avoids degenerate solutions in instantaneous ICA is inadequate for the convolutive case as different lags of the same source may be uncorrelated. For instance example, if the length of the demixing vectors is longer than the filter extent, the 'independence' between filters can be found by simply lagging the filters such that they do not overlap. Consequently, sets of demixing vectors often include short filters at multiple shifts.

The independent component analysis problem for multisource convolutive mixtures possesses a very complicated performance surface. An estimation can yield a set of redundant shifts of a few unique filters and some spurious ones. The spurious filters are often noisy and may correspond to local optima caused by the correlation constraints (3–26). This does not prevent the estimation of some subset of the filters. After running a multi-unit ICA, there is a need to select a subset of the unique and 'useful' filters.

3.5.1.3 Filter subset selection

In the blind estimation setting, it is unknown how many true sources there are. In practice, ICA can estimate as many filters as there are taps in the embedding window. However, as mentioned this may yield many spurious filters that need to be excluded by a relevance criterion.

There is not a clear criterion for filter selection based on independence. It is easier for the user to provide how many filters are desired and use the reconstruction error to select a subset of the filters. Given the desired number of filters, the goal is to find the optimal subset that minimizes the reconstruction cost. However, this problem has a combinatorial number of solutions. In practice, a greedy optimization such as orthogonal matching pursuit (OMP) (Pati et al., 1993), Algorithm 3-2, can be used to find which set of filters best explains the signal.

The primary aim is to find the filters that are most correlated with the data, but are not redundant copies of each other. To do this, an approximation of the signal

is made with each individual filter using time-series MP (Algorithm 3-3). Then these components are treated as the basis vectors, and OMP is used to select a subset of these components to approximate the signal. This is a simpler optimization since only the combination of components is optimized: the timing of the sources having already been estimated. The steps are detailed in Algorithm 3-5.

Algorithm 3-5. OMP Select

Input: $\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^{K}, L$
  For $k = 1, \ldots, K$ do
    $\{(\alpha_i, \tau_i)\}_{i=1}^{L} \leftarrow \text{MP}(\mathbf{x}, \mathbf{v}_k, L)$
    $\mathbf{y}_k \leftarrow \sum_{i=1}^{L} \alpha_i T_{\tau_i} \mathbf{v}_k$
  End for
  $\{q_i\}_{i=1}^{P} \leftarrow \text{OMP}\left(\mathbf{x}, \{\mathbf{y}_k\}_{k=1}^{K}, P\right)$
  $\tilde{\mathbf{v}}_i \leftarrow \mathbf{v}_{p_i} \quad \forall i = 1, \ldots, P$
Output: $\{\tilde{\mathbf{v}}_p\}_{p=1}^{P}$

Using the greedy approach it is unlikely that two filters that are simply shifted versions of each other will be selected, since their components will be approximately the same. In addition, the greedy approach does not require performing MP using all of the filters estimated by ICA at once. After selection, the remaining filters corresponding to the selected components are used for performing a full atomic decomposition of the signal.

3.5.2 Matching Pursuit with K-SVD

Assuming the signal plus noise model model in Equation (3–6), the blind deconvolution and system identification problem can be posed as a non-linear least-squares problem. The overall objective function can be written as

$$\min_{\{\mathbf{v}_p\}_{p=1}^{P}, \{(p_i, \alpha_i, \tau_i)\}_{i=1}^{L}} \sum_{t=1}^{N} \left(x(t) - \sum_{i=1}^{L} \alpha_i v_{p_i}(\tau_i + M - t) \right)^2 \tag{3–27}$$

where all the sources are assumed to be active exactly L times and $\mathbf{v}_p$ is an estimate of the time-reversed filter $\mathbf{a}_p$, i.e., $v_p(t) = \hat{a}_p(M + 1 - t)$. Because the source estimates,

i.e., the atomic decomposition, are intrinsically linked to the filters, it is necessary to perform an alternating optimization. Mailhé et al. (2008) proposed an extension of K-SVD (Aharon et al., 2006) to the time-series case by using an alternating optimization between matching pursuit and K-SVD. For conciseness, we call the combined algorithm MP-SVD.

For clarity, we introduce the same notation as Mailhé et al. (2008). Let T_τ denote the linear operator such that $T_\tau \mathbf{v}$ aligns the M-length filter $\mathbf{v}$ within a N length signal. T_τ is a $N \times M$ matrix with the $M \times M$ identity matrix as a submatrix starting at row τ. The transpose of this operator is denoted T_τ^* and is defined such that $T_\tau^* \mathbf{x}$ extracts the M-length window from $\mathbf{x}$ starting at time τ. Using the alignment operator T_τ the objective function can be written in terms of vectors as

$$\min_{\{\mathbf{v}_p\}_{p=1}^P, \{(p_i, \alpha_i, \tau_i)\}_{i=1}^L} \left\| \mathbf{x} - \sum_{i=1}^{L} \alpha_i T_{\tau_i} \mathbf{v}_{p_i} \right\|_2^2 . \tag{3–28}$$

To update the filters, we first assume we have an estimate of the components using the current filters. Let $\mathbf{x}^{(p)}$ denote the signal consisting only of the pth component and any error

$$\mathbf{x}^{(p)} = \mathbf{e} + \mathbf{y}_p = \mathbf{x} - \sum_{q \in \{1,\dots,P\} \setminus p} \mathbf{y}_q \tag{3–29}$$

where $\mathbf{y}_p = \sum_{j \in \mathcal{I}_p} \alpha_j T_{\tau_j} \mathbf{v}_p$ and $\mathcal{I}_p = \{i : p_i = p\}$. This single-filter signal is used to update each filter, but because of the sparsity in time, only patches of the signal corresponding to source timings are used. These patches are collected into a matrix. The updated filter is selected as the singular vector of this matrix corresponding to the largest singular value. Assuming these were the correct timings, and none of the other filters changed, this update minimizes the reconstruction cost for these patches. All of the steps are detailed in Algorithm 3-6. In the rest of the work we use this algorithm with the non-overlapping approximation of time-series matching pursuit. In addition, we explore two alternative approximations to ameliorate the computational complexity.

Algorithm 3-6. MP-SVD

$\mathbf{v}_p \leftarrow \text{randn}(M, 1) \quad \forall p = 1, \ldots, P$
Repeat
$\{(p_i, \alpha_i, \tau_i)\}_{i=1}^{L} \leftarrow \text{MP}\left(\mathbf{x}, \{\mathbf{v}_p\}_{p=1}^{P}, L\right)$
$\mathbf{e} \leftarrow \mathbf{x} - \sum_{i=1}^{L} \alpha_i T_{\tau_i} \mathbf{v}_{p_i}$
For $p = 1, \ldots, P$ do
$\mathcal{I}_p \leftarrow \{i : p_i = p\}$
$\mathbf{x}^{(p)} \leftarrow \mathbf{e} + \sum_{j \in \mathcal{I}_p} \alpha_j T_{\tau_j} \mathbf{v}_p$
$\breve{X}_p \leftarrow \left[T^*_{\tau_j} \mathbf{x}^{(p)}\right]_{j \in \mathcal{I}_p}$
$\mathbf{v}_p \leftarrow \arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top \breve{X}_p \breve{X}_p^\top \mathbf{w}}{\mathbf{w}^\top \mathbf{w}}$
End for
Until convergence of the filters or other stopping criterion is met

3.5.2.1 Block-based approximation

MoTIF (Jost et al., 2006) is a block-based approximation to running MP on the full signal. The input signal is partitioned into non-overlapping blocks, and the best alignment of the filter is restricted to each block. As only a single alignment is identified in each patch the method works well for sparse sources that are only active at most once per patch. For a filter of length M the length of a patch Q should be chosen such that $Q \geq 2M - 1$.

The unconstrained optimization problem for the single-unit MoTIF can be written as

$$\max_{\|\mathbf{v}\|_2 = 1} \sum_{n=1}^{\lfloor N/Q \rfloor} \max_{\tau \in P_n} \left(\mathbf{v}^\top \mathbf{x}_\tau\right)^2 \tag{3–30}$$

where $P_n = \{1 + Q(n-1), \ldots, Qn\}$, $n = 1, \ldots, \lfloor N/Q \rfloor$. Given the best alignment of the filter within each patch, the objective is to minimize the reconstruction cost. Essentially, the block-based approximation for MP-SVD is solved by alternating between finding the windows' alignments within the blocks based on the current filter, and updating the filter using SVD. Computationally, this is a fast proxy since it finds N occurrences in parallel. In the single filter case, it performs very well. In the multiple filter case, Jost et al. (2006)

propose to use a generalized eigenvalue problem to ensure that the correlation in the filters is minimized. Alternatively, the block-based approximation can also be used with MP-SVD. In either case, the method essentially assumes each filter is active exactly once in each block. This is a very strong assumption. We found that it yields extremely poor results for multiple filter estimation except in the case of extreme sparsity.

3.5.2.2 Greedy approach

A greedy approach for blind system identification is to learn a single filter using MP-SVD and then remove a set number of excitations of this filter from the input signal. The remaining residual is used as the input to estimate the next filter, and this process continues until the desired number of filters are estimated. After learning the filters, this same greedy approach can be used for approximation as it avoids the joint search over both the best filter and best lag. However, the ability of this approach to learn the true source filters is highly limited: on the first iteration it is more likely to learn a filter that best approximates the signal on average, rather than any specific filter.

Algorithm 3-7. Greedy MP-SVD

Input: $\mathbf{x}$, L

$\mathbf{r} \leftarrow \mathbf{x}$

For $p = 1, \ldots, P$ do

$\mathbf{v}_p \leftarrow \text{randn}(M, 1)$

Repeat

$\{(\alpha_i, \tau_i)\}_{i=1}^{L} \leftarrow \text{MP}(\mathbf{r}, \mathbf{v}_p, L)$

$\breve{R} \leftarrow \left[T_{\tau_i}^{*}\mathbf{r}\right]_{i=1}^{L}$

$\mathbf{v}_p \leftarrow \arg\max_{\mathbf{w}} \dfrac{\mathbf{w}^{\mathsf{T}}\breve{R}\breve{R}^{\mathsf{T}}\mathbf{w}}{\mathbf{w}^{\mathsf{T}}\mathbf{w}}$

Until convergence of the filter or other stopping criterion is met

$\{(\alpha_i, \tau_i)\}_{i=1}^{L} \leftarrow \text{MP}(\mathbf{r}, \mathbf{v}_p, L)$

$\mathbf{r} \leftarrow \mathbf{r} - \sum_{i=1}^{L} \alpha_i T_{\tau_i}\mathbf{v}_p$

End for

Output: $\{\mathbf{v}_p\}_{p=1}^{P}$, L

3.6 Synthetic Experiments

Experiments are conducted to compare the performance of algorithms for both for single-source filter estimation, corresponding to a SISO system, and multiple source filter estimation, corresponding to a MISO system. In both cases the true underlying filters are chosen as a set of Daubechies 4 (db4) wavelet packets. This ensures the synthetic filters cover a range of frequency space with varying temporal characteristics. The goal of the experiment is to compare two basic methodologies for system identification: shift-invariant decomposition with mean-square error cost, and single-channel independent component analysis.

The source signals are independent, marked point processes with a homogeneous Poisson point process for the timing and a bimodal excitation amplitude distribution. The excitation amplitude distribution is a mixture of a Gaussian distributions with mean and variance of $(1, \frac{1}{9})$ and $(-1, \frac{1}{9})$ and equiprobable Bernoulli mixing. The shape of the resulting source distribution is shown in Figure 3-1. In the experiments, the rate of the Poisson process controls the sparsity of the sources.

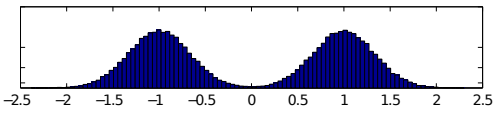


Figure 3-1. Source-excitation amplitude distribution, i.e., the amplitude distribution when a source is active.

For each run, a signal with 10,000 time points was created by convolving the source realizations with the selected filters and combining the results. The wavelet packets were chosen at a scale such that they were 226 elements long, and each filter was windowed to be 180 time points long. To create a signal with reasonable signal-to-noise ratio (SNR), the filters were scaled to have a norm of 180/2, and zero-mean, unit-variance white noise was added.

For the synthesis-based approach, the filters are learned using matching pursuit (MP) for source estimation, and K-SVD on the aligned windows for filter estimation

(Algorithm 3-6). Two different computationally tractable approximations are used to speed up the matching pursuit: the first is the matching pursuit with no overlap (Algorithm 3-4), the second consists of block-based approximation of matching pursuit, e.g. MoTif (Jost et al., 2006). For conciseness, these are referred to as MP(no overlap)-SVD and MP(block)-SVD. In the multiple input case, the greedy one-source-at-a-time approach is also applied (Algorithm 3-7). The greedy method uses the no-overlap method for a single run of MP for each iteration update of the particular filter.

For independent component analysis approach, both FastICA with 40-unit symmetric estimation and single-unit FastICA were used. The $tanh(\cdot)$ activation function (non-linearity) is used for both instances of FastICA. In practice, most of the filters in the 40-unit estimation are meaningless so the OMP-based filter selection (Algorithm 3-5) is used to select a predefined number of filters.

3.6.1 Single Source, Blind System Identification

For a single source, the following source excitation rates were tested: 50%, 30%, 25%, 20%, 15%, 7%, 5%, 3%, 2%, 1%, 0.5%, and 0.1%. At a rate of 0.55% each filter is, on average, excited every 180 samples, the same as its length. This rate turns out to be close to the change point in the estimation performance.

At the given signal and noise characteristics, the SNR is 36.5 dB for 50% excitation and 9.4 dB for 0.1% excitation. Thus, the signal-to-noise ratio is not a useful indicator because a signal with more overlap may prove more difficult to estimate. An example signal at a source rate of 1% and SNR of 19.4 dB is shown in Figure 3-2.

For each run, the filter estimation performance is quantified as the maximum correlation coefficient, across alignments, between the estimated filter and the true filter. This quantity is averaged across all 32 filters and the results are collected across 8 Monte Carlo generations of source and noise activity. The average computation time for the estimation is also recorded. These results are shown in Figure 3-3.

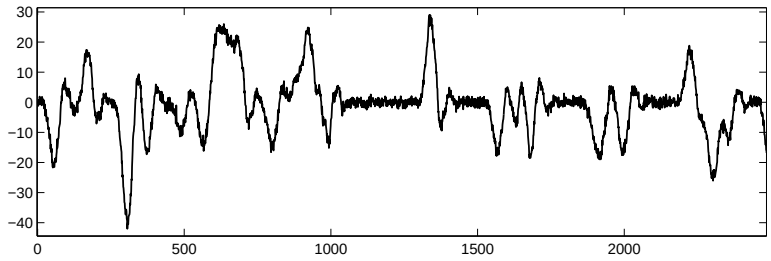


Figure 3-2. Example of a single-source signal corresponding to W1, the scaling function, of Daubechies 4 wavelet packets.

At high and low excitation rates, MP-SVD achieves the best performance. At extremely low rates $<$1% it estimates filters that match the true filters, with a correlation coefficient of nearly 1. Across sparsities, the 40-unit symmetric ICA estimation followed by the filter selection outperforms the single-unit ICA estimation. The ICA approaches exhibit a sharp increase in performance at 10%. In the range of rates between 3% and 10% multi-unit ICA outperforms the MP-SVD approach. At rates of 1% and below, the block-based MP-SVD approach performs at nearly the same level of performance with a much shorter computation time.

3.6.2 Multiple Source, Blind System Identification

A subset of 11 filters was chosen for the multiple source case. The individual source excitation rate was sequentially varied as 10%, 7%, 5%, 3%, 2%, 1%, 0.5%, 0.1%, 0.05%, and 0.025%. This corresponds to an overall rate of excitation of 110%, 77%, 55%, 33%, 22%, 11%, 5.5%, 1.1%, 0.55%, and 0.275%. For the highest rate, the system is continually excited, and the average SNR of a single component in this mixture is -10 dB! As the source excitation rates are equal, the SNR remain constant across all rates.

For each run, the filter estimation performance is quantified using two indices: the first is the average of the correlation coefficient of each estimated filter to its best-matched true filter, and the second is the percentage of the true filters estimated with a correlation coefficient greater than 0.9. The first measure penalizes spurious

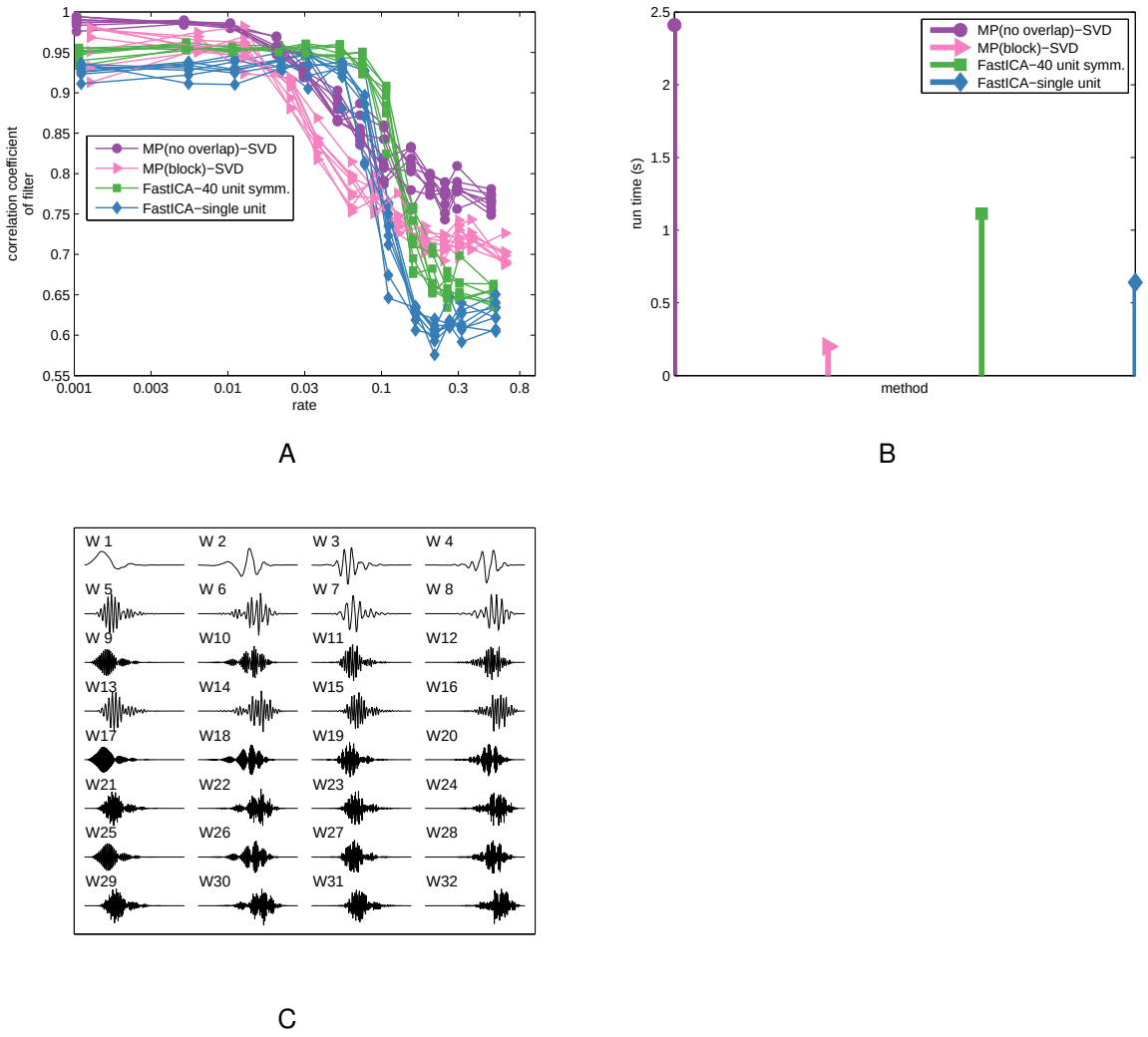


Figure 3-3. Single-source blind waveform estimation performance. A) The average correlation coefficient across the 32 waveforms for each method is recorded across 8 Monte Carlo runs. B) The average run time to estimate one waveform for each method. C) The 32 true filters, Daubechies 4 (db4) wavelet packets.

filters that are not matched to any source. The second measure does not penalize spurious filters and allows an overcomplete system identification because the maximum is taken across all estimated filters. In addition, the computation time is recorded for each method. These results are shown in Figure 3-4.

The MP-SVD approaches achieves the highest performance at select rates, with an average correlation above 0.7 for rates below 1%. The matching performance is

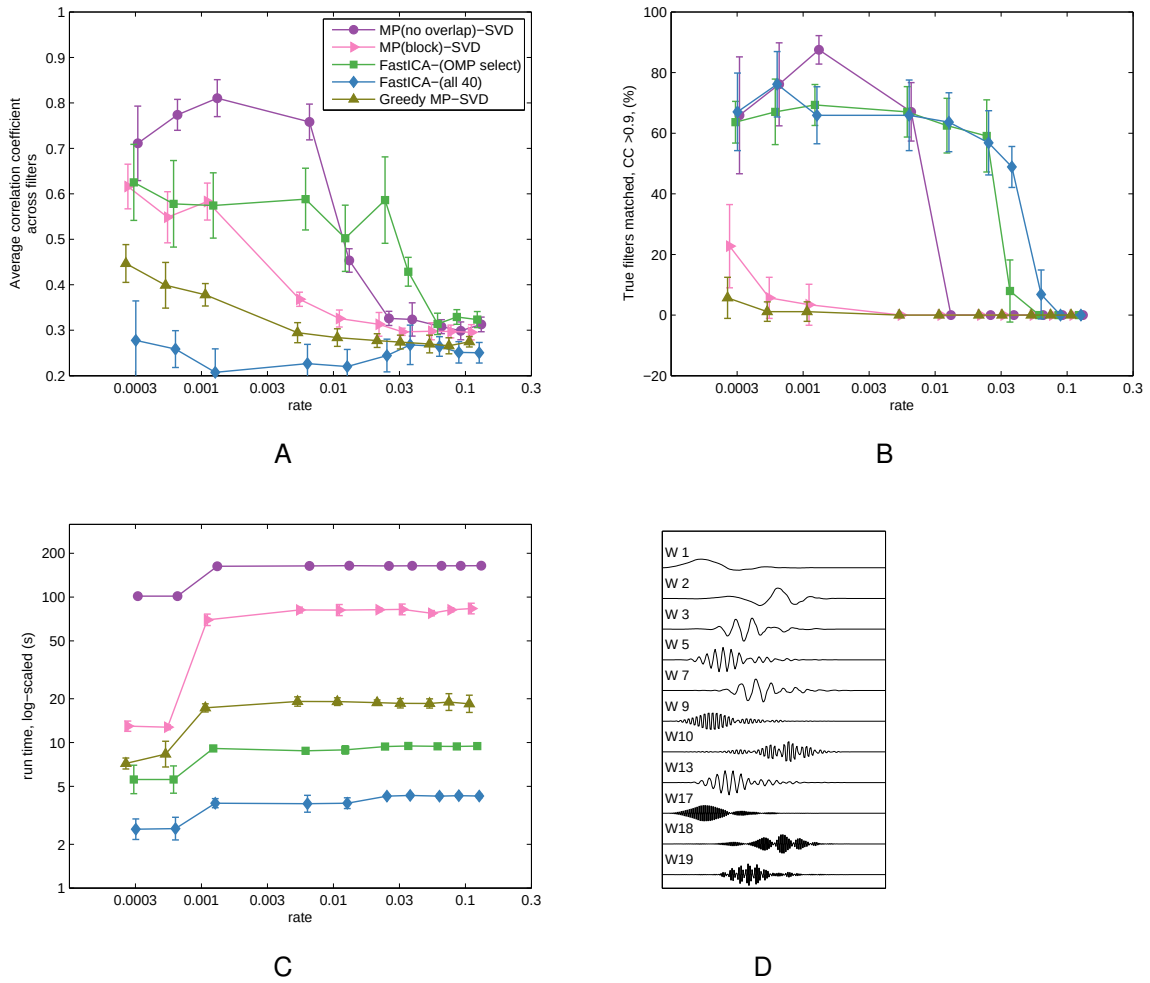


Figure 3-4. Multiple waveform, single-channel blind estimation performance. For each performance criterion, the average and standard-deviation across 8 Monte Carlo runs is shown as error-bars. A) Average correlation coefficient of estimated filters to their best-matched true filter. B) Percentage of the true filters matched, with correlation coefficient greater than 0.9. C) Run time to estimate, and if necessary, select the group of filters. D) The 11 true filters: a subset of the Daubechies 4 (db4) wavelet packets.

more sensitive to the rate, with a peak above 80%, but dropping to zero above 1%. ICA consistently matches 60% of the filters at rates below 3%. The average best-matched correlation coefficient is lower than MP-SVD at low rates, but is consistent across a range of sparsity levels. At extreme sparsity, the block-based approximation for MP-SVD is able to perform as well as ICA in terms of average correlation coefficient, but is still more computationally expensive than ICA. The greedy approximation is the fastest

MP-SVD-based approach but its performance is extremely poor. The ICA-based approach appears to have the best tradeoff in terms of speed and consistent accuracy, with a run time of nearly 100 times less than MP-SVD.

3.6.3 Discussion

For cases of sparse sources, MP-SVD is the most accurate for blind system identification for both SISO and MISO systems. One hypothesis for why the ICA approach performs better at higher rates is its avoidance of explicit modeling. The MP-SVD algorithm updates the filters based on the assumption that the other filters are perfectly represented. However, early on in the high rate case this is very unlikely, and the filter estimates become entrapped at local minima of the overall objective function.

Here we have used matching pursuit that restricts overlaps of the same filter. This approximation may be another reason that MP-SVD is not able to perform as well as ICA for higher rate sources. However, running the full matching pursuit algorithm requires much longer computation time and, furthermore, increases the convergence time. Often, the filters do not converge at high rates. This is most likely the same issue discussed in the previous paragraph, compounded by the overlap of a filter with itself. Indeed, because of the overlap the original SVD update is biased optimal and Mailhé et al. (2008) proposed an unbiased alternative.

These results confirm that both single-channel ICA and shift-invariant sparse coding are able to blindly estimate the underlying filters in a multiple-input single output system. Both approaches are limited to reasonable rates for the source excitations. At high excitation rates, the sparse model is not meaningful as it would require as many parameters as samples in the original signal.

For higher rates, continual excitation of the sources can be used to estimate auto-regressive and moving average models. These models would be able to describe the signals in terms of their time-varying spectral quantities. Thus, an analyst must choose the correct model based on a hypothesis of the underlying source excitations.

3.7 Decomposing Local Field Potentials

In this section, multichannel local field potentials recorded from the motor cortex of a bonnet macaque through a chronically implanted cortical micro-electrode array are analyzed. These signals were recorded as the subject passively observed a monitor that displayed cursor movements and colors that indicated upcoming reward delivery. The details of the task are further described in Section 4.9. Herein, decompositions of a single channel and of multiple channels are performed on 15 minutes of the signal, divided into 60 s segments. The relationship with task timings and behavior is not analyzed.

This data is from a single day's recording collected in Joseph Francis's laboratory at SUNY-Downstate by Brandi Marsh. LFPs were collected on 32 electrodes implanted in M1. On this dataset, 8 of the LFP electrodes were excluded because of artifacts, which were associated with high impedance. The remaining 24 channel signal was high-pass filtered with a cutoff of 4 Hz and notch filtered at 60, 120, 180, 240, and 300 Hz with a quality factor of 69. The data was down-sampled by a factor of 3, such that the effective sampling rate is 666.67 Hz. At this rate, each 60 s window of data corresponds to 40,001 time points.

3.7.1 Single Channel Decomposition

For preliminary investigations, the single-channel blind system identification algorithms were applied to a select signal. After learning the filters on a single 60 s segment, multiple iterations of the approximation algorithms were applied to that segment and the other 14 segments. The non-overlapping convolution-based orthogonal matching pursuit algorithm is used to obtain the atomic decomposition. The greedy single-filter approximation is performed successively with each filter in the same order that they were estimated.

The number of filters to estimate was set to 4. The learned filters, and the magnitude of their frequency response, for the MP-SVD, FastICA-based, and greedy

MP-SVD algorithms are shown in Figure 3-5. MP-SVD yields filters with wavelet-like or evoked-potential-shaped impulse responses. The frequency content of the filters is mainly below 35 Hz peaking, with one filter peaking around 10 Hz. For ICA, the first two components resemble narrow-band waveforms with frequency content between 10 and 20 Hz. The other two components appear as high-frequency ripples. The greedy MP-SVD approximation yielded another sinusoidal-like component along with increasingly high-frequency waveforms. This progression makes sense as the greedy decomposition tries to explain as much as possible with each waveform—naturally following the skewed, 1/f power-spectral density for neural potentials (Bedard et al., 2006; Pritchard, 1992).

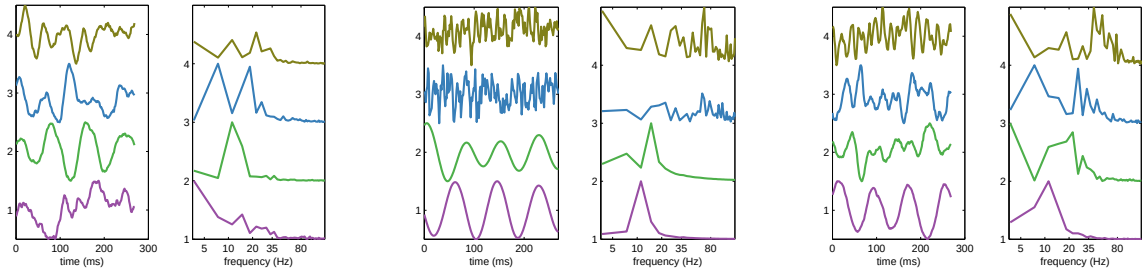


Figure 3-5. Filters estimated from the single-channel of motor cortex LFP using three algorithms: A) MP-SVD; B) FastICA; and C) Greedy MP-SVD.

The signal approximation error when using the estimated filters is calculated as proportion of variance explained (PVE). PVE is equal to one minus the normalized mean squared error. The PVE for the complete approximation and for each component individually are recorded in Table 3-1. The greedy MP-SVD approach yielded the best approximation, followed by MP-SVD, and finally ICA. For MP-SVD and ICA, the distribution of variance explained across components is nearly uniform; whereas, greedy MP-SVD has a highly skewed energy distribution with the first component explaining the majority of the variance. The computation times for both learning the filters and performing the approximation are recorded in Table 3-2.

Table 3-1. Single-channel approximation performance as proportion of variance explained.

	Training PVE (%)					Testing PVE (%)				
Method	1[a]	2	3	4	Total[b]	1	2	3	4	Total
MP-SVD	24	45	21	22	89	25±4	35±4	24±2	24±1	86±2
FastICA	37	34	16	15	84	29±4	32±1	19±1	16±1	82±1
Greedy	60	27	10	2	92	52±3	34±2	11±1	3±1	90±1

[a]Individual component. [b]Using all components at once.

Table 3-2. Computation time for single-channel filter estimation and approximation on 60 s of data sampled at 666.67 Hz.

Method	Learning time (s)	Approximation time (s)
MP-SVD	243	64±5
FastICA	12	52±3
Greedy	60	5±0.03

Note: The learning time is averaged across 2 runs. For the approximation time, the average and standard deviation are taken across 15 segments.

The temporal and spectral aspects of the approximations are shown in Figure 3-6. For the total approximation and each component, PVE is calculated at different frequencies using the Welch spectrogram with 270 ms Hamming windows and 150 ms overlap of consecutive windows. For the MP-SVD algorithm, the total approximation achieves nearly perfect reconstruction up to 25 Hz, and above 35 Hz the PVE rapidly diminishes. The first component, which has an evoked potential shape, explains most of the low-frequency response; the second component accounts for most of the variance around 15 Hz; and the fourth component is limited to 20 to 35 Hz. For ICA, the total PVE is not as uniform, but the method is able to account for over 40% of the variance between 35 and 80 Hz, corresponding to the third and fourth components. The greedy approximation has the best coverage of the spectrum, tapering off above 80 Hz. In addition, the individual components neatly tile the frequency space, with the low-frequency components being estimated first.

The extracted sources can also be visualized to understand how the atomic decomposition of the LFP segment. The atomic decomposition corresponds to waveform index, amplitude, and timing, and can be represented by a sparse time

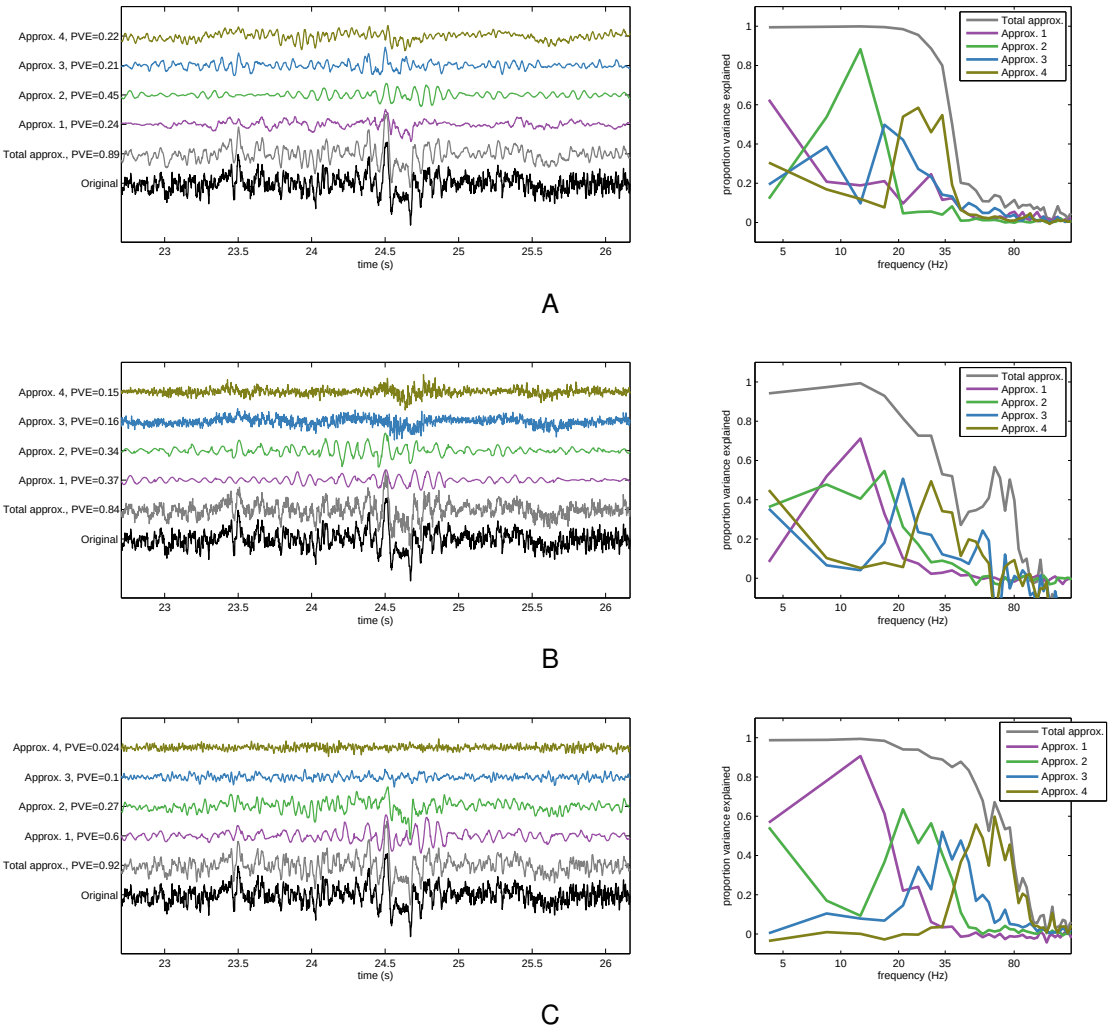


Figure 3-6. Analysis of single-channel decomposition of motor cortex LFP. The decomposition of the signal in a 3.5 s window is shown along with the approximation performance across frequency for the three algorithms: A) MP-SVD; B) FastICA; and C) Greedy MP-SVD.

series of the amplitude and timing for each component. For aid of visualization, we select only the atoms corresponding to the top 10th percentile of magnitude for each waveform and filter the sparse time series. The resulting plots are shown in Figure 3-7 along with the corresponding components.

From the source excitation images it is clear that the temporal distribution of excitation differs among the filters. The sources for filters estimated by MP-SVD appear to be the most uniform in time, those from the greedy approach are the most localized,

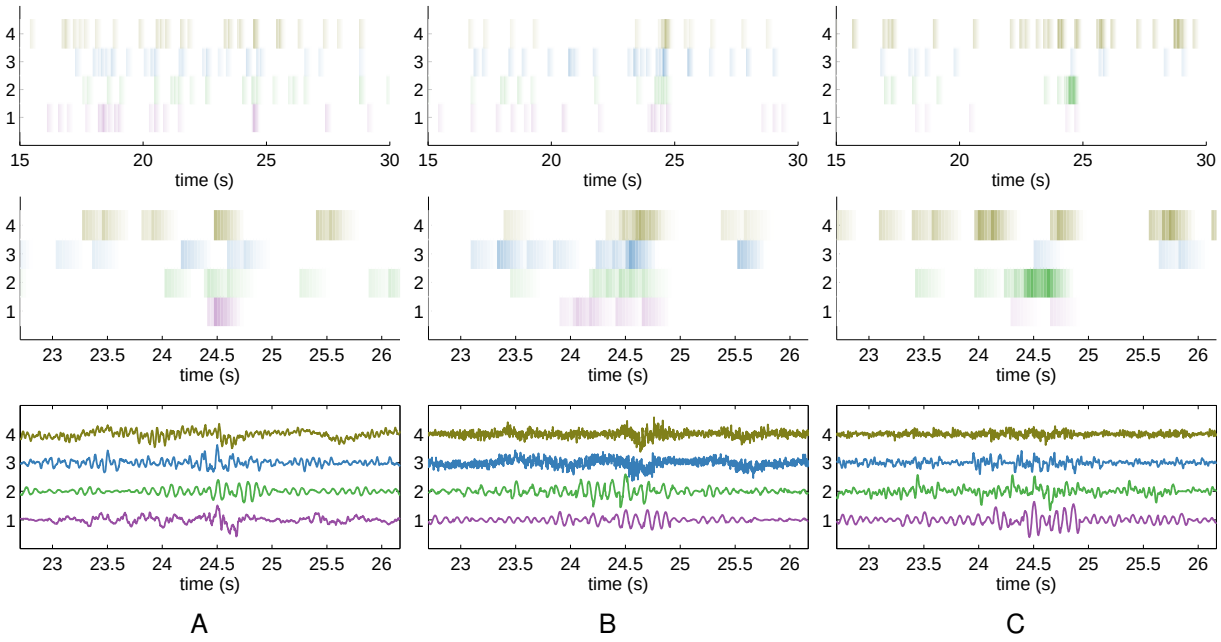


Figure 3-7. Atomic decomposition of a motor cortex LFP. Significant source activity, corresponding to the 90th percentile of magnitudes for each filter is shown in terms of color intensity, normalized to maximum amplitude across the segment. For visualization, the source activity is filtered with a linearly decreasing impulse response the length of the waveform. The images show both the onset and amplitude of the sources in two windows of the segment, component-wise approximation of the shorter window is shown in bottom row. The sources extracted using waveforms estimated by the three algorithms: A) MP-SVD; B) FastICA; and C) Greedy MP-SVD.

and those from ICA appear at more regular intervals. However, a more systematic analysis using marked point process theory is needed to understand the joint distribution of intervals and amplitudes.

3.7.2 Model Complexity

In the previous analysis, the model complexity, in terms of the number of filters and number of atoms in the decomposition, was fixed *a priori*. The approximation performance varies across both of these parameters. The number of atoms in the decomposition is determined by the number of iterations of the orthogonal matching pursuit approximation. A small experiment was conducted on the initial 60 s segment of the single-channel LFP.

The proportion of variance explained are calculated for up to 10 components for both MP-SVD and ICA. The results are displayed in Figure 3-8. For the MP-SVD, learning 5 components appears to yield nearly the best performance, above 5 components there is marginal improvement. For ICA, the performance is not necessarily monotonic function of the number of components, with 4 components appearing to perform the best across a range of atoms.

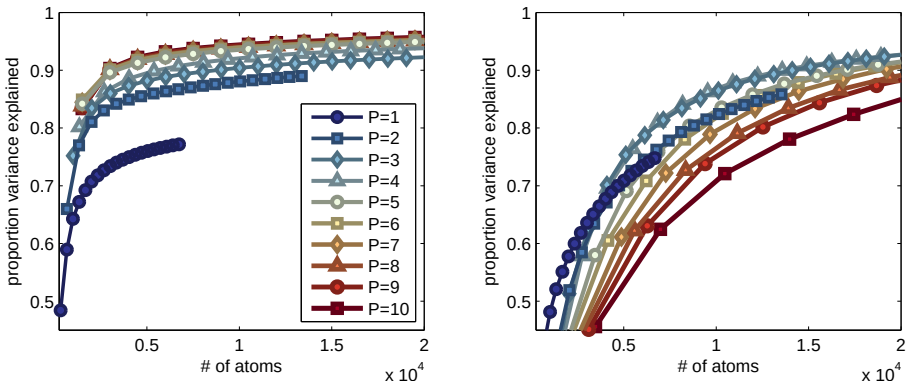


Figure 3-8. Proportion of variance explained versus number of atoms for various number of filters, denoted by P in the legend. A) Filters learned using the MP-SVD algorithm. B) Filters learned using single-channel FastICA.

In the non-overlapping approximation for matching pursuit, the number of atoms is a function of both the number of iterations and the number of bases. The performance across the number of iterations of the approximation is shown in Figure 3-9. The MP-SVD algorithm clearly outperforms ICA in terms of the variance-based measure. For the MP-SVD algorithm, above 5 components and 2 iterations there is only marginal increases in performance.

3.7.3 Multichannel Decomposition

Using PCA across space, all of the methods are extended to the multichannel case. For MP-SVD, the first principal component is estimated at each step of the learning process at the same time that SVD is performed to update the filters. For the greedy approach the same is done, but as only a single filter is updated for each greedy

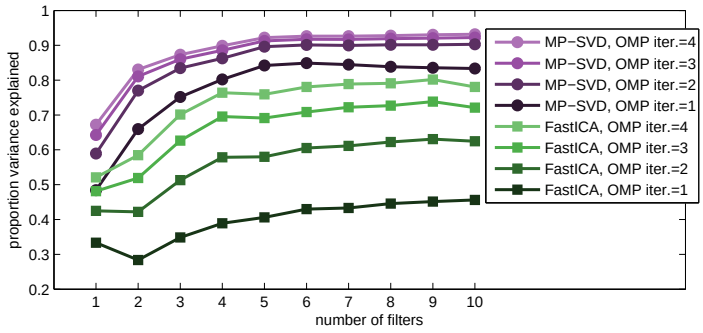


Figure 3-9. Proportion of variance explained across approximation iterations versus the number of filters.

decomposition the spatial factor is allowed to vary between components. Finally, for ICA it is difficult to combine the multiway aspect directly into the optimization. Instead, the first principal component of the raw signal is provided to single-channel ICA. This is a suboptimal solution as the other approaches are able to tune the spatial component for the temporal filters.

The filters estimated with three approaches are shown in Figure 3-10. In the multichannel case, all the filters have their peak frequency content at lower frequencies. The ICA filters appear as a mixture of the high-frequency and low-frequency oscillations, which were distinct components when applied to a raw single channel.

The fact that the components and approximations correspond to lower frequencies for the multichannel case is also apparent in the PVE across the spectrum, as shown in Figure 3-11. All of the variance explained is in the frequency range below 40 Hz; whereas in the single-channel case, ICA and the greedy MP-SVD are able to explain variance up to 80 Hz. This implies that the linear model across space is not a good model at frequencies above 40 Hz. At these frequencies, signals are unlikely to be phase-locked across the array.

The PVE across each component and across all channels are recorded in Table 3-3. Interestingly, MP-SVD performed best for the select single-channel, but greedy MP-SVD, which has a separate spatial factor for each component, was able

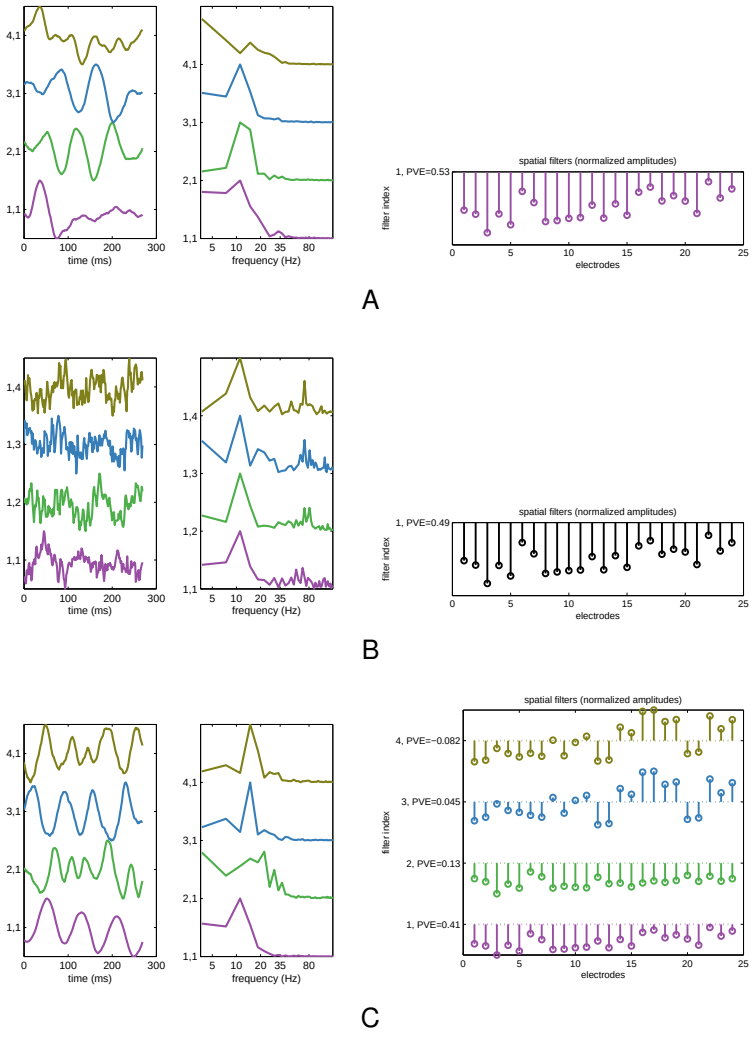


Figure 3-10. Temporal and spatial filters estimated from multi-channel LFP recording of motor cortex from the three PCA-based spatial extensions of the algorithms: A) MP-SVD; B) FastICA; and C) Greedy MP-SVD.

to account for an additional 4% of the variance. Again, MP-SVD and ICA yielded nearly equivariant components; whereas, greedy MP-SVD has a highly skewed energy distribution with the first component explaining the majority of the variance.

3.8 Summary

This study brings together approaches for blindly estimating MISO systems with sparse inputs. This study highlighted the key aspects of the problem, compared existing

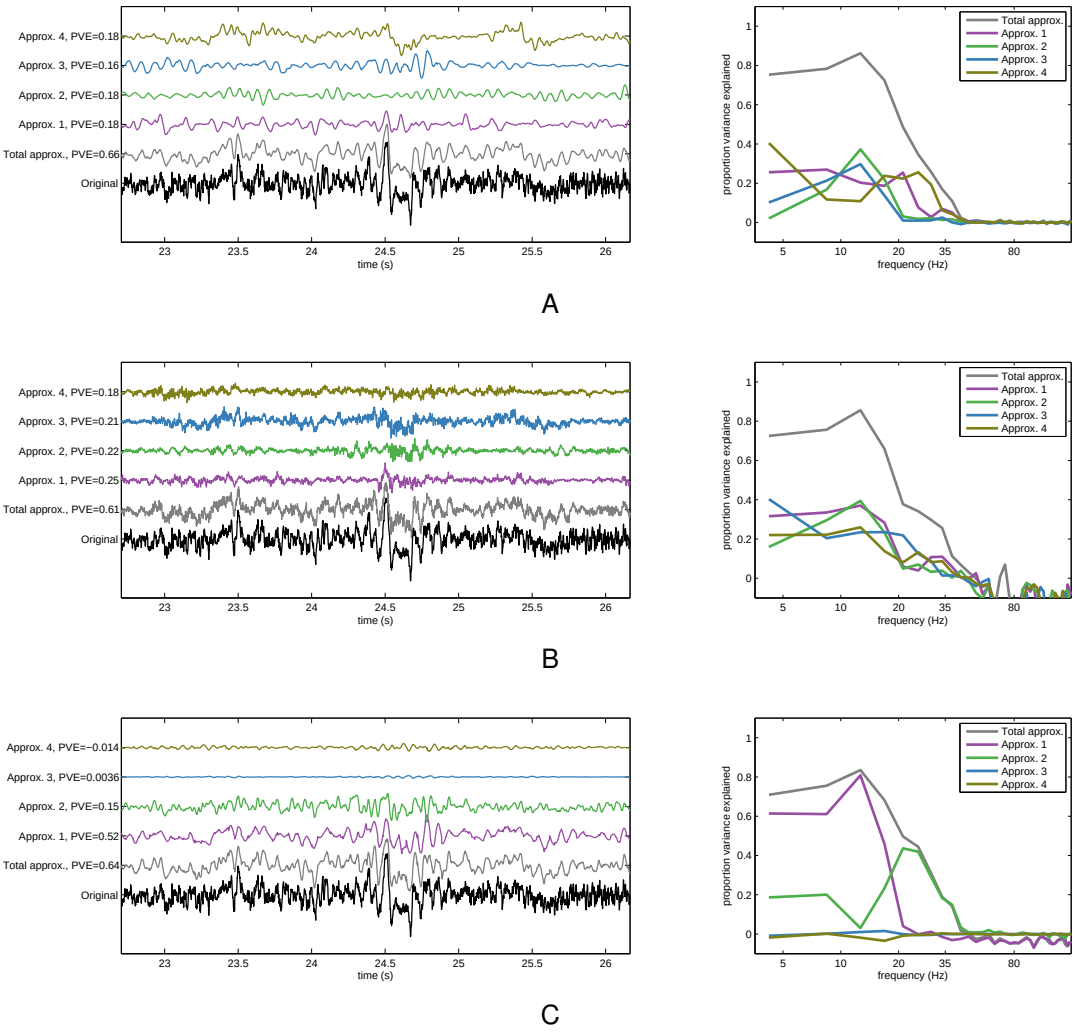


Figure 3-11. Analysis of a select channel of the multichannel decomposition of motor cortex LFPs. The decompositions are shown for a 3.5 s window of the data along with the approximation performance across the full segment across frequency for the three PCA-based spatial extensions of the algorithms: A) MP-SVD; B) FastICA; C) Greedy MP-SVD.

algorithms on a synthetic test with varying sparsity, and explored applications in the decomposition of neural potentials.

A variety of time-series analysis and machine learning tools were used: deconvolution (Shalvi & Weinstein, 1990), sparse coding (Aharon et al., 2006; Mailhé et al., 2008), iterative approximation algorithms (Mallat & Zhang, 1993) and atomic decomposition (Chen et al., 1998), alternating optimization (Stoica & Selén, 2004), and FastICA

Table 3-3. Multichannel approximation performance as proportion of variance explained on a select channel and across all channels.

	Training PVE (%)						Testing PVE (%)					
Method	1[a]	2	3	4	Total[b]	All[c]	1	2	3	4	Total	All
MP-SVD	18	18	16	18	66	53	18±4	15±2	12±4	19±2	61±4	48±4
FastICA	25	22	21	18	61	49	23±1	17±2	19±1	17±3	56±3	44±3
Greedy	52	15	0	0	64	60	45±4	18±2	0±0	0±0	59±4	56±3

[a]Individual component on select channel. [b]Using all components on select channel. [c]Across all channels.

(Hyvarinen, 1999). Nonetheless, it is still possible to incorporate other ideas from signal processing and machine learning. The blind system identification is essentially a modeling problem. In this framework, it is natural to add additional priors or constraints on either the sources or the filters. The most common adjustment is forcing the sources to be strictly positive (Roux et al., 2009), or constraining both the sources and filters to be positive (Smaragdis et al., 2008). These approaches are important since in some applications the unconstrained models lead to uninterpretable components.

Here an approximation of matching pursuit was used to obtain the source estimates, and the number of atoms extracted corresponds to the sparsity of the sources. From the results on real data, this leads to components which uniformly contribute to the variance explained. Alternatives for matching pursuit can provide better sparse coding accuracy (Chalasani et al., 2013; Kavukcuoglu et al., 2010). These algorithms do not enforce the hard, zero or non-zero, constraint on the source amplitudes that greedy algorithms such as matching pursuit enforce. The loosening of this constraint allows the filters to be optimized without assuming the exact timing was correctly estimated, a poor assumption at the early learning stages. In addition, these alternative methods allow adaptive priors on the source distributions, which may lead to the extraction of different components. For instance, the greedy method proposed finds components with decreasing energy, but localized at increasingly higher frequencies. This sort of decomposition resembles the empirical mode decomposition (EMD), a model free time-series analysis technique (Huang et al., 1998). The benefit of the greedy method is that it learns an underlying set

of filters that can be used on novel segments of the signal. For future work along these lines, explicit comparisons to wavelet and empirical mode decompositions should be made.

In terms of the neural potentials, the applied approaches match the underlying generating process: reoccurring waveforms that appear transiently in time (Brockmeier et al., 2011b). With classic time-frequency analysis it is difficult to separate the contribution of unique sources in the local field potential. Using the blind system identification both lower frequency components and high-frequency waveforms, even action potential waveforms (Ekanadham et al., 2011), could be learned and matching pursuit can be applied to separate their contributions.

On the segments of LFP analyzed, MP-SVD and the greedy approach for MP-SVD performed better than ICA in terms of reconstruction cost. Even with increasing number of approximation iterations ICA was not able to match the reconstruction accuracy of MP-SVD. This is reasonable because MP-SVD is tuned to explain the signal in terms of mean squared error; whereas, ICA is only searching for 'independent' sources. Indeed, ICA choose filters at higher frequencies that may indeed be independent of the lower frequency sources MP-SVD estimated.

Additionally, we used SVD across space to extend the single-channel algorithms to the multichannel case. The resulting filters for the multiple channel case were lower frequency waveforms. This relates to the spatial localization of high-frequency components in the LFP (Buzsáki et al., 2012). However, spatial ICA could be substituted for SVD in the multichannel extensions; this may lower the approximation performance, but the spatial components may be more meaningful (Delorme et al., 2012).

The importance of the atomic decomposition is not only its compression and reconstruction ability. The individually extracted components can be useful for the exploratory analysis of neural potentials. Alternatively, a complimentary analysis of the statistics of the sources, in terms of their atomic decompositions, could also be

performed as the atomic decomposition corresponds to a realization of a marked point process. This process is defined by the joint distribution over the timing, amplitude, and indices of the source excitations.

This alternative representation of sparsely excited sources may prove advantageous over classical time-frequency descriptors. For instance, it is natural to consider the dependence between an external stimuli or known cognitive state and the timing and amplitude of a particular source. Preliminary results have indicated that this sparse representation, which is non-linearly extracted from the signal, is more informative than using classical frequency bank approaches. In particular, this approach is used in the single-trial decomposition of evoked potentials in Chapter 4. The marked point process representation is a natural analog to the point process modeling used for neural spike trains.

CHAPTER 4
TRIAL-WISE DECOMPOSITIONS FOR NEURAL POTENTIALS

Electrical potentials in the brain are a combination of spatiotemporal oscillations, indicative of the communication between neural assemblies (Freeman, 2004), intermixed with a broadband component of unsynchronized activity. Brain waves are often associated with rhythmic activity but can take the form of a transient response following a specific cognitive or sensory event, such as brief flash of light (Adrian & Matthews, 1934). The transient responses is a spatiotemporal pattern and can be localized due to the brief involvement of the same neural circuits (Freeman, 2004). The responses following presentation of a specific stimulus or condition are referred to as evoked potentials (Ciganek, 1961), and they are indicative of neural processing.[1]

The recurrent characteristics of the evoked potentials to different stimuli or conditions can be represented by a spatiotemporal model (Freeman, 1979). Then the stimuli or conditions can be compared using these models (Emery & Freeman, 1969). This approach is particularly useful for investigating sensory and cognitive processing on a single-trial basis, and is used by brain computer interfaces that determine a user's attention solely from changes in their evoked potentials to various stimuli (Farwell & Donchin, 1988). To achieve this, methods are needed that readily differentiate between neural responses corresponding to different conditions.

The simplest model for the evoked potential under a specific condition is that the neural response is a combination of the same spatiotemporal waveform plus independent background activity and noise. Under this assumption, the model can be estimated by averaging across multiple trials; however, much of the trial-to-trial variability cannot be explained by a simple average.

[1] However, not all neural activity is related to processing; alpha waves ($\sim$10 Hz) (Adrian & Matthews, 1934; Berger, 1929) are indicative of the deactivation of the visual cortex, and are not present during visual activity.

For instance, the time between stimulus presentation and neural response may vary or the amplitude may vary across trials (Ciganek, 1969). Sometimes this variation stems from physiological effects such as habituation (Megela & Teyler, 1979). Additional free parameters can be added to the model to account for these trial-wise variations. For instance, one model of evoked potentials is a reoccurring temporal waveforms whose exact temporal location may shift and whose amplitude may vary across the trials. On any given trial, both the amplitude and time may be estimated and used for further analysis.

The scope of this study is to review, extend, and compare models for evoked potentials. The multichannel case is explicitly addressed, and clear connections are made with recent advances in tensor decompositions. An analysis of the model performance using model selection criteria is proposed as a consistent methodology to compare models based on different assumptions and with various degrees of freedom varies. However, model selection is the not the ultimate goal and is only considered for post-hoc assessment. Alternatively, this study considers the trial-varying parameters in the models as features for single-trial classification between conditions. In this way, the model fitting performance can be used to guide the model, prior to any classification.

The use of the condition or stimulus class information during model estimation is also considered. Model estimation is typically performed in an unsupervised manner across all trials, or in the partially supervised case where only examples relating to a single class are used. In these cases, the resulting features may not perform well for classification and full supervision may be needed, where the labels are used to form a discriminative model. In addition, varying forms of supervision may be used when fitting different modes of the model. For instance, we explore cases where the spatial factors are trained in a supervised manner at the same time that the temporal factors are trained without supervision.

The proposed methodology is applied to datasets consisting of local field potentials recorded from non-human primates during variable-reward experiments, where trials have the possibility of being rewarding or non-rewarding. The models are used to analyze differences in the evoked potentials between the rewarding and non-rewarding trials, and gauge the models' performance by their single-trial classification between the two conditions. In addition, on one dataset the relationship between the timing of evoked potentials and the neural firing rates is assessed. Overall, the study highlights a comprehensive methodology for using shift-varying models for the analysis and classification of evoked potentials on a single-trial basis.

4.1 Previous Work

The first method to compensate for variations in the temporal alignment of evoked potentials was proposed by Woody (1967). Woody's model is for the single-channel case and corresponds to a temporal waveform whose alignment shifts among the trials, but has fixed amplitude throughout. The method learns the waveform across multiple iterations of the correlation-guided averaging. This basic alternation between optimization and estimation converges and can be used with minimal assumptions: that the background noise is white. The estimation of multiple waveforms per trial is also considered, and it was demonstrated that the quality of the estimated waveforms is inversely proportional to the maximum cross-correlation between pairs waveforms.

Pham et al. (1987) propose the same signal plus noise model, but assume colored noise. The estimation is done in the frequency domain using maximum-likelihood approach. For simplification of the estimation, the shift in alignment is assumed to be small and the template is assumed to be low frequency. Jaskowski & Verleger (1999) add a trial-varying amplitudes to this model. Both models require more parameters to capture the power spectral density of the noise. Based on empirical evidence, both sets

of authors claimed superiority over Woody's algorithm yet they only performed a few iteration of Woody's alternating optimization: not letting it converge. [2]

Weeda et al. (2012) use a predefined basis to estimate a fixed waveform with trial-varying amplitude and alignment; however, in their model, shifts in alignment are limited to those that can be realized by the sum of the waveform and its first derivative, which can only be accurate for a single frequency, or small shifts at relatively low-frequencies: limiting assumptions common to the estimation proposed by Pham et al. (1987).

The aforementioned methods are extensible to having multiple evoked potentials occurring on each trials, yet they concentrated on single component aspect. Truccolo et al. (2003) propose to estimate multiple differentially variable components. The differentially variable components analysis (dVCA) allows waveforms to have independent shifts and amplitudes. Estimation is done in a Bayesian framework, and the waveforms are estimated using a weighted average of the residual. This is an approach that is also used for shift-invariant modeling of audio (Mailhé et al., 2008). This optimization does not always converge: Truccolo et al. (2003) perform only two iterations instead of using a convergence criterion. In the single component case, dVCA generalizes Woody's method to varying amplitudes.

One limitation of all of the aforementioned models is that they assume a fixed shape of the temporal waveform for a given component of the evoked potential. The fixed waveform assumption can be lifted by estimating each component as a linear combination of a set of waveforms, which form a subspace. Whereas a subspace model with a large enough subspace may account for both variability in waveform

[2] One could consider a time-domain method for extending Pham's method to include colored noise by estimating the noise correlation and re-estimating the waveform after pre-whitening.

and variability in alignment; an explicit temporal alignment cannot be derived from the subspace model.

Assuming fixed alignment, Karjalainen et al. (1999) estimate a temporal subspace to explain the evoked potential. Each individual trial is described by the subspace coefficients, the weights of the linear combination of waveforms. These coefficients are estimated with an added regularization constraint.

None of the models introduced so far explicitly consider multiple channels. In the multichannel case, a spatiotemporal model is needed to explain both the spatial variation and temporal pattern. If each channel is allowed a separate waveform, the number of coefficients in the model rapidly increases.

Assume the evoked potential is discretely sampled and represented by a matrix, with M time points and N channels. A full rank model has $M \cdot N$ coefficients; whereas, a rank-1 spatiotemporal model assumes the shape of the temporal waveform is common across channels and the amplitude varies between channels; the rank-1 model has $M + N$ coefficients. The use of a single spatial factor corresponds to a single source of the neural activity.

de Munck et al. (2004) consider the multichannel case: proposing a maximum-likelihood estimator for a full-rank spatiotemporal waveform with trial-varying amplitude, time-locked alignment (no shifts), and a rank-1 spatiotemporal noise model. A rank-1 model of noise assumes that there is a single spatial source with a fixed temporal covariance matrix.

Brockmeier et al. (2012b) propose a simple model for the multichannel case consisting of a predefined waveform, chosen from gamma-tone functions,[3] with trial-varying temporal alignments and trial-varying spatial amplitude vectors. This model only constrains the temporal aspects, and has a much larger number of free

[3] Gamma-tones are amplitude modulated sinusoids with gamma-function-shaped envelopes.

parameters. The waveforms are allowed to shift and the amplitude is allowed to vary across space. However, after model estimation, only the first two singular vectors of the spatial amplitudes are sufficient for classification. Similarly, Li et al. (2009) propose an extended estimation process with a single spatial factor that asks as a spatial filtering. The filter is estimated by taking the average across trials of the spatial amplitudes, and using this average as a spatial filter. The single-trial temporal alignments and spatial amplitudes are estimated again with the new filter. This process changes the norm, used across the channels, thus the selection of the best temporal alignment and spatial amplitudes are affected. The process is repeated until convergence, but the temporal waveform is still a predefined waveform—the authors used a specific gamma function—and is never adjusted.

Rivet et al. (2009) propose learning a fixed amplitude bilinear model, composed of spatial and temporal factors, where the spatial factors are made to discriminate from background activity. The authors also allow for the case that the trials partially overlap. The computation is efficiently computed using QR decompositions. Most recently, Souloumiac & Rivet (2013) show improved estimation of the temporal waveform when trial-varying alignment was combined with spatial enhancing filters. Their model consists of single spatiotemporal waveform with fixed amplitude whose alignment is allowed to vary between trials.

Herein we consider estimating a spatiotemporal model with trial-varying amplitude and alignment. Specifically, we propose a new model with a single, fixed spatial factor, varying temporal alignment, and also temporal waveform shape. The waveform on any given trial is formed from a learned subspace. This effectively combines previously proposed models (Karjalainen et al., 1999; Souloumiac & Rivet, 2013). We explore training the spatial factor in both a discriminative and non-discriminative fashion.

As the above models are defined across space, time, and trials it is natural to consider tensor models (Carroll & Chang, 1970; Harshman, 1970; Hitchcock, 1927;

Kolda & Bader, 2009). Tensor decompositions have been shown to be useful for analyzing EEG (Acar et al., 2007; Cichocki et al., 2008). As a space by time by trials data arrangement is still in three dimensions, the important aspects of the tensor decompositions can be illustrated by three-dimensional diagrams.

Another benefit of treating the evoked potential modeling as a tensor decomposition problem is the common framework for model selection. Model selection for tensors has been extensively studied (Bro & Kiers, 2003; Brockmeier et al., 2013b; Ceulemans & Kiers, 2006, 2009; He et al., 2009; Mørup & Hansen, 2009; Timmerman & Kiers, 2000). For instance, identifying an optimal model can be used to answer whether or not the data requires a shift-tolerant model or how many components are needed. Even for the matrix case, there is a full range of models where increasing rank corresponds to a better fit of the average Eckart & Young (1936). We explore using model selection criteria to choose the best model without cross-validation.

4.2 Mathematical Modeling Framework

We explore two levels of model parameter fitting: at the bottom level, parameters are fit for each trial, and at the top level, the parameters are fit across the trials. Borrowing nomenclature from signal processing, we use the term 'analysis' to indicate the extraction of trial-varying coefficients for a given input. Analysis is done on each trial independently; whereas, 'estimation' refers to adjusting the model coefficients that are fixed across trials. In both cases, the coefficients are adjusted to minimize the cost function, for which we use the mean squared error. The coefficients that are fixed across trials are referred to as 'factors'. For clarity, coefficients that vary per trial will be indicated by a star superscript ($a^\star$), or, when necessary to distinguish specific trials, with a trial indexing variable as a subscript (a_i).

4.2.1 Tensor Representation

We assume a discrete sampling in time and space, and organize the evoked potential for a single trial into a matrix . Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_M] \in \mathbb{R}^{L \times M}$ denote a sample

of the M channels in an L-length window. Let a set of n trials be denoted $\mathcal{X} = \{X_i\}_{i=1}^{n}$. Assuming all of the trials have the same dimensions, it is natural to treat this set of spatiotemporal evoked potentials as a three-way matrix, or tensor. Let $\underline{\mathcal{X}}$ correspond to the tensor representation of the set of trials in $\mathcal{X}$; organized into trials by time by space with dimensions $n \times L \times M$.

Then $\underline{\mathcal{X}}$ is a third-order (or order-3) tensor with 3 modes. A second-order tensor is a matrix; a first-order tensor is a vector; and a scalar can be considered a zeroth-order tensor. Any tensor can be converted to a column vector by simply concatenating all of the entries; this operation is denoted $\mathbf{x} = \text{vec}(\underline{\mathcal{X}})$. Correspondingly, unfolding refers to the operation that converts a order-3 or higher tensor into a matrix—the order along only one mode is retained. After unfolding, matrix operations to be applied to the tensor, and if appropriate the matrix can be folded back into a tensor. For the third-order tensor $\underline{\mathcal{X}}$ there are three possible unfoldings:

1. The 1-mode unfolding is an $n \times (M \cdot L)$ matrix $X^{(1)} = [\text{vec}(X_1), \text{vec}(X_2), \dots, \text{vec}(X_n)]^{\mathsf{T}}$.
2. The 2-mode unfolding is an $L \times (n \cdot M)$ matrix $X^{(2)} = [X_1, X_2, X_3, \cdots, X_n]$.
3. The 3-mode unfolding is an $M \times (n \cdot L)$ matrix $X^{(3)} = [X_1^{\mathsf{T}}, X_2^{\mathsf{T}}, \dots, X_n^{\mathsf{T}}]$.

Tensors can be built out of lower-order vectors, matrices, or tensors using an outer product. Given an order-N tensor $\underline{\mathcal{A}}$ and an order-P tensor $\underline{\mathcal{B}}$, the outer product of $\underline{\mathcal{A}}$ and $\underline{\mathcal{B}}$ is the order-$(N+P)$ tensor $\underline{\mathcal{C}} = \underline{\mathcal{A}} \otimes \underline{\mathcal{B}}$, with entries such that $\underline{\mathcal{C}}_{i_1,i_2,\dots,i_N,j_1,\dots,j_P} = \underline{\mathcal{A}}_{i_1,i_2,\dots,i_N}\underline{\mathcal{B}}_{j_1,j_2,\dots,j_P}$. The tensors in the outer products are referred to as factors, and tensors in a linear combinations of tensors are referred to as components.

Consider a series of outer products when each factor is a vector. An order-N tensor $\underline{\mathcal{X}}$ is considered to be rank-1 if it is formed from the outer product of N vectors, i.e., it can written as $\underline{\mathcal{X}} = \mathbf{a}^1 \otimes \mathbf{a}^2 \otimes \cdots \otimes \mathbf{a}^N$. Otherwise, the rank is R and is equal to the minimal number of rank-1 tensors needed as components such that $\underline{\mathcal{X}} = \sum_{r=1}^{R} \mathbf{a}_r^1 \otimes \cdots \otimes \mathbf{a}_r^N$ (Kruskal, 1977). The set of factors $\{\mathbf{a}_r^i\}_{i,r}$ form the canonical polyadic decomposition (CPD) (Hitchcock, 1927)—which is also called CANDECOMP (Carroll & Chang, 1970) or PARAFAC (Harshman, 1970).

Increasing the rank of the CPD introduces a new factor along each mode. Sometimes it may be desired to increase the number of factors only along certain modes. The Tucker model (Tucker, 1966) allows a different rank along each mode. Unlike the CPD, where a factor from each mode are associated with a single component, the Tucker decompositions relates the interaction between modes using a core tensor. The core tensor has the same number of modes and has dimensions equal to the ranks. The third-order Tucker model with core $\underline{\mathcal{G}}$ and factors A, B, and C is written $\underline{\mathcal{X}} = \underline{\mathcal{G}} \times_1 A \times_2 \times_3 C$, where $\times_k$ denotes tensor multiplication along the kth mode, which can be performed by appropriate unfolding, matrix multiplication, and folding operations (Kolda & Bader, 2009). The CPD is the case of a Tucker model with equal ranks and a super-diagonal core.[4]

De Lathauwer (2008) introduced block term decompositions that share the benefits of the CPD with the added flexibility of the Tucker models. The decompositions consider more general ranks, an order-3 tensor is rank-$(L, L, 1)$ if it can be written as an outer product between a rank-L matrix A and a vector $\mathbf{a}$, i.e., $\underline{\mathcal{X}} = A \otimes \mathbf{a}$. (The rank-1 term could be assigned to any mode by reordering the modes.) An order-3 tensor can be decomposed into a set of rank-$(L_r, L_r, 1)$ tensors as $\underline{\mathcal{X}} = \sum_{r=1}^{R} A_r \otimes \mathbf{a}_r$. Under certain conditions this is unique decomposition (De Lathauwer, 2008).

For real-valued third-order tensors, it is simple to consider the inner-product, norm, and Euclidean distance. The inner product of two tensors of the same size is

$$\langle \underline{\mathcal{A}}, \underline{\mathcal{B}} \rangle_F = \text{vec}(\underline{\mathcal{A}})^{\mathsf{T}} \text{vec}(\underline{\mathcal{B}}) = \sum_{i,j,k} \underline{\mathcal{A}}_{i,j,k} \underline{\mathcal{B}}_{i,j,k}. \tag{4–1}$$

[4] A tensor is super-diagonal if all entries that do not have the same index on each mode are non-zero, i.e., $\underline{\mathcal{A}}_{i,j,k} = 0$ if $i \neq j \neq k$.

The inner-product induces the Frobenius norm, which can be computed as

$$\|\underline{\mathcal{A}}\|_F = \sqrt{\langle \underline{\mathcal{A}}, \underline{\mathcal{A}} \rangle_F} = \sqrt{\sum_{i,j,k} \underline{\mathcal{A}}_{i,j,k}^2}. \tag{4–2}$$

The squared Euclidean distance between two tensors is

$$\|\underline{\mathcal{A}} - \underline{\mathcal{B}}\|_F^2 = \sum_{i,j,k} \left(\underline{\mathcal{A}}_{i,j,k} - \underline{\mathcal{B}}_{i,j,k}\right)^2. \tag{4–3}$$

Assuming $\underline{\mathcal{B}}$ is an approximation of $\underline{\mathcal{A}}$ this is proportional to the mean squared error. Normalizing the squared Euclidean distance by the total number of entries N, yields the mean squared error:

$$\frac{1}{N}\|\underline{\mathcal{A}} - \underline{\mathcal{B}}\|_F^2 = \sum_{i,j,k} \frac{1}{N} \left(\underline{\mathcal{A}}_{i,j,k} - \underline{\mathcal{B}}_{i,j,k}\right)^2. \tag{4–4}$$

Returning to the evoked potentials case, consider bilinear models of a spatiotemporal waveform. A rank-1 model is the outer product between a single pair of vectors, i.e., $A = \mathbf{u} \otimes \mathbf{w} = \mathbf{u}\mathbf{w}^\top$, where $\otimes$ denotes the outer product. The factor $\mathbf{u}$ is the temporal component and $\mathbf{w}$ is the spatial component. To allow the amplitude to vary across trials, an additional scaling term may be added. The trial-varying amplitude is denoted $s^\star$ so that the approximation for a given trial is $s^\star A = s^\star \mathbf{u} \otimes \mathbf{w}$. Considering the full set of trials, the model is a rank-1 CPD, $\underline{\hat{\mathcal{X}}} = \mathbf{s} \otimes \mathbf{u} \otimes \mathbf{w}$. A diagram of a rank-1 CPD is shown in Figure 4-1.

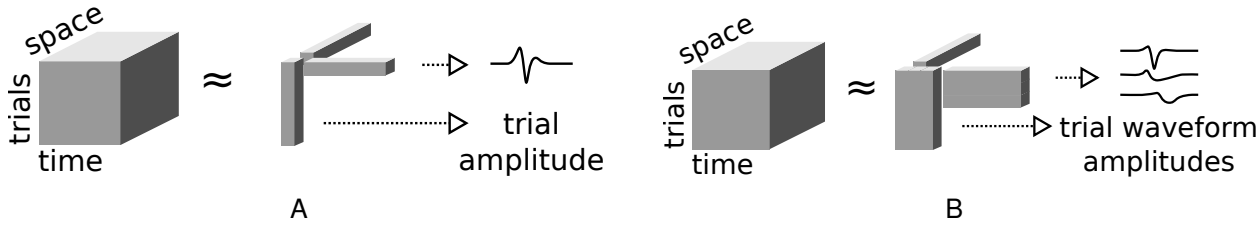


Figure 4-1. Diagrams of third-order tensor models for sets of evoked potentials. A) A rank-1 canonical polyadic decomposition (CPD) corresponds to the outer product between a spatial factor, temporal factor, and trial-varying amplitudes. B) A rank-(3,3,1) block term decomposition that allows each trial to be the linear combination of three waveforms, but a single spatial factor.

Instead of having a single temporal waveform to represent all the trials, a more general model assumes the temporal waveform on a given trial exists in some subspace. Consider the subspace spanned by the columns of B, a individual trial may have temporal aspect $\mathbf{u}^\star = B\mathbf{a}^\star$, where the subspace coefficients are stored in $\mathbf{a}^\star$. The resulting model is $\mathbf{u}^\star \otimes \mathbf{w} = B\mathbf{a}^\star \otimes \mathbf{w}$, which is still rank-1 across time and space, but has more degrees of freedom to describe the temporal aspects. In terms of a tensor model, the models is formed by first concatenating the subspace coefficients across all of the trials $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]^\mathsf{T}$ into a matrix. Then if B is rank-D the model is a block term decomposition with rank-$(D, D, 1)$ (De Lathauwer, 2008), $\underline{\hat{\mathcal{X}}} = AB \otimes \mathbf{u}$. A diagram of this model is shown in Figure 4-1.

For model estimation, a unique parameterization is important. Models involving multiple factors are not uniquely defined if the norms of the factors are allowed to vary independently. Therefore, any models that use both scalar amplitudes and vectors or matrices fixed across trials are constrained such that their factors have unit norm $\|\mathbf{u}\|_2 = \|\mathbf{w}\|_2 = \|A\|_2 = 1$. This constraint also simplifies the mathematical analysis.

4.3 Models with Variable Temporal Alignment

In this section we propose models in which the time-series waveform is modeled as a single waveform or a linear combination of multiple waveforms whose temporal alignment and scaling may vary on a given trial. We restrict our attention to models where all of the waveforms have the same temporal alignment per trial.

Let $\mathbf{x} \in \mathbb{R}^{L\times 1}$ and $\mathbf{g} \in \mathbb{R}^{N\times 1}$, $N \leq L$ denote a discretely-sampled time series and a waveform, respectively. Define T_τ to be the linear operator such that $T_\tau\mathbf{g}$ temporally aligns a waveform $\mathbf{g}$ such that it starts at time τ (Mailhé et al., 2008). Then $\tau - 1$ is the number of zeros that need to be pre-padded to $\mathbf{g}$; if $\tau < 1$ the initial $1 - \tau$ elements of $\mathbf{g}$ are truncated, and if $\tau > L-N+1$ the final $\tau-(L-N+1)$ elements are truncated. Fig. 4-2 shows the range of alignments for $\mathbf{g}, \mathbf{x}$. T_τ is an N by L matrix with the $N \times N$ identity matrix as a submatrix starting at row τ. Let T_τ^* denote the adjoint of this operator such

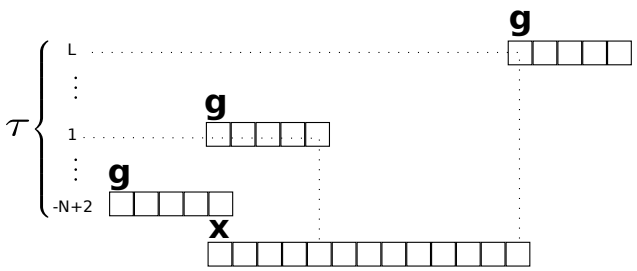


Figure 4-2. The range of alignments between the temporal waveform and the signal. The possible alignment ranges for $\tau \in \{-N+2, \dots, 1, \dots, L\}$ of a waveform $\mathbf{g}$ of length N to the L-length signal $\mathbf{x}$.

that $\langle \mathbf{x}, T_\tau \mathbf{g} \rangle = \langle T_\tau^* \mathbf{x}, \mathbf{g} \rangle$. The adjoint selects N-length window of time-series starting at time τ, and $T_\tau^* = T_\tau^\top$.

The best lag in terms of least squares can be found as:

$$\arg\min_\tau \|\mathbf{x} - T_\tau \mathbf{g}\|_2. \tag{4–5}$$

Squaring and expanding the norm yields:

$$\|\mathbf{x} - T_\tau \mathbf{g}\|_2^2 = \langle \mathbf{x} - T_\tau \mathbf{g}, \mathbf{x} - T_\tau \mathbf{g} \rangle \tag{4–6}$$

$$= \langle \mathbf{x}, \mathbf{x} \rangle + \langle T_\tau \mathbf{g}, T_\tau \mathbf{g} \rangle - 2\langle \mathbf{x}, T_\tau \mathbf{g} \rangle. \tag{4–7}$$

When $\mathbf{g}$ is aligned completely within $\mathbf{x}$, the solution to Equation (4–5) is equivalent to the solution to $\arg\max_\tau \langle \mathbf{x}, T_\tau \mathbf{g} \rangle$. This is useful since $\langle \mathbf{x}, T_\tau \mathbf{g} \rangle = (\mathbf{x} \star \mathbf{g})(1-\tau) = \sum_n \mathbf{x}_n \mathbf{g}_{n-\tau+1}$, where $\star$ denotes the cross-correlation—i.e., convolution with the time-reversed filter—which can be computed across all lags using the fast Fourier transform.

4.3.1 Windowed Tensor Representation

Given a set of timings for all the trials $\{\tau_i\}_{i=1}^n$, let $\breve{\mathcal{X}} = \{\breve{X}_i\}_{i=1}^n$ denote the set of matrices corresponding to the realigned potentials, $\breve{X}_i = T_{\tau_i}^* X_i$, $i = 1, \dots, n$. This set of windows can also be organized as a tensor $\underline{\breve{\mathcal{X}}}$. The approximation of this tensor is denoted $\underline{\tilde{\mathcal{X}}}$.

4.3.2 First Model

Consider a model characterized by a fixed spatial vector $\mathbf{w} \in \mathbb{R}^{M\times 1}$, a fixed temporal waveform $\mathbf{g} \in \mathbb{R}^{N\times 1}$, $N \leq L$, a variable temporal alignment $\tau^\star \in \mathbb{Z}$, and a variable scalar amplitude $s^\star$. For a set of n trials this model has $2n + N - 1 + M$ free parameters. The single-trial approximation is given by $\hat{X} = s^\star T_{\tau^\star}\mathbf{g} \otimes \mathbf{w}$. The approximation of the windowed tensor is a rank-1 CPD: $\underline{\tilde{\mathcal{X}}} = \mathbf{s} \otimes \mathbf{g} \otimes \mathbf{w}$, where $\mathbf{s}$ is the vector of amplitudes across the trials.

The analysis for this model consists of a static spatial projection of the spatiotemporal window $\mathbf{u} = \sum_{m=1}^{M} w_m \mathbf{x}_m = \mathbf{w}^\mathsf{T} X$, followed by finding the alignment that maximizes the norm of the coefficient. The temporal alignment $\tau^\star$ is found as

$$\tau^\star = \text{argmax}_\tau |\langle \mathbf{u}, T_\tau \mathbf{g} \rangle|, \tag{4–8}$$

and the amplitude $s^\star$ is found as $s^\star = \langle \mathbf{u}, T_{\tau^\star}\mathbf{g} \rangle$.

4.3.3 Second Model

Consider a model similar to the first model but characterized by a fixed set of D linearly independent temporal waveforms $G = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_D] \in \mathbb{R}^{N\times D}$, $N \leq L$. This set spans the subspace for the modeled temporal waveform.[5] The model is still characterized by a variable temporal alignment $\tau^\star \in \mathbb{Z}$, but now a variable vector of subspace coefficients $\mathbf{a}^\star = [s_1^\star, \ldots, s_D^\star]^\mathsf{T} \in \mathbb{R}^{D\times 1}$ is used. For a set of n trials this model has $n + n \cdot D + N \cdot D - D^2 + M$ free parameters. The single-trial approximation is given by $\hat{X} = T_{\tau^\star} G\mathbf{a}^\star \otimes \mathbf{w}$. The approximation of the windowed tensor is rank-(D,D,1) block term decomposition: $\underline{\tilde{\mathcal{X}}} = AG \otimes \mathbf{w}$, where $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$.

[5] The previous model is a special case of this model when $D = 1$.

The analysis of this model requires the pseudo-inverse[6] of $T_\tau G$. Again, the alignment that maximizes the norm of the subspace coefficient vector, and minimizes the reconstruction cost, is found as

$$\tau^\star = \text{argmax}_\tau \|(T_\tau G)^\dagger \mathbf{u}\|_2 \tag{4–9}$$

and $\mathbf{a}^\star = (T_{\tau^\star} G)^\dagger \mathbf{u}$, respectively.

4.4 Spatial Covariance-based Models

In this section we propose models with fixed spatial covariance, but with a temporal waveform that is allowed to vary per trial. In terms of neurophysiology, a model with a single fixed spatial factor corresponds to a single source. It assumes the relevant portion of the neural response corresponds to a specific neural population. The product with the spatial vector acts as a simple weighted average, reducing the signal to a univariate time series. In general, spatial factors project the signal to a lower dimensional subspace.

4.4.1 Single Source

Consider a model characterized only by the fixed spatial waveform $\mathbf{w} \in \mathbb{R}^{M\times 1}$ and variable temporal waveform $\mathbf{u}^\star = [u_1^\star, \dots, u_L^\star]^\mathsf{T} \in \mathbb{R}^{L\times 1}$. Given n trials, this model has $n \cdot L + M - 1$ free parameters. The single-trial approximation is given by $\hat{X} = \mathbf{u}^\star \otimes \mathbf{w}$. The approximation of the full tensor is a rank-1 Tucker-1 decomposition: $\underline{\hat{\mathcal{X}}} = U \otimes \mathbf{w}$, where $U = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]$. The analysis for this model consists of a static spatial projection of the spatiotemporal window $\mathbf{u}^\star = \sum_{m=1}^{M} w_m \mathbf{x}_m = \mathbf{w}^\mathsf{T} X$.

4.4.2 Spatial Subspace

Consider a model similar to the previous model but characterized by a fixed set of E linearly independent spatial vectors $W = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_E] \in \mathbb{R}^{M\times E}$, which form a subspace for the spatial pattern. The model is characterized by a multivariate

[6] Let B be a matrix with linearly independent columns, then $B^\dagger = (B^\mathsf{T} B)^{-1} B^\mathsf{T} \in \mathbb{R}^{D\times N}$ is its pseudo-inverse.

time-series $U^\star = [\mathbf{u}_1^\star, \ldots, \mathbf{u}_E^\star]^\mathsf{T} \in \mathbb{R}^{L \times E}$. For a set of n trials this model has $n \cdot L \cdot E + M \cdot E - E^2$ free parameters. The single-trial approximation is given by $\hat{X} = U^\star W^\mathsf{T}$. The previous model is a special case of this model when $E = 1$. The approximation of the full tensor is a rank-E Tucker-1 decomposition: $\underline{\hat{\mathcal{X}}} = \underline{\mathcal{U}} \times_3 W$, where $\underline{\mathcal{U}}$ is the tensor representation of the set of multivariate time-series across all trials $\{U_1\}_{i=1}^n$. The analysis of this model is given by $U^\star = W^\dagger X$.

4.5 Fitting the Spatiotemporal Models

Model fitting is performed in an alternating process where the analysis equations for each model provide an estimate of the coefficients on a given trial, and the model is updated to best explain—in terms of minimizing the mean squared error—the data given these coefficients. For the temporal factors, we use only non-discriminative model fitting, but the model is fit using only trials of one condition. This partial supervisory information does not consider trials of other conditions. For the spatial factors, we do consider discriminative models based on the spatial covariance. Overall, the spatial and temporal factors are alternatively updated.

For the models allowing temporal shift, only the windowed portion of the signal, corresponding to the alignment of the waveform on a particular trial, is used to update the model. For the first two models discussed, this window is found after the signal has been projected to a single channel via the spatial factor. The window alignment is selected for each trial given the current model.

4.5.1 Updating the Temporal Factors

After alignment each spatiotemporal matrix is multiplied by the spatial factors W and the resulting products are concatenated across trials into another matrix $U = [\breve{X}_1 W, \breve{X}_2 W, \ldots, \breve{X}_n W] = \breve{X}^{(2)} W$, where $\breve{X}^{(1)}$ is the mode-2 unfolding of $\underline{\breve{\mathcal{X}}}$.[7]

[7] Recall that $\underline{\breve{\mathcal{X}}}$ denotes the tensor corresponding to the realigned potentials, $\breve{X}_i = T_{\tau_i} X_i$, $i = 1, \ldots, n$.

The waveforms are updated to be the D singular vectors corresponding to the D largest singular values of this matrix, which is the solution to the following optimization consisting of the ratio of the determinants:

$$G = \arg\max_{V} \frac{|V^{\mathsf{T}} U U^{\mathsf{T}} V|}{|V^{\mathsf{T}} V|}. \tag{4–10}$$

Choosing the updated filters in this way minimizes the mean squared error of the model, as the singular value decomposition is the best approximation of matrix (Eckart & Young, 1936).

4.5.2 Updating the Spatial Factors

The spatial projection is also updated from the covariance of the aligned windows of $\underline{\breve{\mathcal{X}}}$. The spatial covariance, without removing the mean, is computed as $R_0 = \sum_i^n \frac{1}{n} \breve{X}^{\mathsf{T}} \breve{X} = \frac{1}{n} \breve{X}^{(3)} \breve{X}^{(3)\mathsf{T}}$, where $\breve{X}^{(3)}$ is the mode-3 unfolding of $\underline{\breve{\mathcal{X}}}$. For discrimination, we consider having an auxiliary matrix R_1, which is the spatial covariance during background or different conditions. This can be used to find projections which maximize a ratio of the variance between the conditions; replacing this auxiliary matrix with an identity matrix will minimize the error. The spatial projection is found as the matrix that maximizes the following ratio of determinants:

$$W = \arg\max_{V} \frac{|V^{\mathsf{T}} R_0^{\mathsf{T}} V|}{|V^{\mathsf{T}} R_1 V|}. \tag{4–11}$$

The solution to this optimization is chosen from the solutions $\{(\lambda_j, \mathbf{v}_j)\}$ for the generalized eigenvalue problem $R_0 \mathbf{v}_j = \lambda_j R_1 \mathbf{v}_j$. The columns of W should be chosen as the E eigenvectors $\mathbf{v}_j$ corresponding to the E largest generalized eigenvalues λ_j. In the pattern recognition literature this is known as the Fukunaga-Koontz transform (Fukunaga & Koontz, 1970; Zhang & Sim, 2007), in the EEG analysis literature it has became popular for extracting features for motor imagery classification and is known as common spatial patterns (CSP) (Ramoser et al., 2000).

4.5.3 An Alternating Optimization Algorithm

Given the updated model, the spatial factors and any trial-varying coefficients, alignment and amplitude coefficients, are re-estimated. One iteration cycle of the full optimization computes the following steps:

1. Alignment via Equation (4–8) or Equation (4–9)—Input: $\underline{\mathcal{X}}$, W, G, Output: $\underline{\breve{\mathcal{X}}}$
2. Temporal factor update via Equation (4–10)—Input: $\underline{\breve{\mathcal{X}}}$, W, D, Output: G
3. Spatial factor update via Equation (4–11)—Input: $\underline{\breve{\mathcal{X}}}$, R_1, E, Output: W

Ideally, after multiple iterations the factors and coefficients will converge. In practice, this is not always the case, and the process should be stopped after a fixed number of iterations. A diagram illustrating an overview of the algorithm is shown in Figure 4-3.

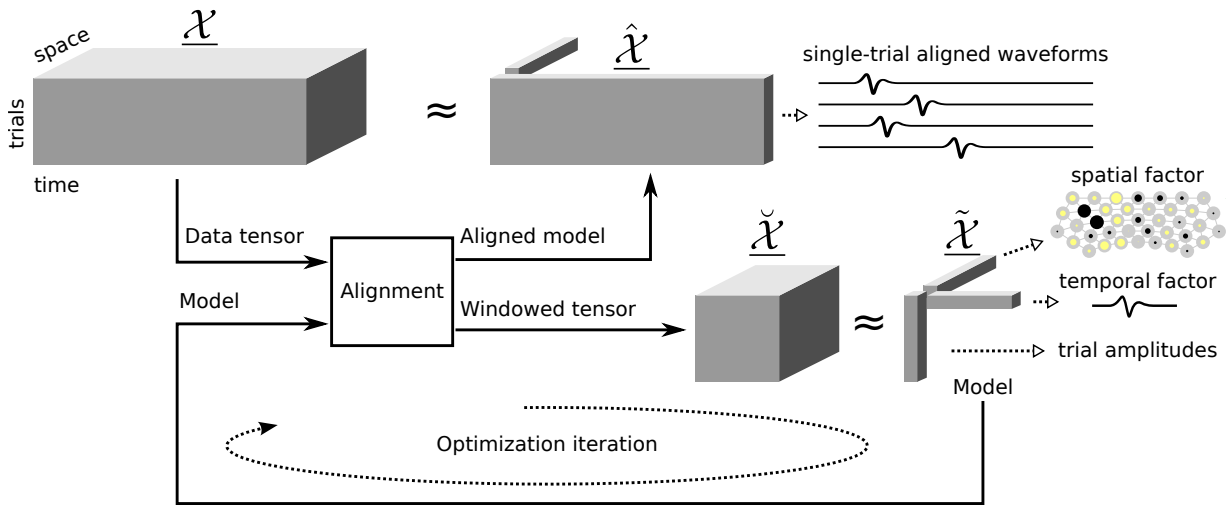


Figure 4-3. Diagram of the optimization process consisting of temporal alignment and tensor decomposition. Here the model of the aligned tensor is a canonical polyadic decomposition (CPD). Although, the windows are approximated using the CPD, the aligned model cannot be considered a CPD, instead it is considered a "Tucker-1" model (Kolda & Bader, 2009; Tucker, 1966) with specific structure.

4.6 Model Selection

The performance trade-off between different model structures or model orders can be quantitatively assessed using a model selection criterion. Depending on the sizes of the waveform and the fit of each model, a model that allows shifts may be more or less efficient than a fixed alignment model.

The most straightforward approach to select a model is to test its ability to predict novel data. Cross-validation can be used to test this using multiple divisions of the trials into training and validation sets. A simple model would perform poorly; whereas, an overly complex model that over-fits—i.e., a model that is tuned to the particulars of the noise in the training samples not the general structure—would also perform poorly on novel data. However, with complex models that already involve alternating optimizations, this approach will require excessive computation. Alternatively, model selection criteria can be used to select a model that balances the approximation performance and the model complexity. As fitting error decreases with model complexity, criteria penalize the number of degrees of freedom in the model (Stoica & Selen, 2004).

The Bayesian Information Criterion (BIC) Schwarz (1978) is a consistent way to compare tensor models of different structure (Brockmeier et al., 2013b). Whereas Akaike Information Criterion (Akaike, 1974) is often used, it is only applicable to nested models—models built by adding parameters to parent models, and should not be used to select among non-nested models (Murata et al., 1994).

BIC is formulated in terms of the log-likelihood, but for a signal plus noise model, where the noise is assumed to be i.i.d. zero-mean and Gaussian distributed, only the mean squared error is needed. In this case, the following optimization problem selects the optimal model from a set of models $\mathcal{M}$:

$$\arg\min_{\hat{\underline{\mathcal{X}}}\in\mathcal{M}} BIC(\hat{\underline{\mathcal{X}}}) = M \ln\left(\frac{1}{M}\|\underline{\mathcal{X}} - \hat{\underline{\mathcal{X}}}\|_F^2\right) + k\left(\hat{\underline{\mathcal{X}}}\right)\ln(M) + C \tag{4–12}$$

where $\underline{\mathcal{X}}$ denotes a tensor with M elements, $\hat{\underline{\mathcal{X}}}$ is the tensor model with $k\left(\hat{\underline{\mathcal{X}}}\right)$ degrees of freedom, and C is a constant that is a function of the noise variance and is independent of $\hat{\underline{\mathcal{X}}}$.

4.7 Using the Models for Classification

To quantify the information captured by the model we investigate single-trial classification. We specifically consider the binary case for two classes of conditions. In

the classification setting, the dataset is divided into the training and testing set and only the training set is used to fit the factors of the spatiotemporal model. We use two model fitting paradigms:

1. Partially supervised modeling: model is trained using only one condition
2. Supervised modeling: spatial factors are trained to be discriminative between conditions, but temporal factors are trained using only one condition

The spatial and temporal factors constrain the waveform of the evoked potential. The amplitude, power, and/or alignment of this waveform is extracted on all trials, both training and testing, and used as features for classification. Standard classifiers, such as nearest-neighbor, linear discriminant analysis, or support vector machines can then employ the training set examples of these trial-varying parameters to form classification decision boundaries.

4.8 Reward Representation in Striatal LFPs during a Reach Task

Using data from a reward expectation experiments, we test the method's ability to model the neural response during two different stimulus conditions—presentation of rewarding and non-rewarding objects—and discriminate between them.

4.8.1 Data Collection and Experimental Setup

All surgical and animal care procedures were consistent with the National Research Council Guide for the Care and Use of Laboratory Animals and were approved by the University of Miami Institutional Animal Care and Use Committee.

A marmoset monkey, *Callithrix jacchus*, was trained to sit and reach for food items. After training, the subject underwent a craniotomy surgery while under isoflurane anesthesia. A 16-channel tungsten microelectrode array (Tucker-Davis Technologies, Alachua, FL) was implanted in both the motor cortex (M1) and the striatum, targeting the nucleus accumbens of the ventral striatum. Local field potentials (LFPs) were processed by a RZ2 processor (Tucker-Davis Technologies, Alachua, FL) and filtered through a cascaded of a 1 Hz high-pass and a 500 Hz low-pass, first-order, biquad

filters with 12 dB/octave roll-off and stored sampled at 1017.3 Hz. The common average reference was used to improve the signal-to-noise ratio. Only the LFPs from the array in the stratium were used in the analysis. LFPs were digitally filtered with a 3rd-order Butterworth high-pass filter (cutoff of 0.1 Hz) to remove low-frequency content. To maintain zero-phase, filtering was done both forward and backward resulting in a 6th-order high-pass filter. The data was further filtered with a Gaussian window with length of 40 samples and standard deviation of 8 samples. This corresponds to a low-pass filter with a cutoff of 33.8 Hz.

The experiment consisted of revealing objects held behind a door within reach and directly in front of the restrained subject. As the objects were either a mushroom piece, a rewarding object for which the subject would reach and eat, or a wooden bead, a non-rewarding target for which the subject would not reach, the set-up could alternatively be considered a go-no/go task or a variable reward task. Each trial began with the subject resting its hand on a touch pad for a random hold period. At the end of the hold period, the door would open revealing either a mushroom piece or a wooden bead in one of four boxed locations: top, bottom, left, and right. Once the door opens the subject sees the object and make a decision to reach or not. Here we analyze a dataset for a single session. There were 83 trials in total; 40 trials with mushrooms and 43 trials with beads. We use only the LFPs in the 1 second window after the door opens.

4.8.2 Model Design

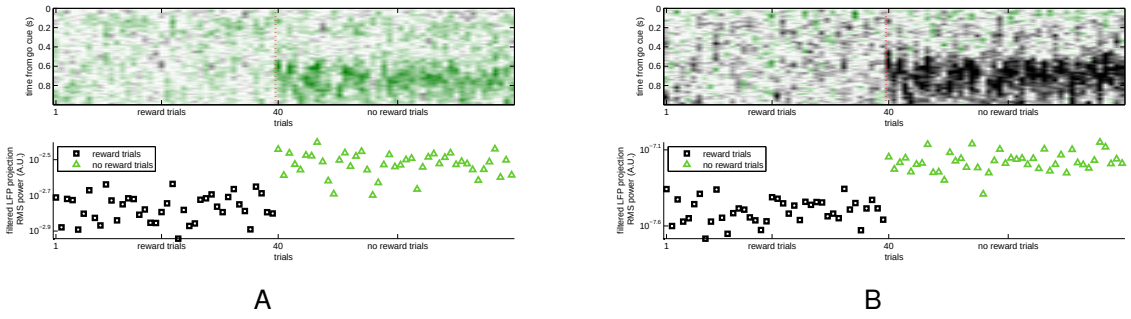
For this experiment, we anticipated strong modulation for the non-rewarding tasks. Because of this, the model of the temporal waveforms was trained using only the non-rewarding trial. Based on this partial supervision, the model should be well-suited to explain the non-rewarding trials. Alternatively, we compare using discriminative spatial projections trained using both conditions, rewarding and non-rewarding.

Three different spatiotemporal models are estimated using the training set. The first model uses a single temporal waveform and spatial factor, a rank-1 model, and it is

trained using partial supervision. The second model is also rank-1 but the spatial factor is discriminatively trained. The third model uses a subspace of 4 temporal waveforms to approximate each response, and the model is trained using only the non-rewarding condition trials.

4.8.3 Results

We illustrate the processing of LFPs using the spatial and temporal factors in spatiotemporal models. First we compare the temporal aspects of the LFP components after spatial projections, one trained in the semi-supervised manner and one trained discriminately, the signals for each trial are shown in Figure 4-4. In addition, in order to quantify the differential variation between the two conditions, we compute the root-mean-squared (RMS) power of the projected signal for each trial. The temporal and spatial factors for the 3 spatiotemporal models are estimated from the training set and are shown in Figure 4-5.



A B

Figure 4-4. LFPs are projected to a one-dimensional time series and the root-mean-squared, RMS, power is computed for each trial. A) Spatial filter that explain the maximum energy during the non-rewarding trials. B) Spatial filter that discriminates the two reward conditions.

To illustrate the single-atom decomposition, the approximations across select trials for each model are shown in Figure 4-6. The non-rewarding trials have larger amplitude oscillations, and more consistent timing. This fact can also be seen by examining the processing stages during the temporal subspace decomposition; shown in Figure 4-7. When using the waveform magnitude and alignment as features, there is a consistent

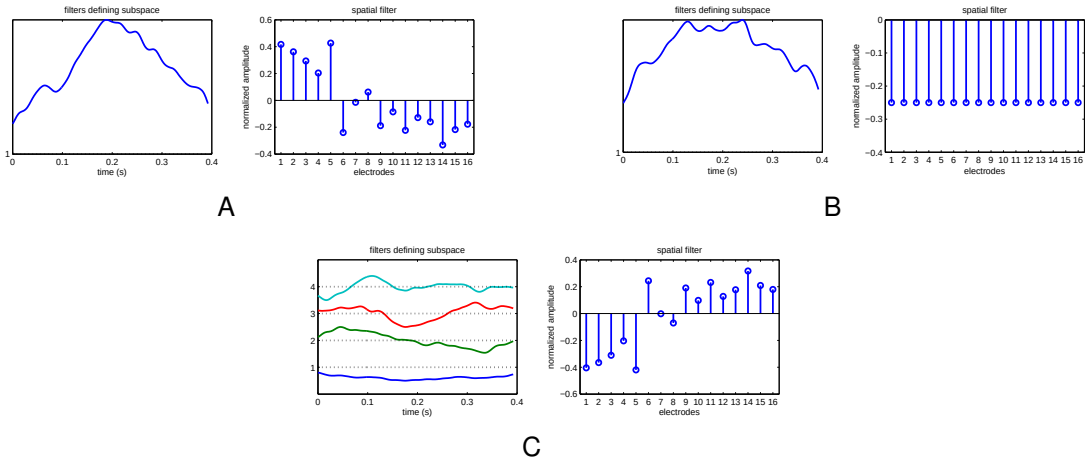


Figure 4-5. Spatial and temporal filters of the spatiotemporal decompositions with different ranks objectives. A) Model trained to explain the maximum energy during the non-rewarding trials. B) Model trained to be spatially discriminative; whereas the spatial filter is the projection that discriminates the conditions in terms of variance. C) Model uses a temporal subspace to explain the maximum energy during the non-rewarding trials. C) A set of temporal filters are trained to span the subspace with the maximum energy during the non-rewarding trials.

pattern under the non-rewarding conditions as shown in Figure 4-8. The concentrated points in the scatter plot indicate a clear mode in the temporal alignment of the waveform under the non-rewarding condition; whereas, in the rewarding condition the temporal alignment of the waveform is inconsistent with an almost uniform distribution for the alignments. The conditions are clearly separated by their single-trial amplitude. Similar separation is evident in the power of the signals as shown in Figure 4-4.

We tested the classification performance when using the various models' trial-varying parameters as features. Specifically we compared 4 sets of 3 features:

1. Time alignment of the temporal waveform
2. Scalar magnitude of temporal waveform (log-transformed)
3. Time-series RMS power after spatial projection (log-transformed)
4. A combination of (1) and (2)

For classifiers we used both a non-parametric and a parametric classifier: nearest neighbor (1NN) and linear discriminant analysis (LDA), respectively. For each we

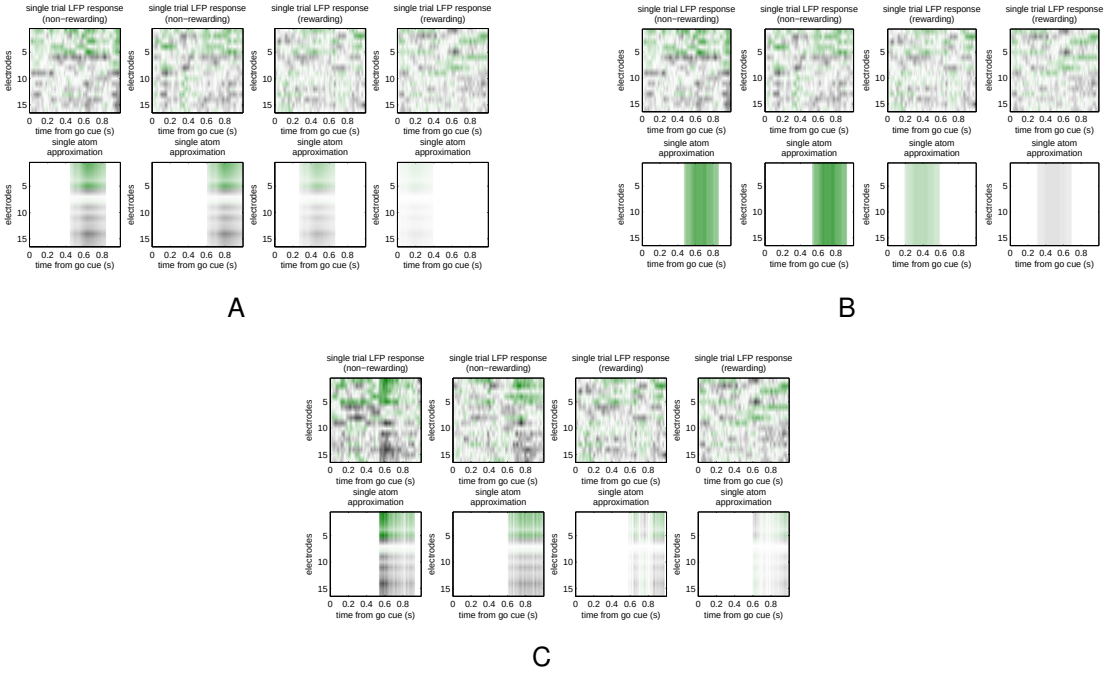


Figure 4-6. Multichannel LFPs from the straitum and rank-1, single-atom model approximations of four example trials, two from each reward condition. A) Model trained to explain the maximum energy during the non-rewarding trials. B) Model trained to be spatially discriminative. C) Model uses a temporal subspace to explain the maximum energy during the non-rewarding trials.

adjusted the number of training examples used for fitting the model and training the classifier, and performed 20 Monte Carlo divisions of the samples into training and testing sets. The 1NN classifier shows much higher performance, which increases with the number of training examples; whereas, LDA's performance did not increase with more training samples. The classification accuracy across training set sizes is shown in Figure 4-9. It appears that a minimum of 20 training examples is needed for consistent performance. Subsequently, we focus on this number of examples.

Given the classifier type and training set size, we compare the performance using the different features and models. In Table 4-1 we report the average and standard deviation in the classification accuracy (% correct) using the 1NN classifier and a training size of 20. Across the models, using both time and amplitude as features

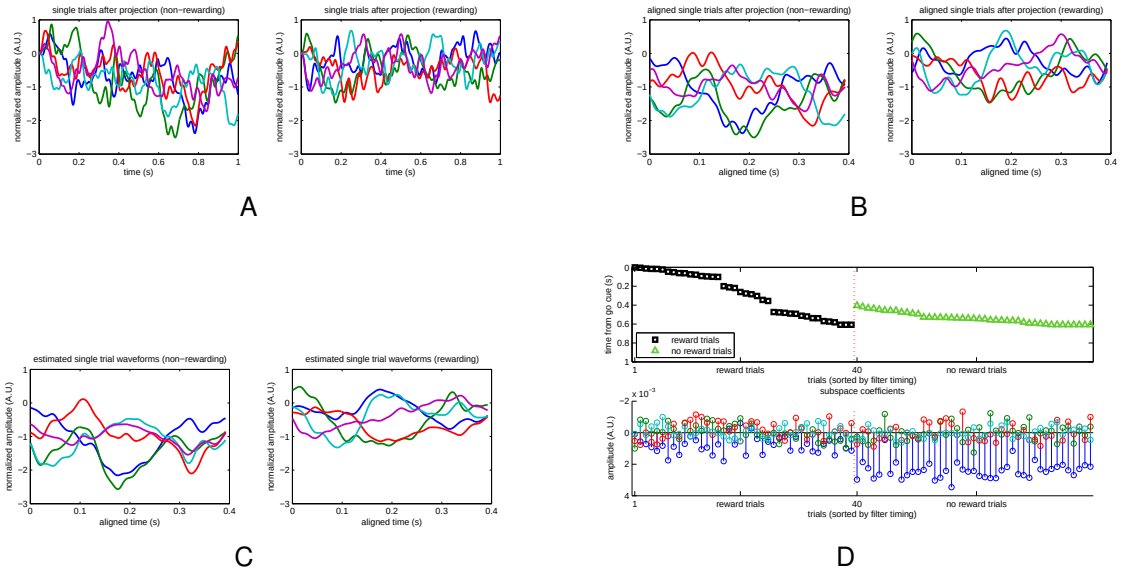


Figure 4-7. Illustration of the processing stages for the model using a temporal subspace. For each condition, 5 example trials are shown. A) The one-dimensional signals after applying the spatial projection. B) The windowed signal, where the timing is found by maximum energy in the subspace. C) Approximation of the windowed signal using the 4 filters in the subspace. D) Atomic representation, timing and amplitude coefficients, across all trials, separated by reward condition.

performs the best. Across all features except the time alignment, the model with a discriminative spatial performs the best. The temporal subspace model most benefits from using both the temporal alignment in conjunction with the amplitude. Throughout the models, using only the timing does not perform as well, but still is significantly above chance. For higher training set samples, as shown in Figure 4-9, using the timing extracted from the subspace model as the sole feature achieves a classification rate of above 90%. This means that the timing (delay) of the evoked potential is better estimated using the subspace model versus a single waveform.

In summary, we have used spatiotemporal models of evoked potentials to extract features from LFPs useful for discriminating between two different conditions. These methods are more computationally expensive than using the power of the signal, but beyond the classification rate improvement they are able to provide clearer insight into

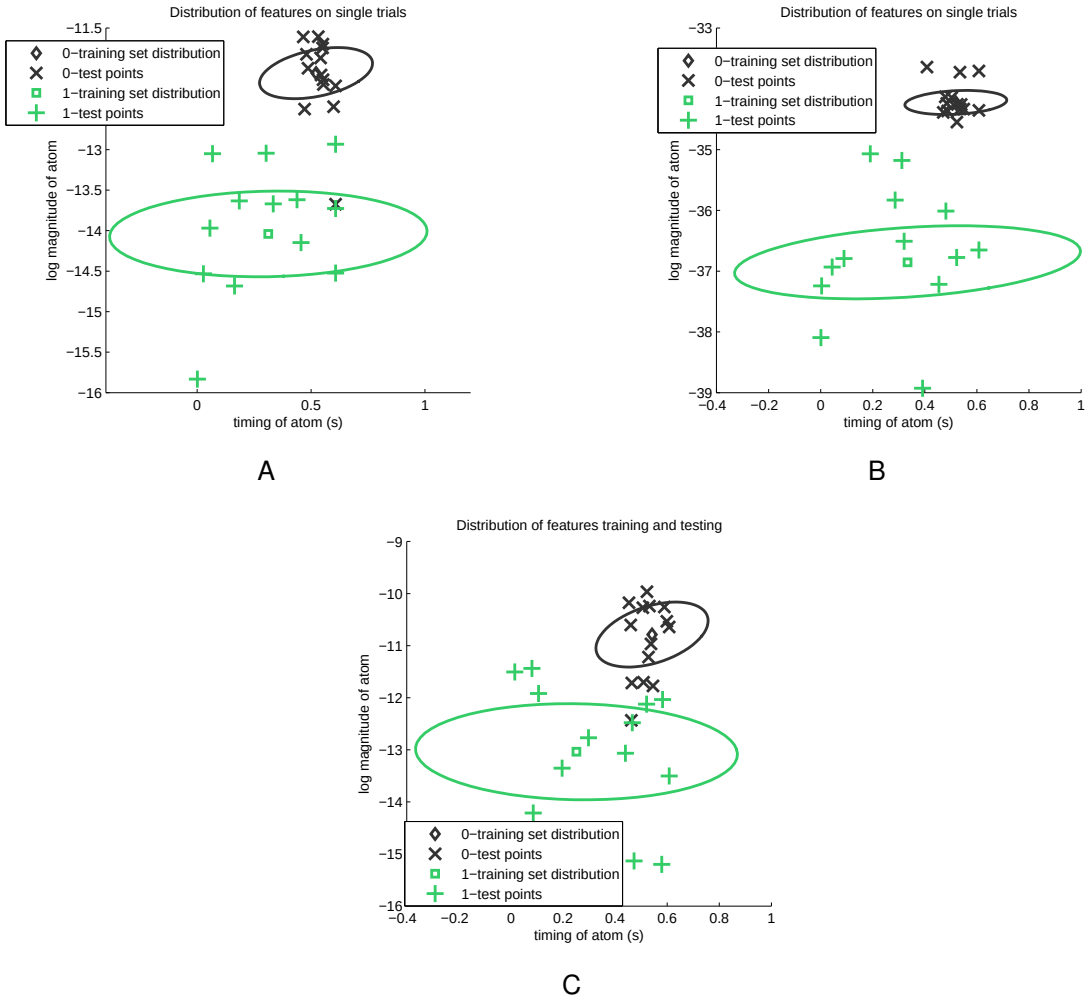


Figure 4-8. Timing and magnitude scatter plots for atomic decompositions of test set examples and normal distributions fitted to the training set. A) Model trained to explain the maximum energy during the non-rewarding trials. B) Model trained to be spatially discriminative. C) Model uses a temporal subspace to explain the maximum energy during the non-rewarding trials.

the single-trial evoked potentials, in terms of both the variability of the timing and of waveform shape. In particular, the timing information was found to be most informative when using the subspace model.

4.9 Reward Expectation in the Motor Cortex during an Observation Task

In this section, we analyze an experiment involving a variable reward task. The motivation of using the shift-varying decompositions on this task is to identify characteristics that consistently distinguish the evoked potentials of rewarding and

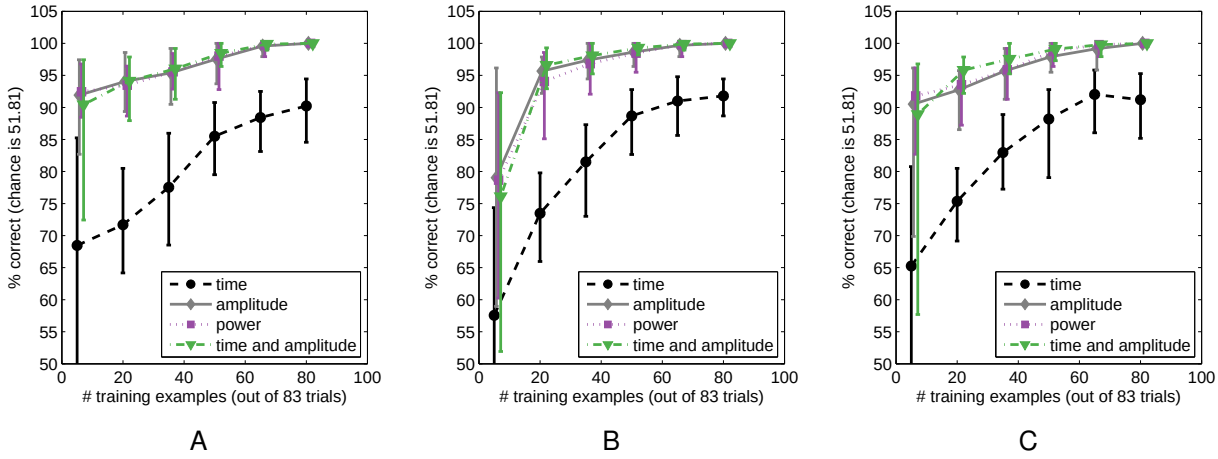


Figure 4-9. Nearest-neighbor classifier performance using different features while varying training set size. The error-bars correspond to the average, 5th percentile, and 95th percentile across 20 Monte Carlo divisions of the samples into testing and training sets. A) Model trained to explain the maximum energy during the non-rewarding trials. B) Model trained to be spatially discriminative. C) Model uses a temporal subspace to explain the maximum energy during the non-rewarding trials.

Table 4-1. Classification performance for decoding object type across features and models.

Model	Time	Amplitude	Power	Time and amplitude
A	71.7±4.7	94.0±2.8	93.5±2.4	94.1±3.3
B	73.5±4.0	95.7±1.8	94.0±4.0	96.6±2.0
C	75.4±3.5	92.8±2.8	93.6±2.5	95.8±1.8

Entries indicate the mean and standard deviation of accuracy (% correct) computed across 20 Monte Carlo divisions of the trials into 20 for training and 63 for testing, using nearest-neighbor for classification. Chance rate is 52%. Model A is trained to explain the maximum energy during the non-rewarding trials. Model B is trained to be spatially discriminative. Model C uses a temporal subspace to explain the maximum energy during the non-rewarding trials.

non-rewarding trials. In addition, for dataset both LFPs and action potential timings were collected, and we investigate any corresponding between the timing extracted from the LFPs and the action potentials.

4.9.1 Data Collection

This data is from a single day's recording collected in Joseph Francis's laboratory at SUNY-Downstate by Brandi Marsh. There were 96 channels of neural data collected

from a macaque monkey. The spiking units were identified on the 96 channels. LFPs were collected on 32 electrodes implanted in M1.

On this dataset, 8 of the LFP electrodes were excluded because of artifacts, which were associated with high impedance. The remaining 24 channel signal was high-pass filtered with a cutoff of 4 Hz and notch filtered at 60, 120, 180, 240, and 300 Hz with a quality factor of 69. The data was further filtered with a Gaussian window with a length of 40 samples and standard deviation of 8 samples. This corresponds to a low-pass filter with a cutoff of 66.4 Hz.

During the experiment, the subject manually performed right-handed center-out reaching tasks using a Kinarm exoskeleton (BKIN Technologies) in a two-dimensional plane, while the left arm was restrained with a cuff. Visual feedback was provided by cursors representing the position of the tip of the subject's middle finger and targets were displayed to the subject. Targets were 4-5 cm from the center target and the targets were 1 cm in diameter. The experiment consisted of watching the center target for 300 ms until the peripheral target appeared and center target disappeared.

During certain sessions, only the target to the right of the center is cued. In addition, reward variability was introduced: only a subset of successful trials would end in a liquid reward—the unsuccessful reach trials never resulted in reward. Both center and reach targets changed color during the hold period to indicate whether the trial outcome would be rewarding or non-rewarding. The subject could learn to associate this color cue with the expected reward outcome. Unsuccessful trials were repeated, so the subject was motivated to complete all trials. The presentation of reward outcomes were randomized. The hypothesis is that the expected reward, which the subject determined by the color cue, could be determined solely from the neural response.

This task was further abstracted by automating the cursor movement to the right target at a speed of 1 cm/s, such that subject had to only observe the movement. Eye tracking was performed using an IR sensitive camera and trials were aborted if gaze

was not maintained. During this modified task, the subject's right arm, its reaching arm, was locked in place by securing the Kinarm exoskeleton and the left arm was still secured in a cuff.

Here we analyze a dataset for a single session. There were 289 trials in total; 145 rewarding trials and 144 non-rewarding trials. We use the LFPs in the 2 s window following reach target appearance, and spike data 2.5 s before and 3 s after this cue.

4.9.2 Spatiotemporal Model

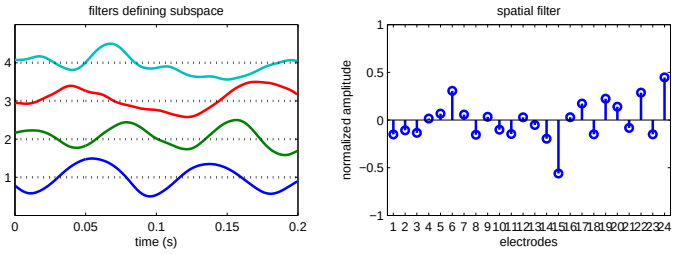


Figure 4-10. Spatiotemporal models trained during reward expectation. A set of 4 temporal filters are trained to span the subspace with the maximum energy during the non-rewarding trials. A single spatial filter is trained to be discriminative using the spatial covariance under both conditions.

A single spatiotemporal model was estimated using two-thirds of the trials. The model uses a subspace of 4 temporal waveforms to approximate each non-rewarding trial and the spatial factor is discriminatively trained with the spatial covariance estimated during rewarding trials. The resulting filters corresponding to the temporal and spatial factors are shown in Figure 4-10. Example trials and the models approximations are shown in Figure 4-11. The distribution of the subspace magnitude and the temporal alignment for the training and testing sets are shown in Figure 4-12. The two conditions are separable by their magnitude, with larger amplitude during the non-rewarding trials (as is intended by the modeling). The timing does not show a clear pattern; however, we explore using the timing extracted from the LFP event as a timing event for aligning histograms of the neural spike trains.

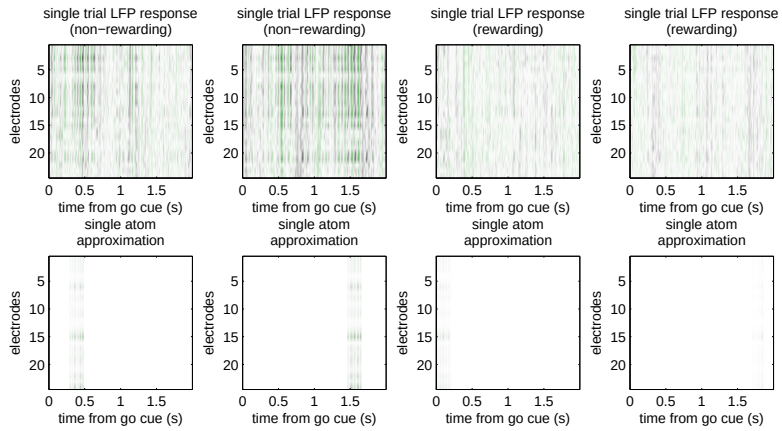


Figure 4-11. Multichannel LFPs from the motor cortex and rank-1, single-atom approximations of 4 example trials, 2 from each reward condition.

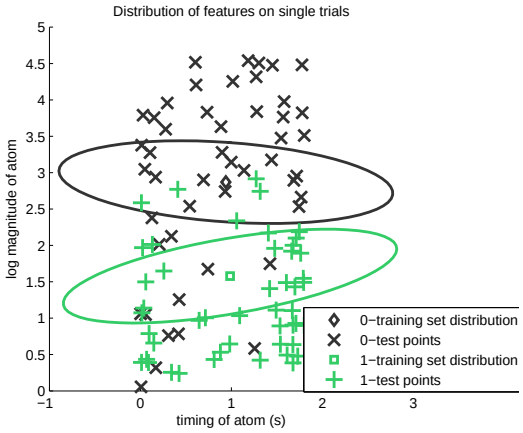


Figure 4-12. Parameter distribution during reward expectation. Timing and magnitude scatter plot for atomic decompositions of test set examples and normal distributions fitted to the training set.

4.9.3 Peristimulus Time Histograms Aligned to LFP Events

The peristimulus time histogram (Gerstein & Kiang, 1960) is an estimator for the time-varying firing rate during trials. The use of histogram assumes a time-locked, inhomogeneous Poisson point process (Brown et al., 2002) where spike timings in a given trial are independent. We seek to test the hypothesis that the timings aligned to the endogeneous LFP-event are a better model for individual neural firing rate than the original cue timing. We form two models: one that realigns the spike-times to the LFP-event and another that uses the original timings relative to the cue. Alternatively, we

could form a nested model using both timings (Park et al., 2011), but here the disparate models with separate dependent variables has a clearer understanding.

We extracted the spike data 2.5 s before and 3 s after the reach target cue. Under the first model we form a fixed width histogram for the spikes in the 2 s window following the cue, and under the second model we form a fixed width histogram for the spikes 1 s before and 1 s after the LFP event. For either model, any spikes that fall before these 2 s windows are pooled into a single bin, a separate bin is used for any spikes that follow after the window. The number of bins in the 2 s window is selected for each neuron and condition based on minimizing a cost function based on the mean and variance of the histogram counts (Shimazaki & Shinomoto, 2007). Specifically, for each spiking unit and condition the bin size is selected as the minimum of the optimal bin size across both models. The actual search is performed on the number of bins, ranging from 1 to 200, and the bin width is chosen to equally divide the 2 s window. If the optimal bin size was more than 250 ms that condition/unit was excluded from further analysis. This yielded 50 units under non-rewarding and 57 units under rewarding.

Given the binning time structure, the spike train recorded on a given trial are converted into a vector of non-negative integers. The elements of this vector are assumed to be independent but Poisson distributed with a time-varying mean. To compare the two time alignments, we perform leave-one-out cross validation of goodness-of-fit using log-likelihood and check for a significantly better fit under one of the two time-structure models using Vuong's test (Vuong, 1989). The test is simply a t-test on the paired log-likelihood values across all trials for the two models. For each condition, Bonferroni's correction was applied to the p-values to account for the multiple tests across neurons. A significance level was chosen at 1%.

There were a total of 7 units—all under the rewarding condition—for which the original cue-time aligned model provides a significantly better fit (Vuong's test, $p < 10^{-5}$). There was a total of 3 units—2 under the non-rewarding condition and 1 under the

rewarding condition for which the LFP-event aligned model provides a significantly better fit. One of these units fit significantly better to the cue-time aligned model for rewarding conditions and the LFP-event aligned model for non-rewarding trials, and the unit's spike trains and time histograms are shown in Figure 4-14. The remaining units' spike trains and histograms which had significantly better fit under the LFP-event aligned PSTH are shown in Figure 4-13. Table 4-2 summarizes the number of units assigned to each model per condition.

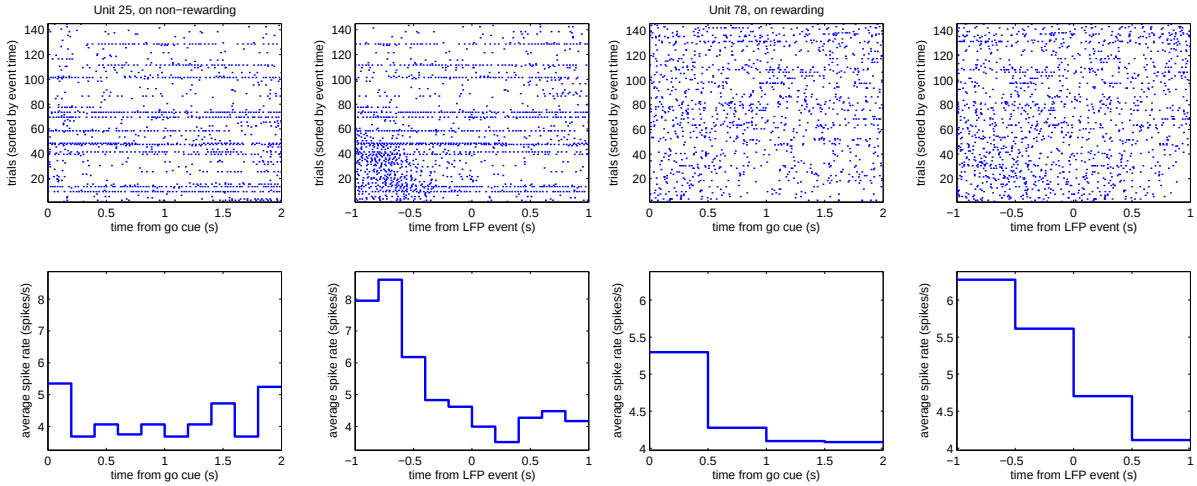


Figure 4-13. Spike trains and histograms for spiking units for which the LFP-event aligned provides a significantly better fit (Vuong's test, $p < 10^{-5}$). A total of 3 units were identified—2 under the non-rewarding condition and 1 under the rewarding condition—2 units are shown here and the remaining unit is shown in Figure 4-14.

Table 4-2. Number of units per condition with statistically better fits for each model

Condition	Number of units below bin width threshold	LFP-event aligned model	Cue aligned model
Non-Rewarding	50	3	0
Rewarding	57	1	7

The results of the statistical test appears consistent with the modeling structure, as the LFP-event is based on a model of the non-rewarding conditions. It is interesting that the two units assigned to the LFP-event model under the non-rewarding condition have

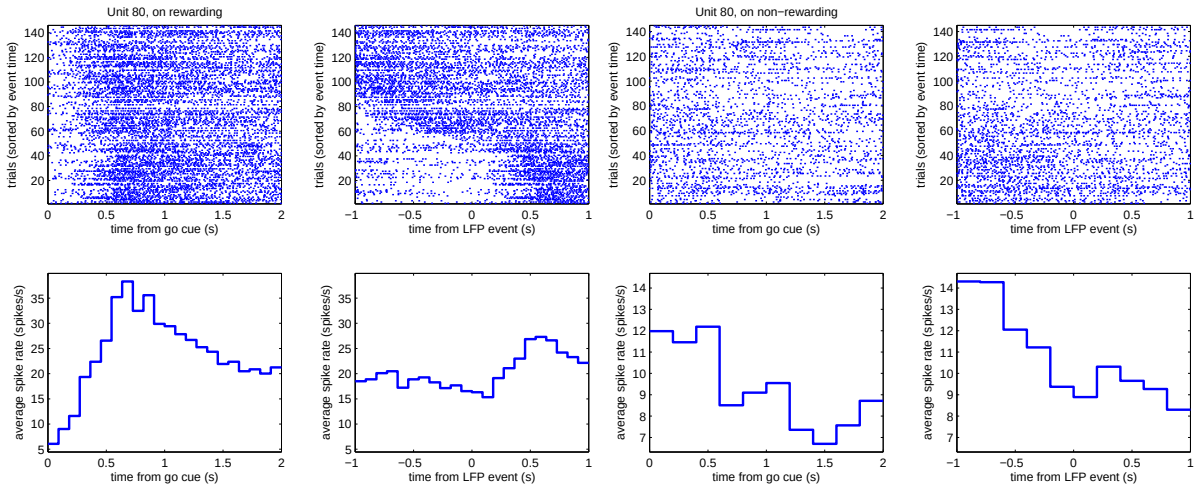


Figure 4-14. A unit that fit significantly better to the cue-time aligned model for rewarding conditions and the LFP-event aligned model for non-rewarding trials (Vuong's test, $p < 10^{-5}$).

spike rates that increase preceding the LFP-event. This points to the fact that the spiking may be an indication of the upcoming LFP-event.

4.9.4 Discussion

Whether Vuong's test is the best approach for comparing models of spike trains is doubtful since the log-likelihood values are not normally distributed for low-spike counts. For instance on trials without spikes the log-likelihood is exactly the same. We explored other tests for non-nested models such as using a non-parametric sign-test on the paired log-likelihood values (Clarke, 2007). Using different tests there were a larger number of significant units; however, on inspection many were very low firing rate (less than 1 Hz), and the models fitting appeared tenuous. This invites the opportunities for future investigation into the best way to compare non-nested models for spike trains.

4.10 Model Selection

In the previous result sections, the model architecture for the evoked potentials were chosen *a priori*; the optimality of the model and model order were not considered. In this section, model selection is considered for the previously analyzed datasets along with the data from two subjects in the publicly available BCI Competition III (Blankertz et al.,

2006) dataset II provided by the Wadsworth Center, NYS Department of Health, and collected by Jonathan R. Wolpaw, Gerwin Schalk, and Dean Krusienski. Specifically, the two human subjects were using a P300 speller interface within the BCI2000 software (Schalk et al., 2004).

The aim of this analysis is to compare models with different underlying assumptions regarding the complexity of the structure in the neural response set. The following models were compared using different sizes of shiftable temporal waveforms and different ranks or number of components:

- A bilinear approximation of the spatiotemporal average, i.e., the average across trials is approximated using the singular value decomposition with different ranks.
- Woody's method for estimating an average temporal waveform that is allowed to shift between trials (Woody, 1967).
- A shift-varying subspace approach proposed here: this is an extension of an amplitude and shift-varying waveform (Jaskowski & Verleger, 1999; Truccolo et al., 2003).
- The differentially variable component analysis (dVCA) (Truccolo et al., 2003) with a single spatial factor and run for a fixed number of iterations (11 full cycles).
- A greedy version of dVCA, (gVCA), that we implemented is also used. The original version of dVCA used an alternating optimization across components which does not always converge. Here we replaced this with a stage-wise greedy approximation wherein a component is completely learned before moving on to the next component. In the proposed greedy version, an explicit spatial factors for each component can be naturally added; whereas, only a single spatial factor is used in the alternating version.
- The canonical polyadic decomposition (CPD) with varying ranks that models the trial-varying spatiotemporal waveforms, but does not account for any shifts. The CPD is learned using a fast damped Gauss-Newton (Levenberg-Marquad) algorithm (Phan et al., 2013a) using publically available implementation (Phan et al., 2013b), which utilizes the MATLAB Tensor Toolbox (Bader et al., 2012).
- The raw spatiotemporal average, or trial-wise principal component analysis, where the spatiotemporal waveform is vectorized, have orders of magnitude more parameters are not considered because of their inefficiency.

The Bayesian information criterion (BIC) (4–12) is used to compare the different models and model orders in terms of mean squared error and degrees of freedom. For each model, different lengths were used for the temporal waveform and different number of components or ranks were used for the spatial and trial-varying components. The number of degrees of freedom used in calculating BIC takes into account both the number of trial-varying parameters (e.g., the trial amplitude, temporal alignment) along with the number of trial-invariant parameters (e.g., the number of spatial factors and the length of the temporal waveform).

A comparison of the models was performed on three datasets:

- LFPs recorded from the ventral striatum in a marmoset for the non-rewarding object trials (Section 4.8)
- LFPs recorded from the motor cortex in a macaque for a subset (the first 48 trials) of the non-rewarding passive observation trials (Section 4.9)
- EEGs recorded across the scalp for two human subjects (subject A and B) during a P300 evoked potentials experiment, the first 4 highlighted responses to the desired character across all 85 character presentations in the BCI Competition III Wadsworth BCI dataset (Blankertz et al., 2006)

For each model and dataset, the following values were calculated: the normalized mean squared error, the degrees of freedom, the BIC value, and the computation time. The results across all models are shown in Figure 4-15 for the LFP datasets and Figure 4-16 for the EEG datasets. For the models compared, the overall trend is that increasing parameters yields better performance. The fact that BIC has not reached a minimum for any model type indicates that the data supports a more complex structure than the simple models are able to provide.

To better understand the model performance with a interpretable number of components, a subset of the models with a fixed number of components are compared. For the shift-varying model the length of the temporal waveform that yields the lowest BIC (closer to optimal) is chosen. The results for 2 and 8 components are presented in Table 4-3 and Table 4-4, respectively.

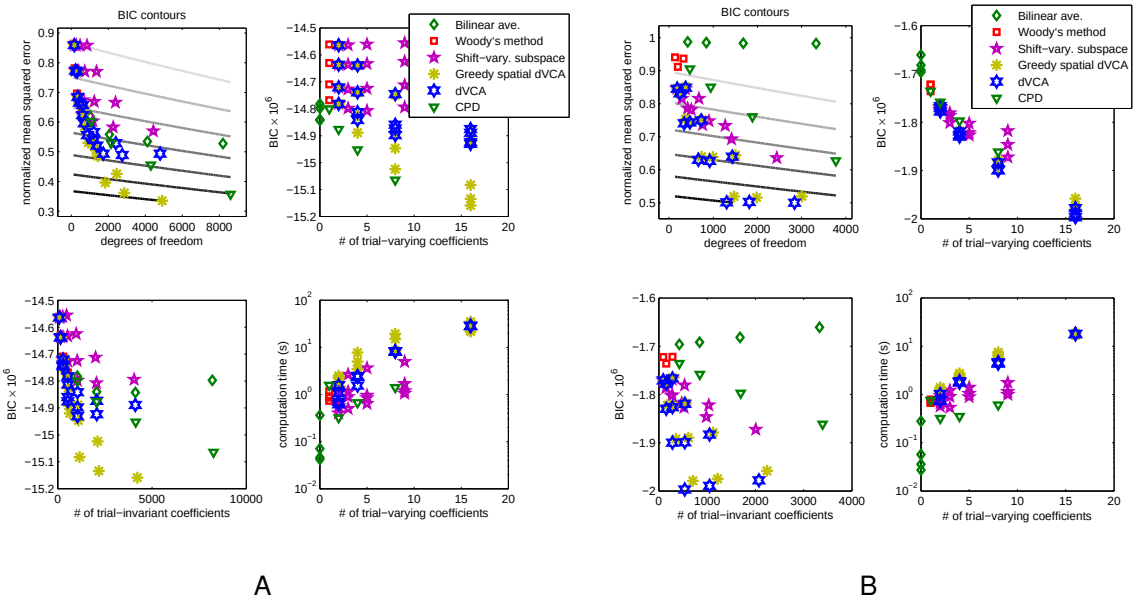


Figure 4-15. Model performance across local field potential datasets. For each model type, models with various number of parameters are compared. A) LFPs in ventral striatum for non-rewarding object trials. B) LFPs in motor cortex for non-rewarding passive observation trials.

The dVCA model, under both optimization methods, offers the optimal model, in terms of BIC, for a fixed number of components: outperforming the tensor model and the single component with a temporal subspace. For 2 components, the original alternating optimization performed best on 3 out of the 4 subjects, and the greedy optimization was the best for other dataset. For 8 components, the greedy optimization performed best on 3 out of the 4 subjects and the original alternating optimization was the best for other dataset. This switch may have been due to the fact that the alternating optimization was ran for a fixed number of iterations, and with a larger number of components the number of iterations needs to increase. However, it should be noted that even with a fixed number of iterations the run time for the dVCA models was substantially greater than the shift-varying subspace or tensor decompositions.

The results indicate there is a clear trade-off between performance and computation time. In addition, the dVCA model is more difficult to interpret since multiple temporal

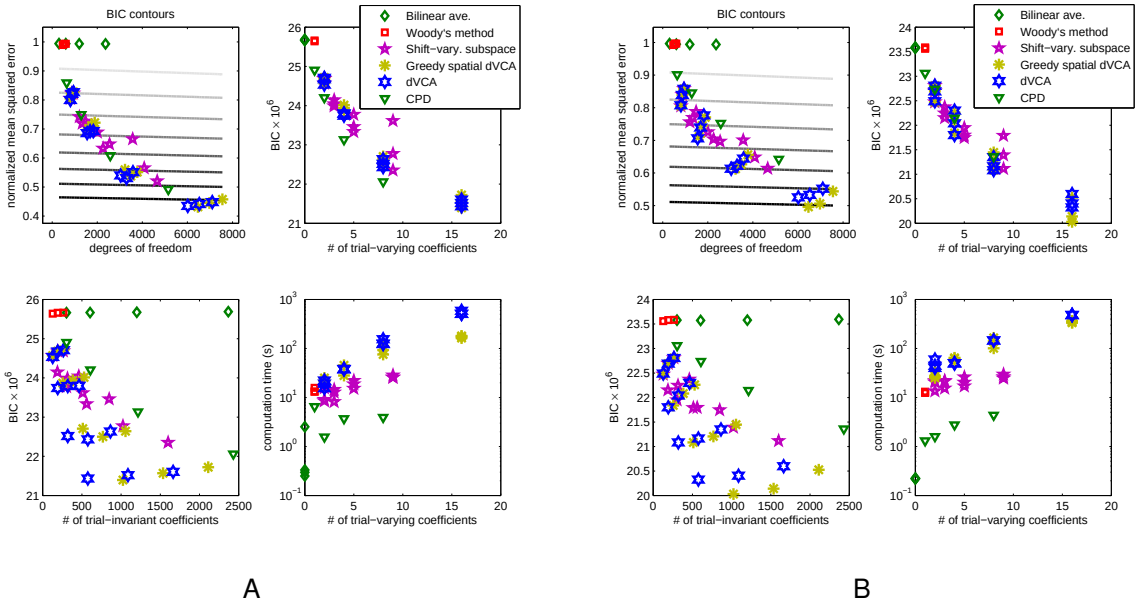


Figure 4-16. Model performance on P300 EEG data on two subjects in the BCI competition III Wadsworth dataset. For each model type, models with various number of parameters are compared. A) Subject A. B) Subject B.

shifts are estimated for each trial. Whether or not the increased explanatory power of dVCA would translate into discrimination as was seen in the shift-varying subspace is yet to be determined.

4.11 Summary

The models for evoked potentials have been guided by hypothesis of the neural responses, and have been constrained to trial-varying parameters that are easy to understand such as amplitude and temporal alignment. This single-trial extraction of these parameters allows testing for trends across trials. Here we have used existing models and proposed new models for which the trial-varying parameters are treated as features for single-trial classification. The models follow the lines of previous research for varying temporal alignment (Jaskowski & Verleger, 1999; Truccolo et al., 2003; Woody, 1967), and included spatial factors (Li et al., 2009; Rivet et al., 2009; Souloumiac & Rivet, 2013). In addition, in one instance the spatial factor was trained to act as a filter that explicitly discriminates between the conditions (Ramoser et al., 2000).

Table 4-3. Model performance using 2 components for each dataset.

Dataset[a]	Dimension	Model[b]	Waveform length	BIC $\times 10^6$	SNR (dB)	Run time (s)
A	16×1017×43					
		CPD	1017	-14.88	6.36	0.32
		SVS	512	-14.80	5.06	2.66
		dVCA	512	-14.84	5.69	2.47
		gVCA	512	*-14.89	6.36	7.77
B	24×400×48					
		CPD	400	-1.76	1.62	0.32
		SVS	128	-1.80	2.40	1.21
		dVCA	64	*-1.83	3.00	1.79
		gVCA	128	-1.83	2.95	2.49
C	64×240×340					
		CPD	240	24.21	2.88	1.57
		SVS	12	23.99	3.29	12.77
		dVCA	64	*23.74	3.77	37.63
		gVCA	64	23.87	3.54	28.09
D	64×240×340					
		CPD	240	22.74	1.68	1.60
		SVS	64	22.15	2.80	22.04
		dVCA	64	*21.80	3.48	50.12
		gVCA	64	21.84	3.40	62.62

* Indicates best model with 2 components in terms of BIC.
[a]Datasets: A) LFPs in ventral striatum during non-rewarding object trials. B) LFPs in motor cortex for non-rewarding passive observation trials. C) Scalp EEGs for subject A during BCI operation. D) Scalp EEGs for subject B during BCI operation.
[b]Models consist of the canonical polyadic decomposition (CPD), shift-varying subspace (SVS) with a single spatial factor, differentially variable component analysis (dVCA) with a single spatial factor, and the greedy version (gVCA) that has a unique spatial factor for each component.

As opposed to the majority of the existing literature, we have explicitly considered the spatiotemporal neural signals across the trials as a tensor.

Tensor decompositions are a natural model for evoked potentials (Acar et al., 2007; Cichocki et al., 2008; Dauwels et al., 2011). Here the use of shift-varying models has been combined with the tensor models using explicit estimation of shifts. This does not require approximations restricted to low-frequency signals (Mørup et al., 2008): an approximation that is used by shift-varying models for evoked potential models (Jaskowski & Verleger, 1999; Pham et al., 1987; Weeda et al., 2012). We introduced a shift-varying model that does not require this approximation, and uses a temporal

Table 4-4. Model performance using 8 components for each dataset.

Dataset[a]	Dimension	Model[b]	Waveform length	BIC $\times 10^6$	SNR (dB)	Run time (s)
A	16×1017×43					
		CPD	1017	-15.06	10.29	1.41
		SVS	512	-14.79	5.63	4.97
		dVCA	128	-14.93	7.05	28.32
		gVCA	512	*-15.16	10.93	35.18
B	24×400×48					
		CPD	400	-1.86	4.66	0.61
		SVS	256	-1.87	4.52	1.76
		dVCA	64	*-2.00	6.90	17.87
		gVCA	64	-1.98	6.56	18.53
C	64×240×340					
		CPD	240	22.06	7.10	3.94
		SVS	64	22.35	6.53	28.47
		dVCA	64	21.43	8.33	513.62
		gVCA	64	*21.40	8.41	180.36
D	64×240×340					
		CPD	240	21.36	4.43	4.37
		SVS	200	21.12	4.88	24.70
		dVCA	64	20.32	6.44	480.10
		gVCA	64	*20.03	7.02	338.37

* Indicates best model with 8 components in terms of BIC.
[a]Datasets: A) LFPs in ventral striatum during non-rewarding object trials. B) LFPs in motor cortex for non-rewarding passive observation trials. C) Scalp EEGs for subject A during BCI operation. D) Scalp EEGs for subject B during BCI operation.
[b]Models consist of the canonical polyadic decomposition (CPD), shift-varying subspace (SVS) with a single spatial factor, differentially variable component analysis (dVCA) with a single spatial factor, and the greedy version (gVCA) that has a unique spatial factor for each component.

subspace to explain each single-trial waveform. After alignment of each trial, the model is a block term decomposition (De Lathauwer, 2008).

The tensor treatment also allows standard tensor decompositions to be compared to the shift-varying methods using standard model selection criterion. Through this analysis, it was confirmed that the additional computational cost of shift-varying decompositions is offset by increased model flexibility. From the results, it is clear that the shift-varying models are able to more parsimoniously explain the data: they have fewer coefficients, better fit, and are still more interpretable than standard tensor decompositions.

The single-trial modeling and classification was applied to two reward expectation datasets. For the dataset with both neural action potentials and LFPs, we were able to evaluate the coupling between these scales. Specifically, models of the neural firing rate aligned to the experimental timing are compared to models aligned to the endogenous timing of the evoked potential. Cross-validation is used to select which time events better explain the neural activity in terms of a simple peristimulus time histograms (PSTH) model for the firing rate (Gerstein & Kiang, 1960).

This study serves to bring together a number of current trends in the evoked potential research. Throughout, we have used real datasets to motivate the methodology for shift-tolerant tensor models. The shift-tolerant models use the explicit timing and amplitude distribution of spatiotemporal waveforms that reoccur across trials. Overall, these efforts attempt to form better instantaneous measures of the characteristics of the brain: from which can hope to capture and understand the ongoing processing involved in cognition.

CHAPTER 5
NEURAL DECODING WITH KERNEL-BASED METRIC LEARNING

When studying the nervous system, the choice of metric for the neural responses is a pivotal assumption. For instance, a well-suited distance metric enables us to gauge the similarity of neural responses to various stimuli and assess the variability of responses to a repeated stimulus—exploratory steps in understanding how the stimuli are encoded neurally. Here we introduce an approach where the metric is tuned for a particular neural decoding task. In particular, neural spike train metrics have been used to quantify the information content carried by the timing of action potentials. While a number of metrics for individual neurons exist, a method to optimally combine single-neuron metrics into multi-neuron, or population-based, metrics is lacking.

Metric-learning algorithms change the metric in the original space to achieve some specific objective, usually exploiting supervisory information (Fukumizu et al., 2004; Lowe, 1995; Xing et al., 2003). When the original space is composed of multiple independent dimensions, some possible ways to change the metric are by removing dimensions, scaling dimensions, or rotating the axes of the space by using linear combinations of dimensions. A linear projection of the dimensions is optimized by a number of supervised learning methods. Removing or weighting features corresponds to feature selection.

Metric learning is used as an intelligent preprocessing for classification methods that depend on a measure of similarity or dissimilarity to determine if two samples are of the same class. For instance, kernel machines or nearest-neighbor-based approaches compare novel samples relative to already observed samples but rely on a predefined similarity measure. These classifiers are highly dependent on the preprocessing and

---

Portions of this Chapter were submitted for publication in the following manuscript: Brockmeier, A. J., Choi, J. S., Kriminger, E. G., Francis, J. T., and Principe, J. C. (2014). Neural decoding with kernel-based metric learning. *Neural Computation, 26*(6), *in press.*

offer little insight into the importance of individual feature dimensions. Metric learning can improve the classification performance by adjusting the importance of individual features for the task (Lowe, 1995; Takeuchi & Sugiyama, 2011), and these weights can be used to highlight the features or dimensions relevant for the objective. Furthermore, metric learning approaches can also improve kernel regression (Fukumizu et al., 2004; Navot et al., 2006; Weinberger & Tesauro, 2007).

We investigate classes of metrics that are parametrized along spatiotemporal dimensions of neural responses. For instance, it is natural to consider which channels or time points in the neural response are most useful for distinguishing among conditions. Unlike previous metric-learning approaches that have concentrated on learning projections and weightings for scalar-valued variables, here we also explore using metric learning where the weights correspond to different neurons in multiunit spike train metrics or vectors of spatial amplitudes from multichannel LFPs.

With vector-valued data each individual metric is defined over a vector space. Using a weighted combination of these metrics we can form strictly spatial or temporal weightings for multivariate time series. In addition, we propose to optimize multi-neuron spike train metrics (Aronov, 2003; Houghton & Sen, 2008) formed as combinations of spike train metrics defined for individual neurons (Paiva et al., 2009; van Rossum, 2001; Victor & Purpura, 1996). To our knowledge, ours is the first attempt to explicitly optimize the parameters of multi-neuron spike train metrics, instead of using pre-defined weightings.

Given the form of the projections or weightings, one must consider their optimization. A number of metric-learning cost functions have been posed in the literature, but we propose using a kernel-based measure of dependence known as centered alignment (Cortes et al., 2012). Centered alignment was shown to be a useful measure for kernel-learning (Cortes et al., 2012), and a similar but unnormalized kernel dependence measure, Hilbert Schmidt information criterion (HSIC), has been used for feature

selection (Song et al., 2012). Another kernel-based dependence measure formulated based on conditional entropy (Sanchez Giraldo & Principe, 2013) has also been shown to be useful for learning a Mahalanobis distance (Brockmeier et al., 2013c; Sanchez Giraldo & Principe, 2013) and a weighted product kernel (Brockmeier et al., 2013a). The objective and optimization techniques used here are most similar to those proposed by Fukumizu et al. (2004), but by replacing the kernel-based canonical correlation measure (Bach & Jordan, 2003) with centered kernel alignment we avoid both matrix inversion and regularization.

Using the kernel-based objective, we highlight the connection between optimizing weighted tensor-product kernels and metric learning. Optimizing a metric in a kernel-based framework has the added benefit that it naturally optimizes the kernel itself for use in support vector machines. This eliminates the need for the user to choose a kernel size through cross-validation or trial-and-error. Kernels also provide a straightforward way to form metrics corresponding to nonlinear projections. This is done by retrieving a metric from a unweighted sum of optimized kernels—an approach distinct from optimizing a convex sum of kernels (Lanckriet et al., 2004). Ultimately, this leads us to propose optimized multi-neuron spike train kernels formed as the product and the sum of product of single-unit spike train kernels (Paiva et al., 2009; Park et al., 2013, 2012).

Using the kernel-based dependence measure we optimize multi-neuron metrics and metrics on local field potentials (LFPs). The approach is demonstrated on invasively recorded neural data consisting of both spike trains and LFPs. The experimental paradigm consists of decoding the location of tactile stimulation on the forepaws of anesthetized rats. We show that the optimized metrics highlight the distinguishing dimensions of the neural response, significantly increase the decoding accuracy, and improve non-linear dimensionality reduction methods for exploratory neural analysis.

The rest of the chapter is organized as follows: we first introduce the mathematical representation of the neural response and different metrics that use linear projections

or weightings; from the metrics we form kernel-based similarity measures; from the kernels we introduce the dependence measure (Cortes et al., 2012); and from these we form metric-learning optimization problems. We verify the classification performance of the proposed approach on benchmark datasets, and show results on the experimental datasets. We conclude with a discussion of the results, the connection of metric-learning to neural encoding, and future applications.

5.1 Metrics and Similarity Functions

5.1.1 Neural Data Representation and Metrics

For the trial-wise classification of different conditions, a sample from each trial is the concatenation of the neural response across all selected time samples, electrode channels, or neural spiking units. Let $x = [x_{(1)} \dots x_{(P)}]$ denote the P-dimensional neural response to a given trial, where parenthetical subscripts denote the response dimension: $x_{(i)}$ may be a scalar, vector, or set (in the case of spike trains). Let x_j denote the neural response for the jth trial, $j \in \{1, \dots, n\}$, and let $l_j \in \{1, \dots, m\}$ denote the discrete class label corresponding to a certain class condition for the jth trial. The set $\{z_j = (x_j, l_j)\}_{j=1}^{n}$ represents a joint sample of the neural responses and labels.

A bivariate function $d(\cdot, \cdot)$ is a distance metric with domain $\mathcal{X} \times \mathcal{X}$, if it satisfies the following requirements:

1. $d(x, x') \geq 0 \quad \forall x, x' \in \mathcal{X}$
2. $d(x, x') = d(x', x) \quad \forall x, x' \in \mathcal{X}$
3. $d(x, x') = 0$ if and only if $x = x'$
4. $d(x, x^+) \leq d(x, x') + d(x', x^+) \quad \forall x, x', x^+ \in \mathcal{X}$

Functions that satisfy all but the third requirement are considered pseudo-metrics. In metric learning, achieving a pseudo-metrics may be useful since it allows points to be considered equivalent even if they differ on certain, hopefully irrelevant, aspects.

Consider the distances between pairs of neural responses along each dimension: the distance between samples x, x' on the ith dimension is denoted $d_i(x_{(i)}, x'_{(i)})$. For

instance, this may be the Euclidean metric for scalars or vectors or a metric on spike trains (Dubbs et al., 2010; Paiva et al., 2009; van Rossum, 2001).

A metric for the joint neural response is formed by combining these individual distances using a feature weighting (Lowe, 1995; Takeuchi & Sugiyama, 2011), where the weights control the importance of the distance along each dimension. Let w denote a nonnegative P-dimensional weighting vector, such that $\forall i, w_i \geq 0$. The metric using this weighting is formed as

$$d_w^\gamma(x, x') = \sum_{i=1}^{P} w_i d_i^\gamma(x_{(i)}, x'_{(i)}), \tag{5–1}$$

where the exponent $\gamma \geq 1$ controls how relatively large distances on individual dimensions contribute to the total distance. This is a weighted Euclidean metric if $\gamma = 2$ and the metric for each dimension is the Euclidean or $\mathcal{L}_2$ metric.

If $w_i = 0$ then the metric is actually a pseudo-metric since it does not satisfy the property that $d(x, x') = 0$ if and only if $x = x'$. However, this invariance to certain dimensions is a goal of metric learning. For vector-valued data the weighting is a special case of a linear transformation that is equivalent to globally scaling the input dimensions: $w_i d_i^\gamma(x_{(i)}, x'_{(i)}) = d_i^\gamma(w_i^{1/\gamma} x_{(i)}, w_i^{1/\gamma} x'_{(i)})$.

If each neural response is a vector of scalars, we can define a more general Euclidean metric parametrized by a matrix $A \in \mathbb{R}^{P \times Q}$ using a linear projection of the samples $y = A^\mathsf{T} x, \quad y_{(j)} = \sum_i A_{i,j} x_{(i)}, \ j = 1, \ldots, Q$. The Euclidean distance in the Q-dimensional feature space $d^2(y, y') = \sum_{j=1}^{Q} \|y_{(j)} - y'_{(j)}\|_2^2$ is equivalent to a Mahalanobis distance in the original space, with the inverse covariance matrix replaced by the symmetric positive definite matrix AA^T:

$$d_A^2(x, x') = d^2\left(A^\mathsf{T} x, A^\mathsf{T} x'\right) = \left\|A^\mathsf{T} x - A^\mathsf{T} x'\right\|_2^2 = (x - x')^\mathsf{T} A A^\mathsf{T} (x - x'). \tag{5–2}$$

As A has $P \cdot Q$ coefficients there are more degrees of freedom to distort the space according to this metric. This matrix is strictly positive definite if $P = Q$ and AA^{T} is full rank; otherwise, the metric is actually a pseudo-metric.

The special case of a weighted metric (5–1) appears if A is diagonal and square, with P diagonal entries $A_{i,i} = \sqrt{w_i}$. More generally, a Mahalanobis distance can be seen as a weighting over a squared distance matrix between all dimensions. Using properties of the trace,

$$d_A^2(x, x') = \mathrm{tr}\left[AA^{\mathsf{T}}(x - x')(x - x')^{\mathsf{T}}\right] = \mathrm{tr}\left[AA^{\mathsf{T}}D\right] = \sum_{i,j}[AA^{\mathsf{T}}]_{i,j}D_{i,j} \tag{5–3}$$

where $D_{i,j} = \|x_{(i)} - x_{(j)}\|_2^2 = \langle x_{(i)}, x_{(j)}\rangle - \langle x_{(i)}, x'_{(j)}\rangle - \langle x'_{(i)}, x_{(j)}\rangle + \langle x'_{(i)}, x'_{(j)}\rangle$. Unless A is diagonal this metric exploits the inner-product between different dimensions. Written in this form, it is clear how a Mahalanobis-type metric can be formed whenever all the dimensions of the neural response correspond, or can be mapped, to the same Hilbert space. Specifically, let $\phi : \mathcal{X} \to \mathcal{H}$ define a mapping from any element $x_{(i)} \in \mathcal{X}$ to an element in the Hilbert space $\phi(x_{(i)}) \in \mathcal{H}$. The Mahalanobis-type metric in this space is defined by Equation (5–3) where $D_{i,j} = \langle\phi(x_{(i)}), \phi(x_{(j)})\rangle - \langle\phi(x_{(i)}), \phi(x'_{(j)})\rangle - \langle\phi(x'_{(i)}), \phi(x_{(j)})\rangle + \langle\phi(x'_{(i)}), \phi(x'_{(j)})\rangle$. As long as the inner-product can be defined between the dimensions, for instance by using the spike-train kernels discussed in Section 5.1.3, one can form metrics that use the distance between different spiking units. This would replicate the interaction between spikes on different units intrinsic to some multi-unit metrics (Aronov, 2003). However, evaluating the inner-product between each pair of dimensions for every pair of samples is computationally demanding, and is not investigated here.

5.1.2 Kernels

Kernel functions are bivariate measures of similarity based on the inner-product between samples embedded in a Hilbert space. Let the domain of the neural response be denoted $\mathcal{X}$ and consider a kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. If κ is positive definite then there is

an implicit mapping $\phi : \mathcal{X} \to \mathcal{H}$ that maps any element $x \in \mathcal{X}$ to an element in the Hilbert space $\phi(x) \in \mathcal{H}$ such that $\kappa(x, x') = \langle \phi(x), \phi(x') \rangle$.

As we want to explore the similarity across the individual dimensions of the data, we compose a joint similarity measure from the marginal similarity on each dimension. Let $\mathcal{X}_i$ denote the neural response domain of the ith dimension and consider a positive-definite kernel $\kappa_i : \mathcal{X}_i \times \mathcal{X}_i \to \mathbb{R}$ and corresponding mapping $\phi_i : \mathcal{X}_i \to \mathcal{H}_i$ for this dimension. The similarity between a pair of samples x and x' on the ith dimension is $\kappa_i(x_{(i)}, x'_{(i)}) = \langle \phi_i(x_{(i)}), \phi_i(x'_{(i)}) \rangle$.

5.1.2.1 Tensor-product kernel

The joint similarity over both dimensions i and j is computed by taking the product between the kernel evaluations $\kappa_{[ij]}(x_{(i,j)}, x'_{(i,j)}) = \kappa_i(x_{(i)}, x'_{(i)}) \cdot \kappa_j(x_{(j)}, x'_{(j)})$. The new kernel $\kappa_{[ij]}$ is called a tensor-product kernel since it corresponds to using a mapping function that is the tensor product between the individual mapping functions $\phi_{[ij]} = \phi_i \otimes \phi_j$ where $\phi_{[ij]}(x_{(i,j)}) \in \mathcal{H}_{[ij]}$. The product of positive-definite kernels is positive definite, and taking the product over all dimensions returns a positive-definite kernel over the joint space: $\kappa(x, x') = \prod_i \kappa_i(x_{(i)}, x'_{(i)})$.

Due to the product, if for one dimension $\kappa_i(x_{(i)}, x'_{(i)}) \approx 0$ then $\kappa(x, x') \approx 0$. If some of the dimensions are noisy with respect to the task, then they will have a deleterious effect on the joint similarity measure. In order to separately weight the contribution of each dimension in the product, consider taking the kernel for the ith dimension to the $\theta_i \geq 0$ power $[\kappa_i(x_{(i)}, x'_{(i)})]^{\theta_i}$. As $\theta_i \to 0$ the influence of ith dimension decreases, and $\theta_i = 0 \implies (\kappa_i(x_{(i)}, x'_{(i)}))^{\theta_i} = 1$, thereby removing its effect altogether. Taking the product over all dimensions results in a weighted product kernel over the joint space:

$$\kappa_\theta(x, x') = \prod_i \left[\kappa_i(x_{(i)}, x'_{(i)})\right]^{\theta_i}, \tag{5–4}$$

where $\theta = [\theta_i \dots \theta_P]$ denotes the nonnegative parameter vector. However, not all positive-definite kernels can be taken to an arbitrary power and still be positive definite.

Only the class of positive-definite kernels that are infinitely divisible (Horn, 1967) can be taken to arbitrary powers such that the resulting kernel κ_θ is positive definite.

5.1.2.2 Infinitely divisible kernels

There are many infinitely divisible kernels, but our interest in metric learning leads us to the special case of kernels that are functions of distances $\kappa(x, x') = f(d(x, x')) = f(u)$. Here we rely on the work of Schoenberg (1938) who explored the connection between distance metrics and positive-definite kernel functions: a kernel that is a function of distance metric is only positive definite if the metric space can be isometrically embedded in Hilbert space. From Schoenberg (1938) Theorem 4, the most general function $f(u)$ which is bound away from zero and whose positive powers $[f(u)]^\lambda, \lambda > 0$ are positive definite is of the form $f(u) = \exp(c + \psi(u))$ where $\psi(u)$ is positive definite and c is a constant. For kernels of this form, positive powers simply scale the constant and function $[f(u)]^\lambda = [\exp\{c + \psi(u)\}]^\lambda = \exp\{c' + \lambda\psi(u)\}, \lambda > 0$.

Thus, a class of kernels whose positive powers are all positive definite are of the form $\kappa(x, x') = f(d(x, x')) = \exp(c + h(x, x'))$ where $h(x, x')$ is positive definite. Given a metric $d(x, x')$, $\kappa(x, x') = \exp(0 + h(x, x')) = \exp\{-g(d(x, x')\}$ is positive definite for only certain choices of $g(\cdot)$. In particular if $d_p(x, x')$ corresponds to a p-norm or an $\mathcal{L}_p$ metric then $\kappa(x, x') = \exp(-d_p^p(x, x'))$ is positive definite for $0 < p \leq 2$ (Schoenberg, 1938). Furthermore, the kernel $\kappa(x, x') = \exp(-d_p^\gamma(x, x'))$ is positive definite for $0 < p \leq 2$ and $0 < \gamma \leq p$ (Schoenberg, 1938). For $p < 1$, d_p is not actually a metric since it violates the triangle inequality; nonetheless, d_p^γ is embeddable in a vector space for $0 < \gamma \leq p$.

Clearly, the Gaussian kernel $\kappa(x, x') = \exp(-\theta d^2(x, x'))$ is positive definite if and only if $d(x, x')$ is a Euclidean or $\mathcal{L}_2$ metric; whereas, the Laplacian kernel $\kappa(x, x') = \exp(-\theta d(x, x'))$ is positive definite for an $\mathcal{L}_p$ metric with $1 \leq p \leq 2$.

5.1.2.3 Weighted product kernels

For kernels of this form, $\kappa_i^{\theta_i}(x_{(i)}, x'_{(i)}) = \exp(-\theta_i d^\gamma(x_{(i)}, x'_{(i)}))$, and substituting this equation into the weighted product kernel (5–4) yields

$$\kappa_\theta(x, x') = \prod_{i=1}^{P} \exp\left(-\theta_i d^\gamma(x_{(i)}, x'_{(i)})\right) = \exp\left(-\sum_{i=1}^{P} \theta_i d^\gamma(x_{(i)}, x'_{(i)})\right), \tag{5–5}$$

where θ_i can now be regarded as a parameter of kernel κ_i. Letting $\theta = w$ we have $\kappa_\theta(x, x') = \exp\left(-d_w^\gamma(x, x')\right)$; this shows the equivalence between the weighted metric (5–1) and parametrized product kernel (5–4).

5.1.2.4 Multivariate Gaussian kernel

Similarly, using the Mahalanobis metric (5–2) on scalar-valued data, a multivariate Gaussian kernel can be defined as the product of Q Gaussian kernels:

$$\kappa_A(x, x') = \exp\left(-d_A^2(x, x')\right) = \prod_{j=1}^{Q} \exp\left(-d^2\left(y_{(j)}, y'_{(j)}\right)\right), \tag{5–6}$$

where $y_{(j)} = \sum_i A_{i,j} x_{(i)}$.

5.1.2.5 Sum of product kernels

For scalar-valued data the weighted and Mahalanobis metrics correspond to linear projections. A nonlinear metric can be formed from the direct sum of kernels—as the sum of positive-definite functions is itself a positive-definite function. Let $\Theta = [\theta^1, \theta^2, \ldots, \theta^Q]$ denote a matrix of different weighting vectors corresponding to a set of product kernels $\{\kappa_{\theta^j}\}_{j=1}^{Q}$. Define κ_Θ as an unweighted sum of Q product kernels:

$$\kappa_\Theta(x, x') = \sum_{j=1}^{Q} \kappa_{\theta^j}(x, x') = \sum_{j=1}^{Q} \prod_{i=1}^{P} \exp\left(-\theta_i^j d^\gamma(x_{(i)}, x'_{(i)})\right). \tag{5–7}$$

Let ϕ_Θ denote the implicit mapping defined by the sum kernel. This mapping defines a metric between x and x' that corresponds to the $\mathcal{L}_2$ distance in the Hilbert space

$$d_\Theta(x, x') = \|\phi_\Theta(x) - \phi_\Theta(x')\|_2 = \sqrt{\kappa_\Theta(x, x) - 2\kappa_\Theta(x, x') + \kappa_\Theta(x', x')}. \tag{5–8}$$

5.1.2.6 Kernel matrices

In terms of a group of samples, the γ power of the distance matrix for the ith dimension is denoted $D_i^{\circ\gamma}$ where $[D_i^{\circ\gamma}]_{j,k} = d^{\gamma}(x_j(i), x_k(i)) \;\; j, k \in \{1, \ldots, n\}$. The notation $D^{\circ 2}$ denotes that each element is squared $D^{\circ 2} = D \circ D$ where $\circ$ denotes the entry-wise (Hadamard) product, as opposed to the matrix product $D^2 = DD$.

The kernel matrix for the ith dimension is $K_i = \exp(-\theta_i D_i^{\circ\gamma})$. The kernel matrix for the product and sum kernels are computed as $K_\theta = K_1 \circ K_2 \circ \cdots \circ K_P = e^{-\sum_i \theta_i D_i^{\circ\gamma}}$ and $K_\Theta = K_{\theta^1} + K_{\theta^2} + \cdots K_{\theta^Q}$. The labels of the trials can also be represented by a kernel matrix L, where each entry $L_{j,k} = \delta(l_j, l_k)$ use the 0-1 kernel, $\delta(l, l') = 1$ if $l = l'$ and $\delta(l, l') = 0$ if $l \neq l'$.

5.1.3 Neural Metrics

Out of the many possible neural response metrics, we consider the following metrics:

- Temporal metrics for multivariate time-series: Each individual distance is the Euclidean distance between the vectors of instantaneous amplitudes across the channels. Each weight corresponds to a particular time lag. The weight adjusts the importance of the distance between the spatial patterns of the two samples at that particular time.
- Spatiotemporal projections: A linear projection matrix is used to form a Mahalanobis distance.
- Spike train metrics: Each individual distance is between spike trains on the same unit at different temporal precisions. The weight adjusts the importance of each unit at a particular temporal precision value. There are a number of spike train metrics, but we consider two different metrics:
  - Spike train alignment metric. The metric is the $\mathcal{L}_1$ or $\mathcal{L}_2$ version of the Victor-Purpura (VP) spike train distance (Dubbs et al., 2010; Victor & Purpura, 1996).
  - Kernel-based spike metric. The metric is defined by the mapping ϕ induced by a spike train kernel (Park et al., 2013, 2012). We use the memoryless cross-intensity (mCI) spike train kernel (Paiva et al., 2009). Let $x = \mathcal{T}$ and

$x = \mathcal{T}'$ be two sets of spike times, the kernel is defined as

$$\kappa(x, x') = \langle \phi(x), \phi(x') \rangle = \sum_{t \in \mathcal{T}} \sum_{t' \in \mathcal{T}'} \exp(-q|t - t'|).$$

Then $\|\phi(x) - \phi(x')\|_2 = \sqrt{\kappa(x, x) - 2\kappa(x, x') + \kappa(x', x')}$ is an $\mathcal{L}_2$ metric (Paiva et al., 2009).

Alternatively, multichannel spike trains can be transformed to vectors in Euclidean space. First the spike timings for each unit, are quantized into fixed-width, contiguous, and non-overlapping bins. Then the binned spike count vectors for each neuron are concatenated and a spatiotemporal projection can be applied.

Based on these metrics we use kernel functions as measures of similarity. On each individual dimensions we use either the Gaussian kernel for the Euclidean and $\mathcal{L}_2$ distances or the Laplacian for $\mathcal{L}_1$ metrics such as the original Victor-Purpura metric. The kernels for individual dimensions are combined using the tensor-product kernel in Equation (5–5). The sum of product kernels (5–7) consists of an unweighted sum of the weighted product kernels with different weightings. For the Mahalanobis metric (5–2) a multivariate Gaussian kernel is used (5–6).

5.2 Kernel-based Metric Learing

We introduce a kernel-based measure to quantify the joint information between neural responses and labels corresponding to stimuli or condition. The measures can be used as an objective function to optimize the metric used to evaluate the similarity among neural responses.

5.2.1 A Kernel-based Measures of Dependence

Kernel target alignment measures the similarity between two kernel functions using their normalized inner-product (Cristianini et al., 2002). For jointly sampled data, the inner-product of kernel functions defines a measure of dependence between random variables (Gretton et al., 2005). Unlike Pearson's correlation-coefficient which uses the values of the random variables, kernel-based dependence assesses the degree to which the similarity of example pairs, as defined by each kernel function, matches or

aligns. In terms of distance-based kernel functions, the dependence could be posed as, “Do nearby examples, as defined by the first random variable, correspond to nearby examples in the second random variable?”

Consider the statistical alignment of two kernel functions. Let $z \in \mathcal{Z}$ denote a random variable and z' be an independent and identically distributed random variable. Let κ_1 and κ_2 be two kernel functions with implicit mappings $\phi_i : \mathcal{Z} \to \mathcal{H}_i$. A natural measure of similarity between these kernel functions is the expected value of their normalized inner product across pairs of realizations

$$A(\kappa_1, \kappa_2) = \frac{\mathrm{E}_{z,z'}[\kappa_1(z, z')\kappa_2(z, z')]}{\sqrt{\mathrm{E}_{z,z'}[\kappa_1^2(z, z')]\mathrm{E}_{z,z'}[\kappa_2^2(z, z')]}}. \tag{5–9}$$

Now, consider when $z = (x, y)$ represents a joint sample of $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ and κ_1, κ_2 only depend on x and y, respectively: $\kappa_1(z, z') = \kappa_x(x, x')$ and $\kappa_2(z, z') = \kappa_y(y, y')$. The marginal behavior of the kernels can be expressed in terms of their mapping functions:

$$\phi_1(z) = (\phi_x \otimes 1_y)(x, y), \text{ where } \phi_x : \mathcal{X} \to \mathcal{H}_x, \text{ and } \forall y \;\; 1_y(y) = 1 \tag{5–10}$$

$$\phi_2(z) = (1_x \otimes \phi_y)(x, y), \text{ where } \phi_y : \mathcal{Y} \to \mathcal{H}_y, \text{ and } \forall x \;\; 1_x(x) = 1. \tag{5–11}$$

$$\text{Then } A(\kappa_1, \kappa_2) = \frac{\mathrm{E}_{x,y}\mathrm{E}_{x',y'}[\kappa_x(x, x')\kappa_y(y, y')]}{\sqrt{\mathrm{E}_x\mathrm{E}_{x'}[\kappa_x^2(x, x')]\mathrm{E}_y\mathrm{E}_{y'}[\kappa_y^2(y, y')]}}. \tag{5–12}$$

is a measure of statistical dependence between x and y, since it is higher when similar pairs of one variable correspond to similar pairs in the other variable. However, the measure performs poorly in practice without centering the kernels first (Cortes et al., 2012).

Centering plays the same role as removing the mean when computing the correlation coefficient between scalar-valued random variables. The centered kernel

alignment is defined by Cortes et al. (2012) as

$$\rho(\kappa_1, \kappa_2) = A(\tilde{\kappa}_1, \tilde{\kappa}_2) \tag{5–13}$$

$$\begin{aligned}\tilde{\kappa}_i(z, z') &= \langle \tilde{\phi}_i(z), \tilde{\phi}_i(z') \rangle = \langle \phi_i(z) - \mathsf{E}_z[\phi_i(z)], \phi_i(z') - \mathsf{E}_{z'}[\phi_i(z')] \rangle \\ &= \kappa_i(z, z') - \mathsf{E}_{z'}[\kappa_i(z, z')] - \mathsf{E}_z[\kappa_i(z, z')] + \mathsf{E}_{z,z'}[\kappa_i(z, z')].\end{aligned} \tag{5–14}$$

Centering the mapping functions is key to a useful measure of dependence. The role of centering can be seen by expanding the numerator of the kernel target alignment in tensor-product form:

$$\begin{aligned}\mathsf{E}_{z,z'}[\kappa_1(z, z')\kappa_2(z, z')] &= \mathsf{E}_{z,z'} \langle (\phi_1 \otimes \phi_2)(z, z), (\phi_1 \otimes \phi_2)(z', z') \rangle \\ &= \langle \mathsf{E}_z[(\phi_1 \circ \phi_2)(z)], \mathsf{E}_{z'}[(\phi_1 \circ \phi_2)(z')] \rangle \\ &= \| \mathsf{E}_z[(\phi_1 \circ \phi_2)(z)] \|_2^2\end{aligned} \tag{5–15}$$

Writing the original kernel in terms of the centered kernel (5–14) yields

$$\begin{aligned}\mathsf{E}_z(\phi_1 \circ \phi_2)(z) &= \mathsf{E}_z(\tilde{\phi}_1 + \mathsf{E}_{z'}[\phi_1(z')]) \circ (\tilde{\phi}_2 + \mathsf{E}_{z'}[\phi_2(z')])(z) \\ &= \mathsf{E}_z(\tilde{\phi}_1 \circ \tilde{\phi}_2)(z) + \mu_1 \circ \mu_2 + \mu_2 \circ \tilde{\phi}_1(z) + \mu_1 \circ \tilde{\phi}_2(z) \\ &= \mathsf{E}_z(\tilde{\phi}_1 \circ \tilde{\phi}_2)(z) + \mu_1 \circ \mu_2\end{aligned}$$

where $\mu_i = \mathsf{E}_z(\phi_i(z))$ and $\mathsf{E}_z(\tilde{\phi}_i(z)) = 0$. In terms of the marginal kernels

$$\mu_1 \circ \mu_2 = \mathsf{E}_{x,y}[(\phi_x \otimes 1_y)(x, y)] \circ \mathsf{E}_{x,y}[(1_x \otimes \phi_y)(x, y)] = (\mathsf{E}_x \phi_x(x)) \otimes (\mathsf{E}_y \phi_y(y)) = \mu_x \otimes \mu_y,$$

which is only a measure of the marginals—not of their joint distribution—thus its biases the norm in Equation (5–15) regardless of the dependence between x and y.

Again if $z = (x, y)$ and $\kappa_1(z, z') = \kappa_x(x, x')$ and $\kappa_2(z, z') = \kappa_y(y, y')$, then $\rho(\kappa_1, \kappa_2) = \rho_{\kappa_x, \kappa_y}(x, y)$ is a measure of statistical dependence between x and y:

$$\rho_{\kappa_x, \kappa_y}(x, y) = \frac{\mathsf{E}_{x,y}\mathsf{E}_{x',y'}[\tilde{\kappa}_x(x, x')\tilde{\kappa}_y(y, y')]}{\sqrt{\mathsf{E}_x\mathsf{E}_{x'}[\tilde{\kappa}_x^2(x, x')]\mathsf{E}_y\mathsf{E}_{y'}[\tilde{\kappa}_y^2(y, y')]}}. \tag{5–16}$$

For positive-definite-symmetric kernels, $\rho_{\kappa_x,\kappa_y} \in [0,1]$ (Cortes et al., 2012). Centered alignment is essentially a normalized version of the Hilbert-Schmidt Information Criterion (Gretton et al., 2005).

5.2.1.1 Correntropy coefficient

Additionally, centered alignment is related to the localized similarity measure known as correntropy (Liu et al., 2007). Specifically, if x and y are in the same domain then a single shift-invariant kernel κ can be used to define the correntropy coefficient $\eta_\kappa(x,y)$ (Rao et al., 2011):

$$\eta_\kappa(x,y) = \frac{\mathsf{E}_{x,y}[\tilde{\kappa}(x-y)]}{\sqrt{\mathsf{E}_{x,x'}[\tilde{\kappa}(x-x')]\mathsf{E}_{y,y'}[\tilde{\kappa}(y-y')]}}$$

When $\kappa(x-y) = \mathsf{E}_{x',y'}[\kappa_x(x,x')\kappa_y(y,y')]$ then $\rho_{\kappa_x,\kappa_y}(x,y) = \eta_\kappa(x,y)$. For instance, this is the case if κ_x and κ_y are Gaussian kernels and (x,y) is normally distributed. However, this approach is not applicable for metric learning where x and y correspond to two different domains, i.e., labels and neural responses.

5.2.1.2 Empirical estimation

An empirical estimate of the centered alignment can be computed directly from the kernel matrices K and L where $[K]_{i,j} = \kappa_x(x_i,x_j)$ and $[L]_{i,j} = \kappa_y(y_i,y_j)$:

$$\hat{\rho}(K,L) = \frac{\langle \tilde{K},\tilde{L}\rangle}{\sqrt{\langle \tilde{K},\tilde{K}\rangle\langle \tilde{L},\tilde{L}\rangle}} = \frac{\langle \tilde{K},\tilde{L}\rangle}{\|\tilde{K}\|_2\|\tilde{L}\|_2} \tag{5–17}$$

where $\tilde{K}$ and $\tilde{L}$ are the centered kernel matrices. The centered kernel is computed as

$$[\tilde{K}]_{i,j} = [K]_{i,j} - \frac{1}{n}\sum_{i=1}^{n}[K]_{i,j} - \frac{1}{n}\sum_{j=1}^{n}[K]_{i,j} + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}[K]_{i,j}. \tag{5–18}$$

Using matrix multiplication, $\tilde{K} = HKH$, where $H = I - \frac{1}{n}\vec{1}\vec{1}^{\mathsf{T}}$ is the empirical centering matrix, I is the $n \times n$ identity matrix, and $\vec{1}$ is a vector of ones. The computational complexity of centered alignment between two $n \times n$ kernel matrices is $\mathcal{O}(n^2)$.

5.2.2 Metric Learning Optimization Using Centered Alignment as an Objective

Our objective for metric learning is to maximize the dependence between the neural data representation and the class label. Centered alignment is used to evaluate the dependence in terms of the kernel representations. The 0-1 kernel on the labels is fixed, and the parameters of a metric-based kernel defined in Section 5.1.2 are optimized in order to maximize the centered alignment.

For convenience, we use the logarithm of the centered alignment as the objective. With or without the logarithm the kernel-based objective is a nonlinear functions of the parameters, and we propose to use approximate inverse Hessian and stochastic gradient methods for optimization. We detail the gradients below.

First, we consider optimizing the sum and product kernels. As the product kernel (5–5) is the trivial case of the sum kernel (5–7), we consider only the optimization of the sum kernel parameters $\Theta = [\theta_i^j]_{i=1,j=1}^{P,Q}$ in the following problem:

$$\max_{\Theta \geq 0} \ \log(\rho_{\kappa_\Theta,\delta}(x, l)). \tag{5–19}$$

When the empirical estimate of centered alignment is substituted the explicit objective function is

$$f(\Theta) = \log(\hat{\rho}(K_\Theta, L)) = \log(\mathrm{tr}(K_\Theta HLH)) - \log(\sqrt{\mathrm{tr}(K_\Theta HK_\Theta H)}) + k \tag{5–20}$$

where K_Θ is the kernel matrix of the responses, L is the 0-1 kernel matrix of the labels, H is the centering matrix, and k is a constant that does not depend on Θ. The gradient of the objective function with respect to the kernel matrix is

$$G = \nabla_{K_\Theta} f(\Theta) = \frac{HLH}{\mathrm{tr}(K_\Theta HLH)} - \frac{HK_\Theta H}{\mathrm{tr}(K_\Theta HK_\Theta H)}. \tag{5–21}$$

The gradient kernel with respect to the kernel parameter θ_i^j is $\frac{\partial K_\Theta}{\partial \theta_i^j} = -K_{\theta^j} \circ D_i^{\circ\gamma}$. Then the gradient of the objective is

$$\frac{\partial f(\Theta)}{\partial \theta_i^j} = \operatorname{tr}\left((-K_{\theta^j} \circ D_i^{\circ\gamma}) G\right). \tag{5–22}$$

The non-negativity constraint on Θ can be removed by performing the optimization in terms of u where $\theta_i^j = 10^{u_i^j}$. The gradients can be made in terms of unconstrained optimization variables u_i by $\frac{\partial f(\Theta)}{\partial u_i^j} = \theta_i \log(10) \frac{\partial f(\Theta)}{\partial \theta_i^j}$. This yields an unconstrained optimization.

For the case of scalar-valued data we explore learning a Mahalanobis metric (5–6) using the logarithm of the empirical centered alignment as the objective:

$$\max_A \; f(A) = \log(\hat{\rho}(K_A, L)) \tag{5–23}$$

The gradient of the objective function with respect to A is

$$\nabla_A f(A) = -4X^\top \left((G \circ K_A) - \operatorname{diag}((G \circ K_A)\vec{1}) \right) XA \tag{5–24}$$

where X is a $n \times P$ matrix of the data, G is the gradient of the objective function with respect to the kernel matrix (5–21), and $\vec{1}$ is a vector of ones.

For the approximate inverse Hessian optimization we use `minFunc` (Schmidt, 2012), using the default limited memory BFGS update. For the sum and product kernels, prior to optimization the individual matrices $D_i^{\circ\gamma}$ are all normalized such that the average across all elements is 1. For the product kernel all the weights are initialized to be 10^{-3} and for the sum of product kernels they are uniformly distributed in $10^{-3} \pm 10^{-4}$. For the Mahalanobis distance, the optimization of A yields varying results depending on the initial value of A, but using the projection from Fisher discriminant analysis for initialization performs well in practice.

As an alternative optimization that can handle large sample sizes, we use a stochastic gradient over small batches. Specifically, we use a paradigm commonly used

in feature selection: at each iteration, one example is sampled and then a pre-specified number of examples of the same class and from differing classes are sampled to form the batch (Kira & Rendell, 1992). For each batch, the weights are updated based on the gradient of the objective. Very small batches—even just four examples—are sufficient for learning the parameters of the product kernel, but to learn a Mahalanobis distance we found larges batches, in the hundreds, are necessary.

5.3 Benchmark Comparison

Using publicly available datasets, we contrast the classification performance using centered alignment metric learning against using optimize a weighted metric to a feature weighting method (Takeuchi & Sugiyama, 2011). The feature weighting is explicitly optimized to improve the k-nearest neighbor classification; this serves as a benchmark for centered alignment metric learning, which is not tuned to any particular classifier. The method was shown to consistently outperform other feature weighting methods. For a valid comparison the specifics of the benchmark comparison by Takeuchi & Sugiyama (2011) are replicated; we use the same UCI machine learning datasets (Bache & Lichman, 2013; Cortez et al., 2009; Little et al., 2007) and classification scenario (one-third for training, one-third to choose k, number of nearest neighbors, through cross-validation, and one-third for testing). However, we increase the Monte Carlo divisions to 200 for statistical comparison. As a sanity check, Euclidean distance after normalizing the variance of the features is also used. We tested both the L-BFGS optimization and the mini-batch with 4 sample batches, 10,000 batches, and a step size of 0.01. We did not rerun the sequential quadratic program-based feature weighting (Takeuchi & Sugiyama, 2011), but instead list the value they report for the mean error rate across 10 Monte Carlos divisions.

The results are displayed in Table 5-1. On these small scale problems—maximum dimensions is 57—none of the compared methods consistently outperforms the best.

Considering the best of the two proposed optimization methods, centered alignment metric learning performs best on half of the datasets.

Table 5-1. Benchmark comparison across UCI datasets using different feature weightings.

Dataset	C	P	n	FW[a]	Euclid.[B]	CAML[c]	∼CAML[d]
Pen-Based Recog.	10	16	10992	1.1	*1.0±0.2	N/A	1.2±0.2
Breast Wisc. (Diag.)	2	30	569	*4.0	4.8±1.7	4.4±1.5	4.3±1.4
Page Blocks	5	10	5473	4.6	*4.1±0.5	4.6±0.5	4.3±0.5
Image Segmentation	7	18	2310	5.2	6.2±1.0	*3.3±0.8	4.6±0.8
Ionosphere	2	33	351	12.2	16.3±3.9	*10.7±4.9	13.7±3.5
Parkinsons	2	22	195	*10.2	12.1±4.3	13.6±4.7	11.5±3.9
Spambase	2	57	4601	*10.4	11.0±0.9	14.6±4.7	*10.4±0.8
Waveform (ver. 1)	3	21	5000	18.4	19.0±0.9	*17.9±0.9	18.5±0.8
Connectionist (Sonar)	2	60	208	22.1	*20.8±5.4	27.5±4.9	22.4±5.4
Wine Quality	7	11	6497	46.3	46.3±1.0	48.9±1.2	*46.0±1.0

Entries are classification error as percent incorrect, average and standard deviation taken across Monte Carlo runs, using a k-nearest neighbor classifier. Columns denoted $|C|$, P, and n indicate the number of classes, features, and samples, respectively.
[a]Sequential quadratic program-based feature weighting (Takeuchi & Sugiyama, 2011)
[b]Unweighted normalized Euclidean distance
[c]Centered alignment metric learning optimizing a product kernel
[d]Centered alignment metric learning with mini-batch approximation
*Indicates best performing methods, which were not significantly different, p-value greater than 0.05, for either a one-sample t-test versus SQPFW, or a two-sample t-test between other methods.

5.4 Decoding Forepaw Touch Location from Rat Somatosensory Cortex

In this section,multi-unit spike train metrics and temporally-weighted and spatiotemporal metrics on local field potentials are used to decode the location of touch on the forepaw of a rat. The spike trains and local field potentials from the forepaw region of the somatosensory cortex (S1) are used.

5.4.1 Data Collection

All animal procedures were approved by the SUNY Downstate Medical Center IACUC and conformed to National Institutes of Health guidelines. Cortical local field potentials and action potentials were recorded during natural tactile stimulation of forepaw digits and palm of 4 female Long-Evans rats under anesthesia. After induction using isoflurane, urethane was used to maintain anesthetic depth. A 32-channel

microelectrode array (Blackrock Microsystems) was inserted into the hand region of the primary somatosensory cortex (S1). The array was arranged in a 6×6 grid (excluding the four corners) with 400 μm spacing between neighboring electrodes. Another array was inserted into the VPL region of the thalamus, but the signals are not used here.

Using a motorized probe, the right forepaw was touched 225 times at up to 9 sites—4 digits and 5 sites on the palm. For each touch site, the probe was positioned 4 mm above the surface of the skin and momentarily pressed down for 150 ms, as seen in Figure 5-1; this was repeated 25 times at random intervals. The 4 datasets had 3, 8, 9, and 9 touch sites resulting in 75, 200, 225, and 225 samples, respectively.

Figure 5-1. Experimental setup showing motorized lever touching digit 1 on the forepaw.

The LFPs were band-pass filtered with cutoffs (5 Hz, 300 Hz) and sampled at a rate of 1220.7 Hz. Then the LFPs were digitally filtered using a 3rd-order Butterworth high-pass filter with cutoff of 4 Hz and notch filters at 60 Hz and harmonics. For analysis, the neural response in a 270 ms window following each touch onset was used, which corresponds to 330 discrete time samples. For 32 channels, this results in 330×32=10,560 dimensions.

Across the 4 datasets, automatic spike-sorting selected 95, 64, 64, and 38 multi-neuron units from the 32 channels. Of these, only 68, 62, 36, and 24 units were used, whose average firing rate was below 30 Hz in the 270 ms window following touch onset.

5.4.2 Results

We explored centered alignment metric learning (CAML) for both spike trains and local field potentials (LFPs) using the cases listed in Section 5.1.3. For LFPs and binned spike trains we compared with multi-class Fisher discriminant analysis (FDA) (Fukunaga, 1990) and large-margin nearest neighbor (LMNN) (Weinberger et al., 2006; Weinberger & Saul, 2009). For the linear projections, PCA was used to reduce the dimensionality to 1/2 of the number of samples in the dataset. The FDA solution is the set of eigenvectors corresponding to a generalized eigenvalue problem. The dimensions can be chosen as the maximum number of non-zero eigenvalues, which is one less than the number of classes (Fukunaga, 1990). The FDA solution was used as the initial projection for LMNN and CAML Mahalanobis metric. An efficient MATLAB implementation of LMNN is publicly available, and besides the initialization (which greatly increased the performance) default parameters were used.

To compare classification performance, 20 Monte Carlo divisions of the datasets into training and testing sets were made. For training, two-thirds of the samples in each class were used, the remainder of the samples were used in testing. On each Monte Carlo run, the metrics were optimized on the training set. Testing set samples were labeled by either a one-nearest-neighbor (1-NN) or a support vector machine (SVM) classifier. SVM training and testing was performed using the `libsvm` (ver. 3.17) (Chang & Lin, 2011) implementation with the user-provided kernel matrix. The regularization parameter was chosen through 5-fold cross-validation. For CAML the kernel is directly optimized as part of the metric learning, but for FDA, LMNN, and the unweighted metrics a Gaussian kernel was used with the kernel size chosen from a discrete set using 5-fold cross-validation.

The set of the highest performing methods for each dataset was found by selecting the best performing method and finding those that were not significantly different using a two-sample Welch test with significance of 0.05.

5.4.2.1 Learning multi-unit spike train metrics

Multiunit spike-train metrics using the single-unit Victor-Purpura (VP) and kernel-based (mCI) metrics were optimized for touch location classification. For each unit the distance is computed with different values for the temporal precision value q (higher values of q require more precise alignment): for the Victor-Purpura distance the set (0.01, 0.1, 1.0) s^{-1} was used, and for the spike train kernel-based metrics (mCI) the set (10^{-9}, 0.01, 0.1, 1, 10, 100) s^{-1} was used. For the Victor-Purpura distance, the $\mathcal{L}_2$ version (Dubbs et al., 2010) was used.[1] The classification rates for the weighted spike-train metrics are in Table 5-2. With the CAML-optimized product kernel the average classification rate increased by at least 8 percentage points for both metrics and both classifiers. For the sum kernel with 5 product kernels the accuracy was further increased.

For binned spike trains a Mahalanobis metric was optimized using FDA, CAML, and LMNN. The results across a range of different bin sizes are shown in Figure 5-2. On three datasets the best binned metrics performed worse than the optimized spike-train metrics. For each dataset and method the performance using the bin size with the highest average accuracy is shown in Table 5-3. On three of the datasets the Mahalanobis metric optimized with CAML tied or outperformed the FDA solution, and on all datasets using LMNN decreased performance.

We used classical multidimensional scaling (MDS) (Torgerson, 1952) and t-distributed stochastic neighborhood embedding (t-SNE) (van der Maaten & Hinton, 2008) to find a two-dimensional embedding of the distance matrices before and after training the metric. The embedding is formed without knowledge of the class labels. From Figure 5-3 it is clear that metric learning with the product kernel increases distances among the different classes while decreasing the distances among samples

[1] Experiments with the original $\mathcal{L}_1$ version (Victor & Purpura, 1996) with a Laplacian kernel were also performed, but there was no significant difference.

Table 5-2. Comparison of touch site classification accuracy using multi-unit spike train metrics.

	VP metric unweighted		mCI metric unweighted		VP metric θ		$\ominus$	mCI metric θ		$\ominus$
Dataset	1-NN	SVM	1-NN	SVM	1-NN	SVM	SVM	1-NN	SVM	SVM
1	53±9	69±9	60±9	59±8	86±6	87±5	85±9	85±4	*90±6	*92±5
2	35±4	80±5	70±5	78±5	77±5	87±5	*91±4	78±4	87±5	*89±3
3	28±5	50±4	43±4	50±6	44±6	53±4	*59±5	48±5	58±6	*61±4
4	22±4	28±6	25±4	27±5	22±5	*38±5	*38±5	22±4	29±9	34±4

Entries indicate the mean and standard deviation of percent correct computed across 20 Monte Carlo runs. Victor-Purpura (VP) and kernel-based (mCI) metrics in unweighted combinations are compared alongside of using centered alignment metric learning (CAML) to optimize a product kernel (θ) or sum of 5 weighted product kernels ($\ominus$). Nearest-neighbor (1-NN) and support vector machine (SVM) were used as classifiers. *Indicates methods with highest accuracy with or without binning, see Table 5-3.

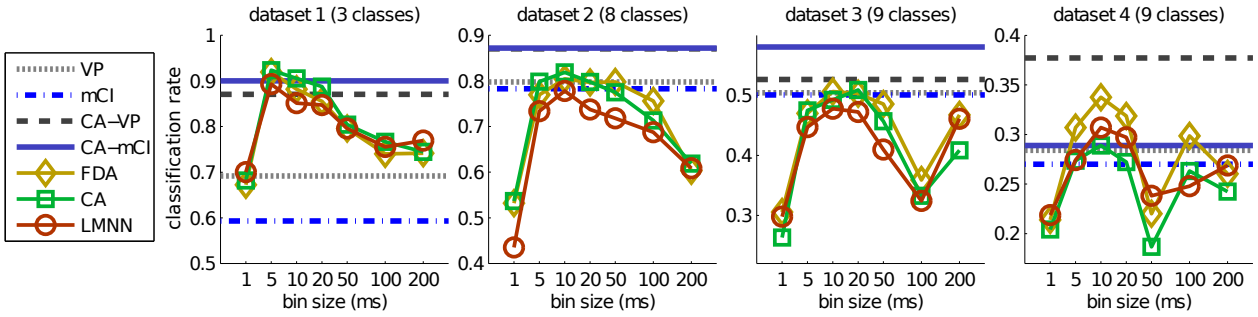


Figure 5-2. Comparison between multi-unit spike train metrics and binned spike count metrics with varying bin sizes. The SVM-based classification rate is shown for the unweighted Victor-Purpura (VP) and kernel-based (mCI) metrics, centered alignment metric learning optimized spike train metrics (CA-VP, CA-mCI), and Mahalanobis metrics on binned spike trains optimized with Fisher discriminant analysis (FDA), centered alignment (CA), and large margin nearest neighbor (LMNN).

within the same class. In Figure 5-4, we show how the optimized spike-train metric can be used to identify both the temporal precision and spiking units most useful for the decoding task.

5.4.2.2 Learning local field potential metrics

The classification rates for learning spatiotemporal metrics on LFP are tabulated in Table 5-4. Using CAML to optimize just a single temporal weighting improves the accuracy by 21 and 14.7 percentage points for 1-NN and SVM, respectively. Using a

Table 5-3. Comparison of touch site classification accuracy using binned spike trains and Euclidean or Mahalanobis-based metrics.

	Euclidean		A-FDA		A-CA		A-LMNN	
Dataset	1-NN	SVM	1-NN	SVM	1-NN	SVM	1-NN	SVM
1	58±10	73±9	*91±8	*92±14	*92±9	*92±10	*91±12	*89±14
2	66±6	76±7	80±8	80±9	82±6	82±7	77±8	78±10
3	42±5	46±8	47±6	51±6	51±6	51±7	46±7	48±10
4	24±5	28±6	30±6	34±7	29±6	29±6	29±6	31±6

Entries indicate the mean and standard deviation of percent correct computed across 20 Monte Carlo runs. Mahalanobis-based metrics are parametrized by matrix A and optimized using either Fisher discriminant analysis (FDA), centered alignment (CA), or large margin nearest neighbor (LMNN). For each dataset and method the bin size with the maximum performance was selected.
*Indicates highest performing methods for each dataset with or without binning see Table 5-2.

sum kernel composed of 5 product kernels further increased the performance by 2.2 and 4 percentage points. The optimized weights for a single product kernel are shown in Figure 5-5. Overall, using FDA to optimize a linear projection was able to achieve the highest classification rates with average improvement over Euclidean distance by 33.5 and 23.4 percentage points for 1-NN and SVM.

Table 5-4. Comparison of touch site classification accuracy using LFPs.

Dataset	Euclidean	θ-CAML	$\ominus$-CAML	A-CA	A-FDA	A-LMNN
1	89±4.4	94±3.7	97±2.2	*98±2.0	*98±2.0	97±2.4
2	75±5.0	84±3.0	89±2.4	95±2.6	*97±1.9	94±5.0
3	61±5.4	79±4.9	83±4.0	*91±4.7	*92±3.5	87±2.9
4	54±5.1	81±4.2	*85±3.8	*86±3.8	*86±4.5	80±5.9

Entries indicate the mean and standard deviation of percent correct computed across 20 Monte Carlo runs. SVM is used as classifier. Centered alignment metric learning (CAML) is used to optimize a single temporal weight vector θ or a sum kernel with multiple temporal weightings $\ominus$ Mahalanobis-based metrics (parametrized by a matrix A) optimized using Fisher discriminant analysis (FDA), centered alignment (CA), and large margin nearest neighbor (LMNN).
* indicates highest performing methods for each dataset.

Finally, a multiscale metric was learned as the weighted combination of the optimized spike distance and optimized LFP distance. On these datasets the combination of spiking and LFPs did not increase the classification rate versus using only LFPs, and the weight assigned to the spiking metric was insignificant compared to the weight

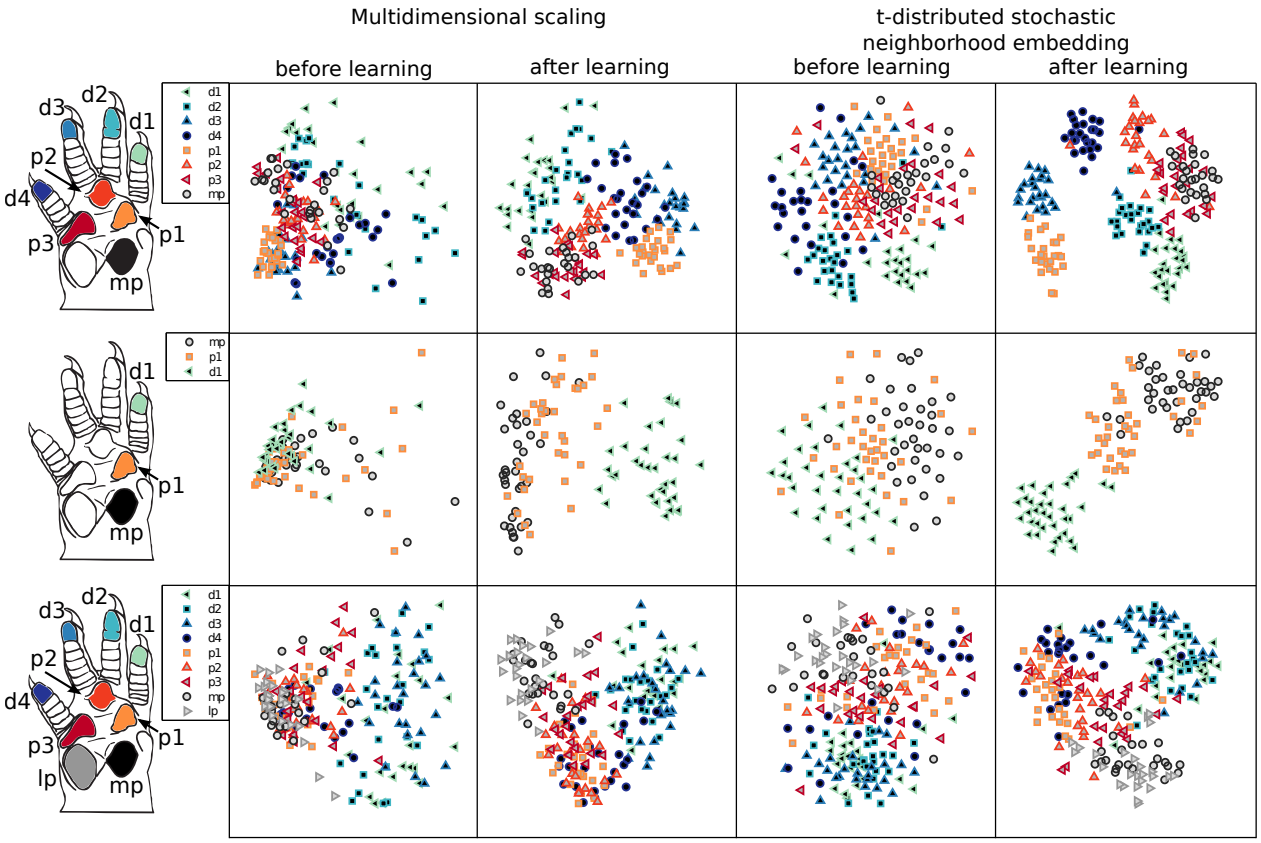


Figure 5-3. Comparison of metric-based dimensionality reduction before and after using centered alignment metric learning (CAML) to optimize a weighted combination of Victor-Purpura spike train distances. A two-dimensional embedding is found with t-distributed stochastic neighborhood embedding (t-SNE) and multidimensional scaling (MDS) before and after learning. For the t-SNE algorithm, the perplexity parameter was fixed at 10.

assigned to the LFP metric. The average classification accuracy across the datasets was slightly lower than using just the LFP metric.

5.4.2.3 Discussion

From the results it is clear that metric learning achieves three goals: increases the decoding accuracy, identifies important dimensions of the neural response, and improves the ability of manifold learning techniques to visualize the data in a low-dimensional space.

For spike trains, the average performance of the optimized multi-unit spike train metrics exceeded those based on binning. To our knowledge this is the first work

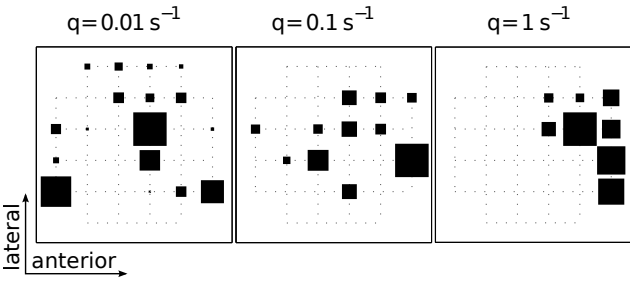


Figure 5-4. Learned weights for spiking unit and temporal precision pairs for the optimized Victor-Purpura spike-train metric (CAML-VP) shown across the array as a Hinton diagram—the size of each square is relative to the maximum weight of all units on the channel. Each subplot shows the weights for a different choice of the temporal precision value q; the weights for all temporal precision values are learned at the same time for dataset 2.

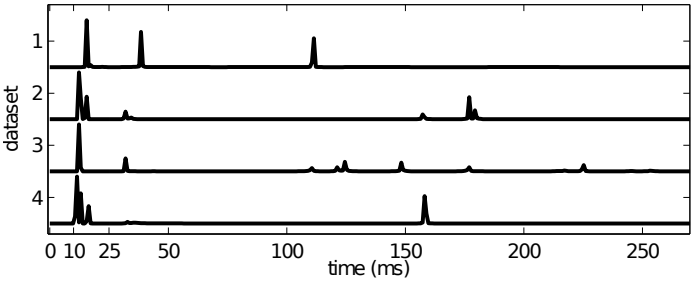


Figure 5-5. Learned temporal weighting of the optimized local field potential metric for decoding touch site across all datasets.

on optimizing a multi-neuron metric that is non-parametric and does not require binning. In the framework of the kernel-based dependence measure, this optimization explicitly optimizes the contribution of each dimension using tensor-product kernels for multi-neuron spike trains.

On all the datasets the performance from using the unweighted multi-neuron spike train metrics was lower than using the optimized Mahalanobis metrics on the binned representation. In essence, a simple linear projection of binned spike trains performs better than binless metrics that are not optimized. The precision offered by binless methods is only realized after optimization. This highlights the importance of metric learning versus naively combining the single-unit metrics.

FDA achieved the best performance for this touch decoding task using the time-locked evoked LFPs. FDA is well-suited for this setting since the class conditional

LFP responses are approximately normally distributed—an underlying assumption for FDA. In addition, the FDA solution is also the fastest solution; consequently, FDA should always be the baseline for discrete decoding of sensory evoked LFPs. Alternatively, for binned spikes using CAML to further optimize the FDA projection marginally increased the classification performance. Overall, FDA and CAML outperformed LMNN in optimizing a Mahalanobis metric.

One drawback of the Mahalanobis metric is the ability to analyze the projection matrices themselves, i.e, it is difficult to match and compare linear projections learned across multiple subjects or tasks, especially for high-rank projections. In this case using a weighted metric, which has lower accuracy but far fewer parameters, is more easily interpretable. From Figure 5-5 it is clear that the weighted metrics can be used to identify dimensions, in this case time lags, that are useful for discrimination. In addition, it appears that the optimization leads to a very sparse set of weights.

In terms of neural decoding, we compared the classification rate, as a proxy for the information content, of the neural responses. We have also highlighted how changing the underlying metric of the neural response space can improve the visualization results from unsupervised manifold learning algorithms. Indeed from Figure 5-3 a user can immediately judge which classes are more similar or indistinguishable. The non-linear embeddings preserve some features of the stimulus space's topology, e.g., the separation between digit responses and palm responses in dataset 3 and the preservation of the relative arrangement of the three touch sites in dataset 2.

5.5 Decoding Reach Target from Monkey Premotor Cortex

In this section, we used spike trains recorded from the dorsal premotor cortex (PMd) cortex of a female bonnet macaque through chronically implanted cortical microelectrode arrays as the subject performed center-out reaching task. The data was provided by Pratik Chhatbar and Joseph Francis at SUNY Downstate Medical Center and used with their permission. Brandi Marsh and Shaohua Xu helped record the data.

The details of the task, microelectrode implantation, and behavioral recordings were described in Section 2.3.1. PMd is an area known to encode premeditation of movement (Santhanam et al., 2006). Only units whose average firing rates was greater than 2 Hz and less than 30 Hz during the reach trials are used for analysis; 38 units met this requirement. There were 150 trials among the 8 targets.

5.5.1 Results

The goal is to decode the intended reach target during a 300 ms hold period where the subject is shown the location of the goal target but before the reach has begun; during the hold period subject's hand must remain at the center target or the trial aborts and no reward is delivered. For comparison, the window of activity for the first 300 ms of the reach and a control window consisting of the 300 ms hold period before the reach target are used. Decoding performance is gauged using metric and kernel-based classifiers. The mCI spike train kernel is used for the single-unit kernels and the corresponding metric; the temporal precision is set to 100 ms. Unweighted multi-unit metrics and kernels are compared to metrics optimized using CAML.

5.5.1.1 Classification across windows of each trial

The classification accuracy in three disjoint windows—pre-cue (control), cued with hold, and movement—of each trial is calculated across 80 Monte Carlo divisions using two-thirds of the trials for testing and the remaining for testing. The statistics of the classification accuracy (in percent correct) are shown in Table 5-5; the optimized multi-unit metrics and kernels outperform the unweighted versions.

5.5.1.2 Visualization using metric-based embedding

Again, classical multidimensional scaling (MDS) (Torgerson, 1952) is used to compare how well the unweighted and optimized metrics capture intrinsic relationship between the neural activity and the reach movements. The visualization in Figure 5-6 helps elucidate why the classification performance saturated around 55%: the neural responses are very similar for two distinct sets of reaches. The two groups are reaches

Table 5-5. Comparison of multi-unit metrics for reach target decoding.

Window	3NN unweighted	3NN CAML	SVM unweighted	SVM CAML
pre-cue (control)	15.5±3.9	13.8±3.3	16.6±4.4	15.1±4.6
cued and hold	43.1±5.2	47.1±5.5	55.8±5.9	57.2±5.7
reach	47.4±5.2	53.1±6.0	63.0±5.7	64.0±5.4

Entries indicate the mean and standard deviation of percent correct computed across 80 Monte Carlo runs. For each unit, the spike train kernel induced distance is computed, then an unweighted combination of these distances is compared to using centered alignment metric learning (CAML) to optimize a weighted distance. Results for the third nearest neighbor (3NN) and support vector machine (SVM) are reported.

between 9 o'clock and 2 o'clock and reaches between 3 o'clock to 8 o'clock, where the clock is in the horizontal plane with 12 o'clock corresponding to a ventral movement along the subject's midline.

This analysis could have been done manually by analyzing the confusion matrix of the classifiers, but the visualization provides simple confirmation. This illustrates how dimensionality reduction can enable visualizations that are predictive of the classification performance. In some cases, a classifier may be able to perform better in a higher-dimensional space, yet the separation seen in the visualization suggests a lower bound on the classification accuracy.

5.5.1.3 Effect of training set cardinality on performance of metric learning

The effect of training set size on both the metric-learning optimization and the classifiers is studied using more Monte Carlo experiments. The initial batch of samples given to CAML is varied along with the number of samples provided to the classifier. For each case, 80 Monte Carlo divisions are made and the averages are shown in Figure 5-7. The results indicate that the metric learning optimization outperforms the unweighted distance only when there is least as many samples in its batch as there are optimized parameters. In this dataset there are 38 weights corresponding to the 38 units, so an initial batch of 40 samples was sufficient for 1NN, but for SVM, 60 training samples (34% of the dataset) were needed.

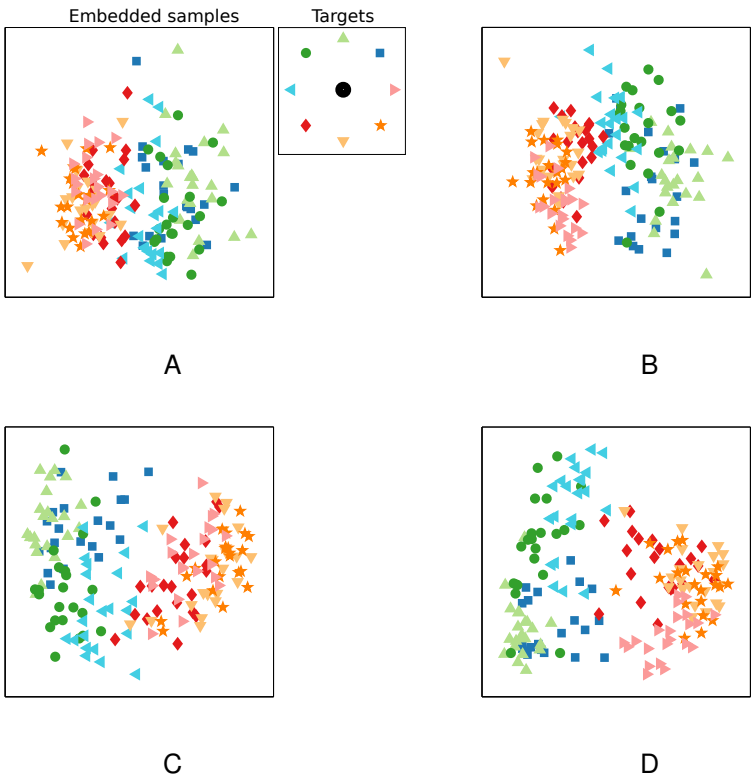


Figure 5-6. Metric-based dimensionality reduction on the premotor cortex data: before and after using centered alignment metric learning. Two-dimensional embedding is formed using multidimensional scaling (MDS) on different windows and different metrics: A) Using unweighted distance during the hold period. B) Using unweighted distance during the reach. C) Using the optimized distance during the hold period. D) Using the optimized distance during the reach.

5.5.1.4 Analysis of spike train unit weights

A regression analysis was performed between the optimized unit weights and measures of the single-unit firing rate and spike train distances. Besides the firing rates, the average of the log-normalized distances was computed for each unit between samples within the same class and samples of different classes. The log was used since single-trial distances are positive and right skewed. Since the weights should be indicative of the utility of a unit for discrimination, not the absolute value of the distances, the ratio of the interclass to intraclass log-normalized distances was also computed. Four different independent variables were tested: a unit's firing rate, unit's

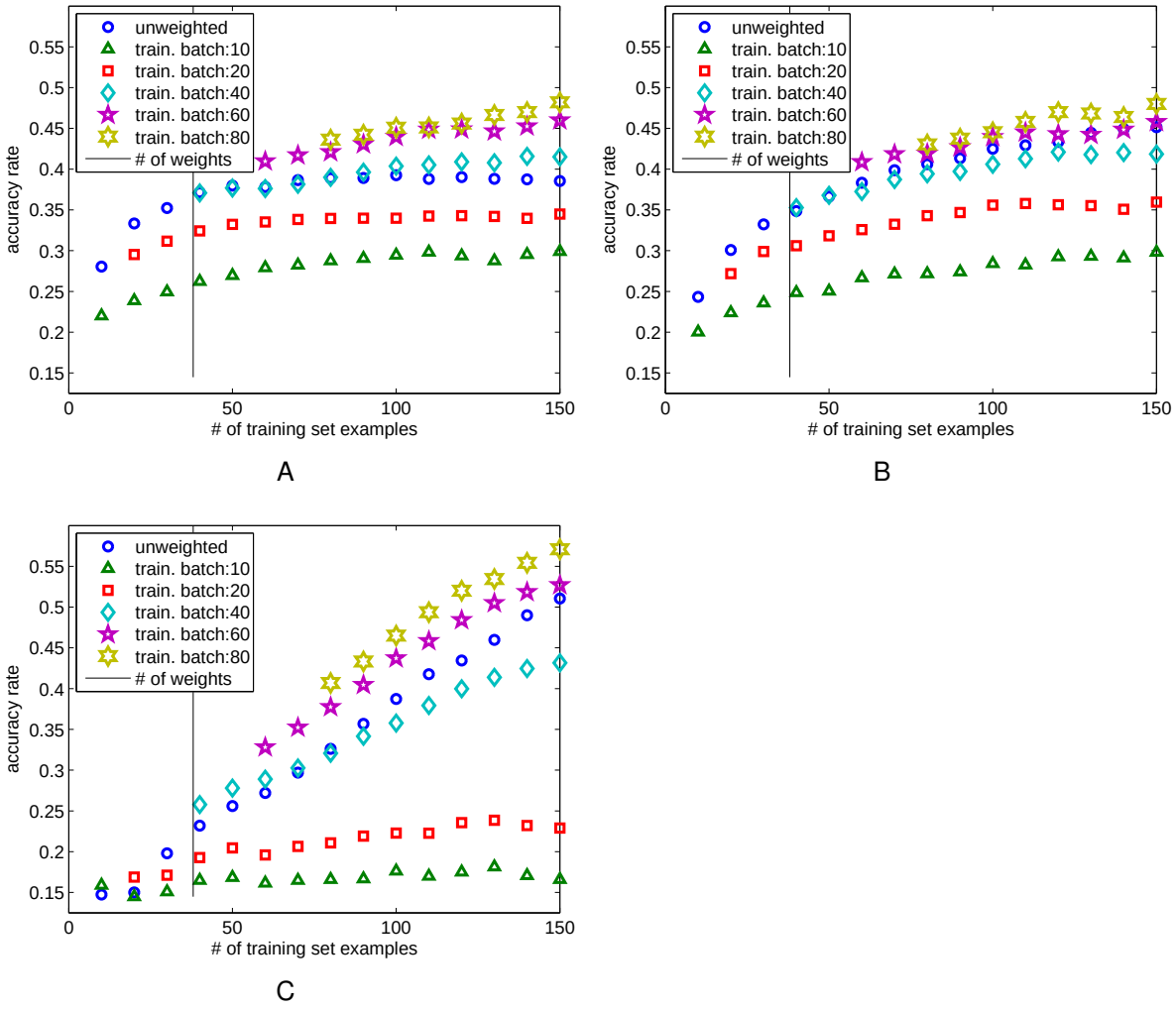


Figure 5-7. Reach target decoding performance across different sizes of training sets. The unweighted multi-unit metric is compared to the weighted metric optimized by centered alignment metric learning: using different sizes of initial batches. A) First nearest neighbor (1NN) classifier. B) Third nearest neighbor (3NN) classifier. C) Support vector machine (SVM) classifier.

average log-transformed distances between samples in the same class, a unit's average log-transformed distances between samples in different classes, and the ratio of a unit's average log-transformed interclass distances to intraclass distance. There are $n = 38$ feature weights corresponding to the single-units from PMd. The scatter plots and correlation coefficients are reported in Figure 5-8. Only for the distance ratio was there a

statistically significant correlation ($p < 10^{-11}$), which is well below the significance level of 5/4=1.25%, adjusted via Bonferroni's correction.

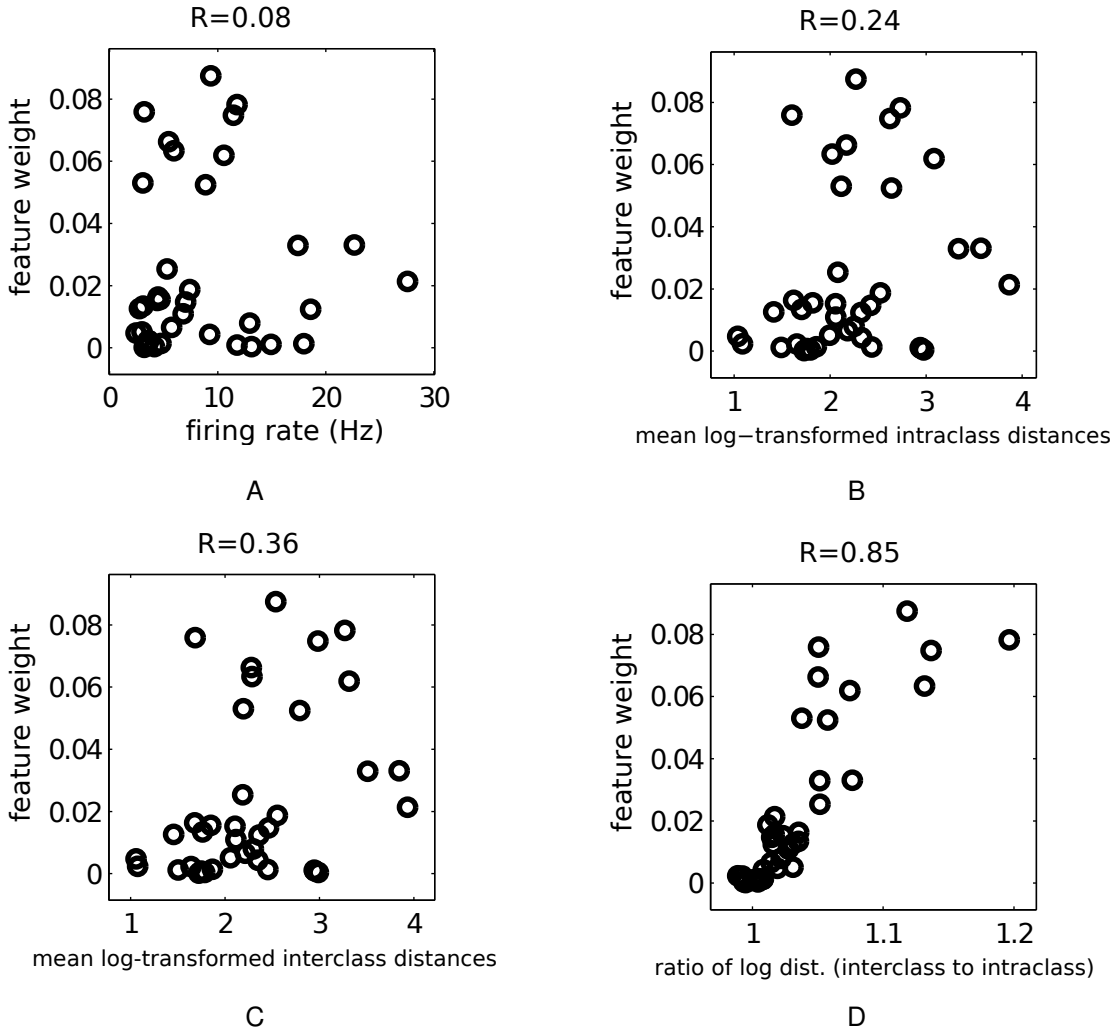


Figure 5-8. Regression analysis between the optimized weights and different independent variables derived from the spike trains and metric. The correlation coefficient is denoted above each scatter plot. Each subplot shows a different dependent variable: A) Firing rate. B) Average log-transformed distances between samples in the same class, using the single-unit spike train distance. C) Average log-transformed distances between samples in different classes. D) Ratio between the average log-transformed distances for interclass versus intraclass.

This analysis shows, that in this dataset, the weights optimized with CAML indicate the single-units ability to discriminate between classes. This is a common aspect of all feature selection algorithms, beyond this relative ordering, CAML was able to find

the absolute value necessary for improving kernel classification—without separately optimizing the kernel size.

5.6 Metric Learning for Neural Encoding

We have concentrated on the problem of neural decoding, but the proposed algorithms are also applicable to the neural encoding problem, wherein, the role of the stimulus and neural response are reversed. More specifically, for neural encoding, the metric on the neural response is fixed, the neural activity, e.g., the spiking of a single neuron, is treated as the target or label variable and a metric on the stimulus is adjusted. For instance, if the neuron is assumed to be a simple cell with a linear receptive field, then learning the receptive field is equivalent to learning a Mahalanobis distance on the stimulus space.

The ideas that have been developed for metric-learning/supervised dimensionality reduction in the machine learning community are fundamentally similar to the algorithms for inferring the linear receptive fields of neurons in the computational neuroscience community, but the nomenclature and domain has differentiated them. Recently, researchers have begun to bridge this gap using kernel-based measures of dependence (Sinz et al., 2013). To further highlight this connection, we replicated an experiment used to explain the maximally informative directions algorithm (Sharpee et al., 2004), using centered alignment metric learning to learn the neural encoding model.

The underlying model corresponds to a predefined filter consisting of 3 Gaussian bumps with equal covariance, see Figure 5-9. This resembled the shape of the filter used by Sharpee et al. (2004), but here the Gaussian bumps are offset instead of being centered. This filter corresponds to the linear weights of a model simple cell, a stochastic neuron. The value of the inner product between an input image and the filter, denoted s, is proportional to the probability of the neuron spiking/firing or not. Specifically, a zero-mean Gaussian random variable e with variance a is added to the inner-product, if this sum is greater than the threshold b then a spike is generated.

As input, we use patches from a database of natural images (van Hateren & van der Schaaf, 1998) consisting of buildings, parks, trees, etc.

Square patches of 30 by 30 pixels were randomly sampled from the images. The simulated cells parameters a and b are set relative to the standard deviation of s. Specifically $a = 0.31\sigma(s)$ and $b = 1.8\sigma(s)$, using the same values as Sharpee et al. (2004). The absence or presence of spike for a given patch is treated as a label. A set of 40,000 patches and the corresponding labels were given to the metric learning algorithm. Mini-batch optimization was run and the results are displayed in Figure 5-9 for the Mahalanobis-based metric and a weighted metric. To our knowledge, this was the first attempt to use a weighted metric algorithm to infer the importance of individual pixels on a simulated simple cell.

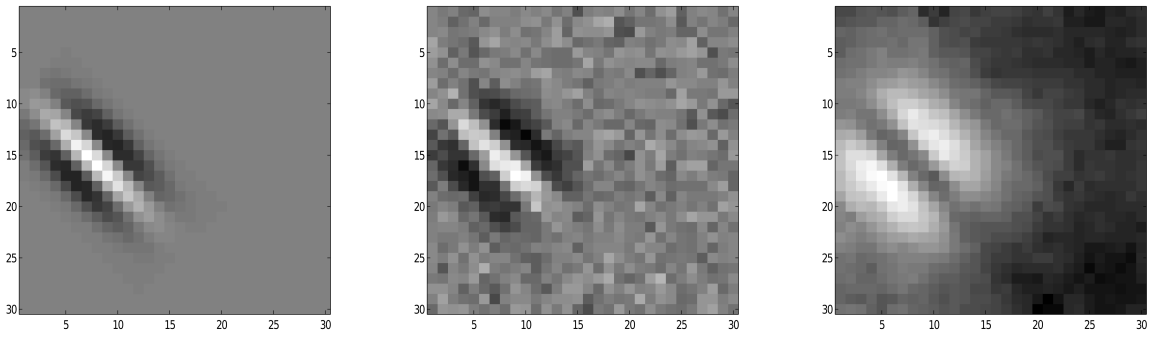


Figure 5-9. Using metric learning to estimate the filter for a simple cell model. A) True filter. B) Learned projection using 100 sample batches and 2000 batches. C) Learned weighting using 100 sample batch and 2000 batches.

Interestingly, most neural encoding models have concentrated on linear projections corresponding to Mahalanobis-based distances, whereas recent work has shown that the stimulus metric corresponding to a neural population can be highly non-Euclidean Tkačik et al. (2013). Thus, future work can investigate how non-Euclidean metrics can be learned. Additionally, the joint optimization of the metrics on both the neural response and the stimulus is worth investigating in future work.

5.7 Summary

We have introduced kernels applicable to metric learning allowing us to use a kernel-based dependence measure to optimize metrics. In the kernel-learning framework, changing the metric corresponds to changing the kernel. This leads to non-convex optimization, which we solve using first-order methods. Adjusting the kernels is a distinct approach from multiple kernel learning Cortes et al. (2012); Lanckriet et al. (2004); Yamada et al. (2013), where there is an explicit weight associated with each kernel in the sum and each summand kernel is chosen a priori (i.e., in the case of Gaussian kernel, the kernel size is not optimized and must be preselected). The main benefit of multiple kernel learning is that a convex optimization problem can be posed to optimize the weights. Alternatively, weighted product kernels and the sum of weighted product kernels constitute a much richer family of kernel functions than the weighted sum of kernels. We only need to select the number of summand kernels; fully exploring how to choose this number is left for future work.

Linear projections and weighted metrics are two special cases of metric learning that have received the most study. Indeed, the weighted combination of metrics was used in some of the earliest work on metric learning (Lowe, 1995). We have gone beyond this by using a sum of weighted product kernels, which computes distances in the Hilbert space that correspond to nonlinear transformations of the data samples. The sum of weighted product kernels still has interpretable parameters, quite unlike kernelized projections (Baudat & Anouar, 2000), where the transformations are only defined in terms of the samples instead of the original dimensions.

These metrics were optimized on neural datasets consisting of both spike trains and local field potentials, for which metric learning improves both nearest neighbor and SVM classification accuracy over unweighted alternatives. Within the proposed framework, the optimized multi-unit spike train metrics, which avoid binning, outperform both unweighted multiunit metrics and metrics optimized for the binned spike trains.

In addition, metric learning improves the quality of the visualizations—obtained via metric-based dimensionality reduction—for analyzing the relationship between high-dimensional neural data and target variables. The optimized weights themselves indicate the relative relevancy of different dimensions of the neural response. This can be used to explore how the weights of specific channels or neurons change for different tasks. Overall, optimizing metrics is a worthwhile approach for investigating neural representations.

CHAPTER 6
CONCLUSION

This dissertation highlights several specific applications of how learning new representations from neural signals is useful for exploratory analysis and improving decoding. The approaches naturally lead to data-dependent processing tuned to the particular structure of the signals. The methods developed and tested in this study are either unsupervised, as in Chapter 2, or semi-supervised, i.e., where knowledge of the variable of interest is used to learn the relative importance of specific dimensions of neural signals, as in Chapter 5. For neural potential signals, a linear synthesis model is applicable, and the methods proposed in Chapter 3 and Chapter 4 provide the means to decompose and extract meaningful features from neural potentials when linear filtering is inadequate. These decompositions exploit structure in both time and across space.

6.1 Applications to Electroencephalography

Throughout this dissertation, the majority of the analysis has been performed on invasive neural recordings of non-human subjects. Non-invasive recordings from human subjects offers the ability to use electroencephalography for brain-computer interfaces (Farwell & Donchin, 1988), attention monitoring (Davidson et al., 2007; Jung et al., 1997; Ray & Cole, 1985), pathological diagnosis (Jeong, 2004), and even entertainment purposes. The methodology developed in this dissertation could certainly impact these avenues of research.

6.2 Sparse Decompositions of Long-term Recordings

One area for future work is the application of the recurrent waveform decompositions to continuous, long-term neural recordings. The goal would be to train a generative model that can explain the structure seen across hours or even days of behavior using the atomic decomposition—consisting of the timing, waveform index, and amplitude. These factors reveal the reoccurring patterns in neural data, which then can be applied to the task of decoding.

The atomic decomposition was applied for segments of LFPs in Chapter 3, but the relationship of the aspects of this decomposition (timing, index, and amplitude of the sources) with the experimental cues, conditions, or neural spike train activity was not assessed. In Chapter 4, this relationship was assessed using a restricted form of the decomposition, which was limited to exactly one single activation per trial per filter. Can the unconstrained decomposition be useful for the same purpose?

Preliminary evidence indicates that this is the case. The simplest approach is to locally summarize the atomic decomposition in terms of the energy of each filter in short time segments. An alternative is to treat the atomic decomposition as a realization of a marked point process. Then, marked extensions of point process analysis methods can be applied. In particular, it may be interesting to consider extending the multi-unit spike train kernels and metrics discussed in Chapter 5 to the marked case. A trivial extension simply uses the amplitude as another dimension. An elegant approach to balance the importance of both timing and amplitude—across all the filters—will require further research.

6.3 Extensions of the Spatiotemporal Models

In Chapter 4, we considered a range of tensor decompositions and evoked-potential models previously proposed in the literature including the differentially variable component analysis (dVCA) (Truccolo et al., 2003) with multiple components. The dVCA algorithm is in reality a block-based version of the MP-SVD algorithm (Mailhé et al., 2008) most resembling MoTIF (Jost et al., 2006). As in Chapter 3, we introduced a greedy version of this approach. In certain cases, the greedy approach was faster and had better approximation performance. The main benefit is that the greedy approach always converges and naturally allows a different spatial factor for each component; future work should analyze whether these spatial factors indicate unique sources. These greedy approaches come with the caveat that they may increase the rank of the

underlying model—a fact established for tensor decompositions (Stegeman & Comon, 2010)—resulting in a higher-order model being estimated than necessary.

For evoked potentials, the time-locked model and the shift-varying model are both relevant. Instead of choosing one or the other, one could investigate a heterogenous tensor model (Brockmeier et al., 2013b) that consists of both a time-locked tensor model and a shift-varying model. This may be more interpretable than having two differentially variable models.

The models were optimized in terms of the mean squared error of the approximation assuming white noise. This cost function guided the estimation of factors along all the modes. Alternatively, prior knowledge or assumptions about the structure along the spatial mode could have been exploited. For instance, each channel may be modeled as the linear combination of a set of independent sources. If this is the case, then blind-source separation techniques, specifically independent component analysis (ICA), can be used to resolve the spatial mode's factors (Zhou & Cichocki, 2012). If these are the correct factors for the tensor model, then they can be used to demix the signal. Then each demixed signal is approximated by a rank-1 tensor, which is a much easier problem (Zhou & Cichocki, 2012). However, the model will not be correct if the the temporal waveforms are allowed to shift. Thus, future work can explore combining the approaches of BSS for tensor estimation with the shift-varying approaches discussed here.

Along different lines, Niknazar et al. (2014) proposed using a cost function very similar to correntropy cost (Liu et al., 2006) as another alternative for mean squared error. In the same work, the authors also proposed a smoothing estimator for the single-trial potential using the extended Kalman filter, which further decreases the noise.

6.4 Kernel-based Metric Learning

From the results in Chapter 5, it is apparent that the kernel-based metric learning is a powerful and flexible tool for neural decoding. The foremost advantage is its ability to be applied to spike trains and even heterogeneous activity via weighted product kernels.

Many metric-learning algorithms optimize a Mahalanobis metric, which is equivalent to taking a linear transformation of the data—projecting the data onto a lower dimensional subspace. In contrast, the weighted product kernel does not require the data space to be endowed with linear operations and allows the full joint information across the dimensions to be used. In addition, avoiding the projection decreases the computational cost when all of the distances can be pre-computed. Another benefit of the weighted product kernel is the straight-forward interpretation of its weights. It is much more difficult to interpret high-rank linear projections of a Mahalanobis-based metric. From the results it appears that the weighted product optimization often returns a very sparse set of weights. Thus, it is able to both select a subset of important dimensions and weight their contribution appropriately to maximize the mutual information with the target variable.

The use of a weighted product kernel for neural decoding builds from previous work using joint kernels (Li et al., 2012), particularly the tensor-product and direct-sum kernels. The main benefit of the weighted product kernel over a tensor-product is its ability to minimize the contribution of noisy dimensions. It should be noted that the weighted product can only be applied when using infinitely divisible kernel functions. For instance, the weighted product kernel cannot be used to combine multiple direct sum kernels. Nonetheless, a direct sum of weighted product kernels is always positive definite and was shown to improve performance. The optimal choice of the number of weighted product kernels in the summation still needs to be investigated. One interesting approach would be to choose different subsets of the inputs for each weighted product kernel. A group of subsets could be chosen *a priori* or randomly,

and the weights for each set of inputs could be optimized within the kernel-based framework. Future work should investigate the performance of the framework for continuous regression problems.

6.5 One-stage Does Not Fit All

The methods explored in this dissertation are in stark contrast with one-stage decoding where the raw signal is mapped directly to the output. For instance, non-parametric classifiers or regression algorithms such as k-nearest neighbor (kNN) and support vector machines (SVM) rely on the underlying distribution of the data, and a reasonable similarity metric. These methods do not learn the relative importance nor exploit any correlation of the dimensions; they are equally influenced by all dimensions and cannot separate signals from background noise.

In the case of kNN classifiers, every sample in the training set gets equal weight and a good metric is essential; whereas, in the case of support vector machines, samples themselves are weighted, for low-dimensional problems the similarity metric does not need to be perfectly tuned. Interestingly, tuning a metric for improving the kNN classification rate does not increase the SVM performance (Xu et al., 2012); however, the dimensionality of the samples may not have been considered. In small-scale benchmarks (with less than a hundred variables) found in the UCI dataset (Bache & Lichman, 2013), a Gaussian kernel with cross-validated kernel size outperforms k-NN performance with a tuned metric. However, we have shown, in Chapter 5, that this is not the case for high-dimensional neural data.

One-stage learning works well if the data is already in a reasonable representation. In the early days of face recognition, it was accepted to preprocess the images to the point that faces were aligned to the same size and nearly the same viewing angle, much like a passport photo. However, modern face recognition systems are expected to identify faces in candid photos, in a variety of poses. The more robust classification systems have benefited from the deep neural networks paradigms (Bengio, 2009;

Bengio et al., 2007; Hinton et al., 2006; Larochelle et al., 2009; Lee et al., 2009; Ranzato et al., 2007). Fundamentally, face recognition benefits from the ubiquity of examples and our innate ability to identify faces—providing a sanity check on the recognition results. In addition, face recognition is essentially an object recognition problem: the head itself is in physical space and the problem is to be learn a representation invariant to the pose and lighting.

However, these aspects are not the case in neural data. Signals are indicators of dynamics of the electrochemical reactions of the cells. The goal is not invariance, except it terms of time delays, as the signals are not simply a representation of some fixed form. In addition, for neural data there is a limited number of trials of any experiment, the brain is constantly adapting, and the correct classification of neural signals is not visually obvious. With these limitations, can the recent successes of deep learning be translated to neural data analysis, and will it have a transformative impact?

Perhaps, but for the broader neuroscience community, neural data analysis should provide an opportunity to better analyze the intrinsic structure of neural signals, rather than simply differentiating specific conditions. For many applications, it is necessary to understand what exactly delineated the conditions. In this view, the intermediate descriptors proposed in this dissertation are progress in the right direction.

REFERENCES

Acar, E., Aykut-Bingol, C., Bingol, H., Bro, R., & Yener, B. (2007). Multiway analysis of epilepsy tensors. *Bioinformatics*, *23*(13), i10–i18.

Achtman, N., Afshar, A., Santhanam, G., Byron, M. Y., Ryu, S. I., & Shenoy, K. V. (2007). Free-paced high-performance brain–computer interfaces. *Journal of Neural Engineering*, *4*(3), 336.

Adrian, E. D. (1926). The impulses produced by sensory nerve endings: Part I. *The Journal of Physiology*, *61*(1), 49–72.

Adrian, E. D., & Matthews, B. H. (1934). The Berger rhythm: Potential changes from the occipital lobes in man. *Brain*, *57*(4), 355–385.

Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, *54*(11), 4311–4322.

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

Aronov, D. (2003). Fast algorithm for the metric-space analysis of simultaneous responses of multiple single neurons. *Journal of Neuroscience Methods*, *124*(2), 175–179.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, *68*(3), 337–404.

Bach, F. R., & Jordan, M. I. (2003). Kernel independent component analysis. *The Journal of Machine Learning Research*, *3*, 1–48.

Bache, K., & Lichman, M. (2013). UCI machine learning repository. Available at http://archive.ics.uci.edu/ml.

Bader, B. W., Kolda, T. G., et al. (2012). Matlab tensor toolbox version 2.5. Software available at http://www.sandia.gov/~tgkolda/TensorToolbox.

Bae, J., Giraldo, L., Chhatbar, P., Francis, J., Sanchez, J., & Principe, J. (2011). Stochastic kernel temporal difference for reinforcement learning. In *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on*, (pp. 1–6).

Balcan, D. C., Lewicki, M. S., et al. (2009). Point coding: Sparse image representation with adaptive shiftable-kernel dictionaries. In *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*.

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, *12*(10), 2385–2404.

Bedard, C., Kroeger, H., & Destexhe, A. (2006). Does the 1/f frequency scaling of brain signals reflect self-organized critical states? *Physical Review Letters*, *97*(11), 118102.

Bell, A., & Sejnowski, T. (1996). Learning the higher-order structure of a natural sound. *Network: Computation in Neural Systems*, *7*(2), 261–266.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, *2*(1), 1–127.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems*, *19*, 153.

Berger, H. (1929). Über das elektrenkephalogramm des menschen. *European Archives of Psychiatry and Clinical Neuroscience*, *87*(1), 527–570.

Blankertz, B., Muller, K., Krusienski, D. J., Schalk, G., Wolpaw, J. R., Schlogl, A., Pfurtscheller, G., Millan, J. R., Schroder, M., & Birbaumer, N. (2006). The BCI competition III: Validating alternative approaches to actual BCI problems. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, *14*(2), 153–159.

Blumensath, T., & Davies, M. (2006). Sparse and shift-invariant representations of music. *Audio, Speech, and Language Processing, IEEE Transactions on*, *14*(1), 50–57.

Bobin, J., Starck, J.-L., Fadili, J. M., Moudden, Y., & Donoho, D. L. (2007). Morphological component analysis: An adaptive thresholding strategy. *Image Processing, IEEE Transactions on*, *16*(11), 2675–2681.

Bro, R., & Kiers, H. A. L. (2003). A new efficient method for determining the number of components in PARAFAC models. *Journal of Chemometrics*, *17*(5), 274–286.

Brockmeier, A. J., Choi, J. S., Emigh, M. M., Francis, J. T., & Principe, J. C. (2012a). Subspace matching thalamic microstimulation to tactile evoked potentials in rat somatosensory cortex. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, (pp. 2957–2960).

Brockmeier, A. J., Choi, J. S., Kriminger, E. G., Francis, J. T., & Principe, J. C. (2014). Neural decoding with kernel-based metric learning. *Neural Computation*, *26*(6).

Brockmeier, A. J., Giraldo, L. G., Choi, J. S., Francis, J. T., & Principe, J. C. (2013a). Learning multiscale neural metrics via entropy minimization. In *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*, (pp. 247–250).

Brockmeier, A. J., Hazrati, M. K., Freeman, W. J., & Principe, J. C. (2012b). Locating spatial patterns of waveforms during sensory perception in scalp EEG. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, (pp. 2531–2534).

Brockmeier, A. J., Kriminger, E., Sanchez, J. C., & Principe, J. C. (2011a). Latent state visualization of neural firing rates. In *Neural Engineering (NER), 2011 5th International IEEE/EMBS Conference on*, (pp. 144–147).

Brockmeier, A. J., Mahmoudi, B., Sanchez, J. C., & Principe, J. C. (2011b). Efficient temporal decomposition of local field potentials. In *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on*, (pp. 1–6).

Brockmeier, A. J., Park, I., Mahmoudi, B., Sanchez, J. C., & Principe, J. C. (2010). Spatio-temporal clustering of firing rates for neural state estimation. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, (pp. 6023–6026).

Brockmeier, A. J., & Príncipe, J. C. (2013). Decoding algorithms for brain machine interfaces. In B. He (Ed.) *Neural engineering*, (pp. 23–257). Springer.

Brockmeier, A. J., Principe, J. C., Phan, A. H., & Cichocki, A. (2013b). A greedy algorithm for model selection of tensor decompositions. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, (pp. 6113–6117).

Brockmeier, A. J., Sanchez Giraldo, L. G., Emigh, M. S., Bae, J., Choi, J. S., Francis, J. T., & Principe, J. C. (2013c). Information-theoretic metric learning: 2–D linear projections of neural data for visualization. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, (pp. 5586–5589).

Broome, B. M., Jayaraman, V., & Laurent, G. (2006). Encoding and decoding of overlapping odor sequences. *Neuron*, *51*(4), 467–482.

Brown, E. N., Barbieri, R., Ventura, V., Kass, R. E., & Frank, L. M. (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation*, *14*(2), 325–346.

Brown, E. N., Frank, L. M., Tang, D., Quirk, M. C., & Wilson, M. A. (1998). A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *The Journal of Neuroscience*, *18*(18), 7411–7425.

Bunte, K., Biehl, M., & Hammer, B. (2012). A general framework for dimensionality-reducing data visualization mapping. *Neural Computation*, *24*(3), 771–804.

Buzsáki, G., Anastassiou, C. A., & Koch, C. (2012). The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes. *Nature Reviews Neuroscience*, *13*(6), 407–420.

Buzsáki, G., & Draguhn, A. (2004). Neuronal oscillations in cortical networks. *Science*, *304*(5679), 1926–1929.

Canolty, R. T., Ganguly, K., Kennerley, S. W., Cadieu, C. F., Koepsell, K., Wallis, J. D., & Carmena, J. M. (2010). Oscillatory phase coupling coordinates anatomically dispersed functional cell assemblies. *Proceedings of the National Academy of Sciences*, *107*(40), 17356–17361.

Carmena, J., Lebedev, M., Crist, R., O'Doherty, J., Santucci, D., Dimitrov, D., Patil, P., Henriquez, C., & Nicolelis, M. (2003). Learning to control a brain–machine interface for reaching and grasping by primates. *PLoS Biology*, *1*(2), e42.

Carroll, J. D., & Chang, J.-J. (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, *35*(3), 283–319.

Ceulemans, E., & Kiers, H. A. L. (2006). Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. *British Journal of Mathematical and Statistical Psychology*, *59*(1), 133–150.

Ceulemans, E., & Kiers, H. A. L. (2009). Discriminating between strong and weak structures in three-mode principal component analysis. *British Journal of Mathematical and Statistical Psychology*, *62*(3), 601–620.

Chalasani, R., Principe, J. C., & Ramakrishnan, N. (2013). A fast proximal method for convolutional sparse coding. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, (pp. 1–5).

Chandrasekaran, R., & Tamir, A. (1989). Open questions concerning Weiszfeld's algorithm for the Fermat-Weber location problem. *Mathematical Programming*, *44*(1-3), 293–295.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chapin, J. K., & Nicolelis, M. A. (1999). Principal component analysis of neuronal ensemble activity reveals multidimensional somatosensory representations. *Journal of Neuroscience Methods*, *94*(1), 121–140.

Chen, S. S., Donoho, D. L., & Saunders, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, *20*(1), 33–61.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *17*(8), 790–799.

Chhatbar, P. Y., von Kraus, L. M., Semework, M., & Francis, J. T. (2010). A bio-friendly and economical technique for chronic implantation of multiple microelectrode arrays. *Journal of Neuroscience Methods*, *188*(2), 187–194.

Choi, J. S., DiStasio, M. M., Brockmeier, A. J., & Francis, J. T. (2012). An electric field model for prediction of somatosensory (S1) cortical field potentials induced by ventral posterior lateral (VPL) thalamic microstimulation. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, *20*(2), 161–169.

Christoforou, C., Haralick, R., Sajda, P., & Parra, L. C. (2010). Second-order bilinear discriminant analysis. *The Journal of Machine Learning Research*, *11*, 665–685.

Churchland, M., Yu, B., Sahani, M., & Shenoy, K. (2007). Techniques for extracting single-trial activity patterns from large-scale neural recordings. *Current Opinion in Neurobiology*, *17*(5), 609–618.

Churchland, M. M., Yu, B. M., Cunningham, J. P., Sugrue, L. P., Cohen, M. R., Corrado, G. S., Newsome, W. T., Clark, A. M., Hosseini, P., Scott, B. B., Bradley, D. C., Smith, M. A., Kohn, A., Movshon, J. A., Armstrong, K. M., Moore, T., Chang, S. W., Snyder, L. H., Lisberger, S. G., Priebe, N. J., Finn, I. M., Ferster, D., Ryu, S. I., Santhanam, G., Sahani, M., & Shenoy, K. V. (2010). Stimulus onset quenches neural variability: A widespread cortical phenomenon. *Nature Neuroscience*, *13*(3), 369–378.

Cichocki, A., Washizawa, Y., Rutkowski, T., Bakardjian, H., Phan, A., Choi, S., Lee, H., Zhao, Q., Zhang, L., & Li, Y. (2008). Noninvasive BCIs: Multiway signal-processing array decompositions. *Computer*, *41*(10), 34–42.

Ciganek, L. (1961). The EEG response (evoked potential) to light stimulus in man. *Electroencephalography and Clinical Neurophysiology*, *13*(2), 165–172.

Ciganek, L. (1969). Variability of the human visual evoked potential: Normative data. *Electroencephalography and Clinical Neurophysiology*, *27*(1), 35–42.

Clarke, K. A. (2007). A simple distribution-free test for nonnested model selection. *Political Analysis*, *15*(3), 347–363.

Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing*, *36*(3), 287–314.

Cortes, C., Mohri, M., & Rostamizadeh, A. (2012). Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, *13*, 795–828.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, *47*(4), 547–553.

Cowley, B., Kaufman, M., Churchland, M., Ryu, S., Shenoy, K., & Yu, B. (2012). DataHigh: Graphical user interface for visualizing and interacting with

high-dimensional neural activity. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, (pp. 4607–4610).

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., & Kandola, J. (2002). On kernel-target alignment. In T. Dietterich, S. Becker, & Z. Ghahramani (Eds.) *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press.

Dauwels, J., Srinivasan, K., Ramasubba Reddy, M., & Cichocki, A. (2011). Multi-channel EEG compression based on matrix and tensor decompositions. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, (pp. 629–632).

Davidson, P. R., Jones, R., & Peiris, M. T. R. (2007). EEG-based lapse detection with high temporal resolution. *Biomedical Engineering, IEEE Transactions on*, *54*(5), 832–839.

Davies, M., & James, C. (2007). Source separation using single channel ICA. *Signal Processing*, *87*(8), 1819–1832.

De la Torre, F., & Kanade, T. (2005). Multimodal oriented discriminant analysis. In *Proceedings of the 22nd International Conference on Machine learning*, (pp. 177–184). ACM.

De Lathauwer, L. (2008). Decompositions of a higher-order tensor in block terms-Part II: Definitions and uniqueness. *SIAM Journal on Matrix Analysis and Applications*, *30*(3), 1033–1066.

de Munck, J. C., Bijma, F., Gaura, P., Sieluzycki, C. A., Branco, M. I., & Heethaar, R. M. (2004). A maximum-likelihood estimator for trial-to-trial variations in noisy MEG/EEG data sets. *Biomedical Engineering, IEEE Transactions on*, *51*(12), 2123–2128.

de Ruyter van Steveninck, R. R., Lewen, G. D., Strong, S. P., Koberle, R., & Bialek, W. (1997). Reproducibility and variability in neural spike trains. *Science*, *275*(5307), 1805–1808.

Delorme, A., Palmer, J., Onton, J., Oostenveld, R., & Makeig, S. (2012). Independent EEG sources are dipolar. *PLoS ONE*, *7*(2), e30135.

Douglas, S. C., Gupta, M., Sawada, H., & Makino, S. (2007). Spatio–temporal FastICA algorithms for the blind separation of convolutive mixtures. *Audio, Speech, and Language Processing, IEEE Transactions on*, *15*(5), 1511–1520.

Dubbs, A. J., Seiler, B. A., & Magnasco, M. O. (2010). A fast $\mathcal{L}_p$ spike alignment metric. *Neural Computation*, *22*(11), 2785–2808.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York: John Wiley & Sons, second ed.

Durka, P., & Blinowska, K. (1995). Analysis of EEG transients by means of matching pursuit. *Annals of Biomedical Engineering*, *23*(5), 608–611.

Durka, P. J., Ircha, D., & Blinowska, K. J. (2001). Stochastic time-frequency dictionaries for matching pursuit. *Signal Processing, IEEE Transactions on*, *49*(3), 507–510.

Dyrholm, M., Christoforou, C., & Parra, L. C. (2007). Bilinear discriminant component analysis. *The Journal of Machine Learning Research*, *8*, 1097–1111.

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, *1*(3), 211–218.

Eichhorn, J., Tolias, A., Zien, A., Kuss, M., Rasmussen, C. E., Weston, J., Logothetis, N., & Schölkopf, B. (2004). Prediction on spike data using kernel algorithms. In S. Thrun, L. Saul, & B. Schölkopf (Eds.) *Advances in Neural Information Processing Systems 16*. Cambridge, MA: MIT Press.

Ekanadham, C., Tranchina, D., & Simoncelli, E. (2011). Recovery of sparse translation-invariant signals with continuous basis pursuit. *Signal Processing, IEEE Transactions on*, *59*(10), 4735–4744.

Elad, M., Starck, J.-L., Querre, P., & Donoho, D. L. (2005). Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis*, *19*(3), 340– 358.

Emery, J. D., & Freeman, W. J. (1969). Pattern analysis of cortical evoked potential parameters during attention changes. *Physiology & Behavior*, *4*(1), 69–77.

Farwell, L. A., & Donchin, E. (1988). Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, *70*(6), 510–523.

Freeman, W. J. (1975). *Mass action in the nervous system*. Academic Press New York.

Freeman, W. J. (1979). Measurement of cortical evoked potentials by decomposition of their wave forms. *Journal of Cybernetics and Information Science*, *2*, 44–56.

Freeman, W. J. (2004). Origin, structure, and role of background EEG activity. Part 1. Analytic amplitude. *Clinical Neurophysiology*, *115*(9), 2077–2088.

Fritz, J., Shamma, S., Elhilali, M., & Klein, D. (2003). Rapid task-related plasticity of spectrotemporal receptive fields in primary auditory cortex. *Nature Neuroscience*, *6*(11), 1216–1223.

Fukumizu, K., Bach, F. R., & Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *The Journal of Machine Learning Research*, *5*, 73–99.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. Academic Press.

Fukunaga, K., & Koontz, W. L. (1970). Application of the Karhunen-Loève expansion to feature selection and ordering. *Computers, IEEE Transactions on*, *100*(4), 311–318.

Gärtner, T., Flach, P., & Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, (pp. 129–143). Springer.

Gat, I., Tishby, N., & Abeles, M. (1997). Hidden Markov modelling of simultaneously recorded cells in the associative cortex of behaving monkeys. *Network: Computation in Neural Systems*, *8*(3), 297–322.

Georgiev, P., Theis, F., & Cichocki, A. (2005). Sparse component analysis and blind source separation of underdetermined mixtures. *Neural Networks, IEEE Transactions on*, *16*(4), 992–996.

Gerstein, G. L., & Kiang, N. Y.-S. (1960). An approach to the quantitative analysis of electrophysiological data from single neurons. *Biophysical Journal*, *1*(1), 15–28.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. In S. Jain, H. U. Simon, & E. Tomita (Eds.) *Algorithmic Learning Theory*, vol. 3734 of *Lecture Notes in Computer Science*, (pp. 63–77). Springer Berlin Heidelberg.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, *3*, 1157–1182.

Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, *16*, 1–84.

Hartigan, J., & Wong, M. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100–108.

He, Z., Cichocki, A., & Xie, S. (2009). Efficient method for Tucker3 model selection. *Electronics Letters*, *45*(15), 805–806.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hinton, G. E., & Zemel, R. S. (1993). Autoencoders, minimum description length and Helmholtz free energy. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.) *Advances in Neural Information Processing Systems 6*, (pp. 3–10).

Hitchcock, F. L. (1927). Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics*, *7*(1), 39–79.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, *117*(4), 500–544.

Horn, R. A. (1967). On infinitely divisible matrices, kernels, and functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, *8*(3), 219–230.

Houghton, C., & Sen, K. (2008). A new multineuron spike train metric. *Neural Computation*, *20*(6), 1495–1511.

Huang, N., Shen, Z., Long, S., Wu, M., Shih, H., Zheng, Q., Yen, N., Tung, C., & Liu, H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *454*(1971), 903–995.

Huber, P. J. (1985). Projection pursuit. *The Annals of Statistics*, (pp. 435–475).

Hyvarinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *Neural Networks, IEEE Transactions on*, *10*(3), 626–634.

Jaskowski, P., & Verleger, R. (1999). Amplitudes and latencies of single-trial ERP's estimated by a maximum-likelihood method. *Biomedical Engineering, IEEE Transactions on*, *46*(8), 987–993.

Jeong, J. (2004). EEG dynamics in patients with Alzheimer's disease. *Clinical Neurophysiology*, *115*(7), 1490–1505.

Jmail, N., Gavaret, M., Wendling, F., Kachouri, A., Hamadi, G., Badier, J.-M., & Benar, C.-G. (2011). A comparison of methods for separation of transient and oscillatory signals in EEG. *Journal of Neuroscience Methods*, *199*(2), 273–289.

Jost, P., Vandergheynst, P., Lesage, S., & Gribonval, R. (2006). MoTIF: An efficient algorithm for learning translation invariant dictionaries. In *Acoustics, Speech and Signal Processing, IEEE International Conference on*, vol. 5, (pp. 857–860).

Jung, T., Makeig, S., Humphries, C., Lee, T., Mckeown, M., Iragui, V., & Sejnowski, T. (2000). Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, *37*(02), 163–178.

Jung, T.-P., Makeig, S., Stensmo, M., & Sejnowski, T. (1997). Estimating alertness from the EEG power spectrum. *Biomedical Engineering, IEEE Transactions on*, *44*(1), 60–69.

Karjalainen, P. A., Kaipio, J. P., Koistinen, A. S., & Vauhkonen, M. (1999). Subspace regularization method for the single-trial estimation of evoked potentials. *Biomedical Engineering, IEEE Transactions on*, *46*(7), 849–860.

Kass, R. E., & Ventura, V. (2001). A spike-train probability model. *Neural Computation*, *13*(8), 1713–1720.

Kavukcuoglu, K., Sermanet, P., Boureau, Y., Gregor, K., Mathieu, M., & LeCun, Y. (2010). Learning convolutional feature hierarchies for visual recognition. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, & A. Culotta (Eds.) *Advances in Neural Information Processing Systems 23*, (pp. 1090–1098).

Kemere, C., Santhanam, G., Byron, M. Y., Afshar, A., Ryu, S. I., Meng, T. H., & Shenoy, K. V. (2008). Detecting neural-state transitions using hidden Markov models for motor cortical prostheses. *Journal of Neurophysiology*, *100*(4), 2441–2452.

Kira, K., & Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, (pp. 129–134). AAAI Press.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480.

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, *51*(3), 455–500.

Koldovskỳ, Z., & Tichavskỳ, P. (2011). Time-domain blind separation of audio sources on the basis of a complete ICA decomposition of an observation space. *Audio, Speech, and Language Processing, IEEE Transactions on*, *19*(2), 406–416.

Kruskal, J. B. (1969). Toward a practical method which helps uncover the structure of a set of multivariate observations by finding the linear transformation which optimizes a new index of condensation. In *Statistical Computation*, (pp. 427–440). Academic Press, New York.

Kruskal, J. B. (1972). Linear transformation of multivariate data to reveal clustering. *Multidimensional Scaling: Theory and Applications in the Behavioural Sciences*, *1*, 179–191.

Kruskal, J. B. (1977). Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, *18*(2), 95–138.

Kuś, R., Różański, P. T., & Durka, P. J. (2013). Multivariate matching pursuit in optimal Gabor dictionaries: Theory and software with interface for EEG/MEG via Svarog. *Biomedical Engineering Online*, *12*(1), 94.

Lanckriet, G. R., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, *5*, 27–72.

Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2009). Exploring strategies for training deep neural networks. *The Journal of Machine Learning Research*, *10*, 1–40.

Lebedev, M. A., & Nicolelis, M. A. (2006). Brain-machine interfaces: Past, present and future. *Trends in Neurosciences*, *29*(9), 536–546.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, (pp. 609–616). ACM.

Lewicki, M. (2002). Efficient coding of natural sounds. *Nature Neuroscience*, *5*(4), 356–363.

Li, L., Choi, J. S., Francis, J. T., Sanchez, J. C., & Príncipe, J. C. (2012). Decoding stimuli from multi-source neural responses. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, (pp. 1331–1334). IEEE.

Li, R., Principe, J. C., Bradley, M., & Ferrari, V. (2009). A spatiotemporal filtering methodology for single-trial ERP component estimation. *Biomedical Engineering, IEEE Transactions on*, *56*(1), 83–92.

Liddell, E. G. T., & Sherrington, C. (1924). Reflexes in response to stretch (myotatic reflexes). *Proceedings of the Royal Society of London. Series B, Containing Papers of a Biological Character*, *96*(675), 212–242.

Little, M. A., McSharry, P. E., Roberts, S. J., Costello, D. A., & Moroz, I. M. (2007). Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Biomedical Engineering Online*, *6*(1), 23.

Liu, J., Oweiss, K., & Khalil, H. (2010). Feedback control of the spatiotemporal firing patterns of neural microcircuits. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, (pp. 4679–4684).

Liu, W., Pokharel, P., & Principe, J. (2006). Correntropy: A localized similarity measure. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, (pp. 4919–4924).

Liu, W., Pokharel, P., & Principe, J. (2007). Correntropy: Properties and applications in non-Gaussian signal processing. *Signal Processing, IEEE Transactions on*, *55*(11), 5286–5298.

Lloyd, S. (1982). Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, *28*(2), 129–137.

Lowe, D. G. (1995). Similarity metric learning for a variable-kernel classifier. *Neural Computation*, *7*(1), 72–85.

Lucena, F., Barros, A., Príncipe, J., & Ohnishi, N. (2011). Statistical coding and decoding of heartbeat intervals. *PLoS ONE*, *6*(6), e20227.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, (pp. 281–297). California, USA.

Mahmoudi, B., & Sanchez, J. C. (2011). A symbiotic brain-machine interface through value-based decision making. *PLoS ONE*, *6*(3), e14760.

Mailhé, B., Gribonval, R., Bimbot, F., Lemay, M., Vandergheynst, P., & Vesin, J.-M. (2009). Dictionary learning for the sparse modelling of atrial fibrillation in ECG signals. In *Acoustics, Speech and Signal Processing, IEEE International Conference on*, (pp. 465–468).

Mailhé, B., Lesage, S., Gribonval, R., Bimbot, F., & Vandergheynst, P. (2008). Shift-invariant dictionary learning for sparse representations: Extending K-SVD. In *16th European Signal Processing Conference*.

Mallat, S., & Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, *41*(12), 3397–3415.

Megela, A. L., & Teyler, T. J. (1979). Habituation and the human evoked potential. *Journal of Comparative and Physiological Psychology*, *93*(6), 1154.

Mijović and, B., De Vos, M., Gligorijević and, I., Taelman, J., & Van Huffel, S. (2010). Source separation from single-channel recordings by combining empirical-mode decomposition and independent component analysis. *Biomedical Engineering, IEEE Transactions on*, *57*(9), 2188–2196.

Miwakeichi, F., Martinez-Montes, E., Valdés-Sosa, P., Nishiyama, N., Mizuhara, H., & Yamaguchi, Y. (2004). Decomposing EEG data into space-time-frequency components using parallel factor analysis. *NeuroImage*, *22*(3), 1035–1045.

Mørup, M., Hansen, L., Herrmann, C., Parnas, J., & Arnfred, S. (2006). Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG. *NeuroImage*, *29*(3), 938–947.

Mørup, M., & Hansen, L. K. (2009). Automatic relevance determination for multi-way models. *Journal of Chemometrics*, *23*(7-8), 352–363.

Mørup, M., Hansen, L. K., Arnfred, S. M., Lim, L.-H., & Madsen, K. H. (2008). Shift-invariant multilinear decomposition of neuroimaging data. *NeuroImage*, *42*(4), 1439–1450.

Murata, N., Yoshizawa, S., & Amari, S.-I. (1994). Network information criterion—determining the number of hidden units for an artificial neural network model. *Neural Networks, IEEE Transactions on*, *5*(6), 865–872.

Navot, A., Shpigelman, L., Tishby, N., & Vaadia, E. (2006). Nearest neighbor based feature selection for regression and its application to neural activity. In Y. Weiss,

B. Schölkopf, & J. Platt (Eds.) *Advances in Neural Information Processing Systems 18*, (pp. 995–1002). Cambridge, MA: MIT Press.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.) *Advances in Neural Information Processing Systems 14*, (pp. 849–856). MIT Press.

Niknazar, M., Becker, H., Rivet, B., Jutten, C., & Comon, P. (2014). Blind source separation of underdetermined mixtures of event-related sources. *Signal Processing*, *101*(0), 52–64.

Novey, M., & Adali, T. (2008). Complex ICA by negentropy maximization. *Neural Networks, IEEE Transactions on*, *19*(4), 596–609.

Nunez, P., & Srinivasan, R. (2006). *Electric fields of the brain: The neurophysics of EEG*. Oxford University Press, USA.

O'Doherty, J. E., Lebedev, M. A., Ifft, P. J., Zhuang, K. Z., Shokur, S., Bleuler, H., & Nicolelis, M. A. (2011). Active tactile exploration using a brain-machine-brain interface. *Nature*, *479*(7372), 228–231.

O'Doherty, J. E., Lebedev, M. A., Li, Z., & Nicolelis, M. A. (2012). Virtual active touch using randomly patterned intracortical microstimulation. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, *20*(1), 85–93.

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, *37*(23), 3311–3325.

Paiva, A. R., Park, I., & Príncipe, J. C. (2009). A reproducing kernel Hilbert space framework for spike train signal processing. *Neural Computation*, *21*(2), 424–449.

Paiva, A. R., Park, I., & Príncipe, J. C. (2010). A comparison of binless spike train measures. *Neural Computing and Applications*, *19*(3), 405–419.

Paninski, L., Ahmadian, Y., Ferreira, D. G., Koyama, S., Rad, K. R., Vidne, M., Vogelstein, J., & Wu, W. (2010). A new look at state-space models for neural data. *Journal of Computational Neuroscience*, *29*(1-2), 107–126.

Papoulis, A. (1990). *Probability & statistics*, vol. 2. Prentice-Hall Englewood Cliffs.

Park, I. M., Meister, M., Huk, A., & Pillow, J. W. (2011). Detailed encoding and decoding of choice-related information from LIP spike trains. In *Frontiers in Systems Neuroscience. Conference Abstract: Computational and systems neuroscience (COSYNE)*.

Park, I. M., Seth, S., Paiva, A., Li, L., & Principe, J. (2013). Kernel methods on spike train space for neuroscience: A tutorial. *Signal Processing Magazine, IEEE*, *30*(4), 149–160.

Park, I. M., Seth, S., Rao, M., & Príncipe, J. C. (2012). Strictly positive-definite spike train kernels for point-process divergences. *Neural Computation*, *24*(8), 2223–2250.

Park, J., & Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. *Neural Computation*, *3*(2), 246–257.

Pati, Y., Rezaiifar, R., & Krishnaprasad, P. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, (pp. 40–44).

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *27*(8), 1226–1238.

Petreska, B., Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., & Sahani, M. (2011). Dynamical segmentation of single trials from population neural data. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Weinberger (Eds.) *Advances in Neural Information Processing Systems 24*, (pp. 756–764).

Pham, T. D., Möcks, J., Köhler, W., & Gasser, T. (1987). Variable latencies of noisy signals: Estimation and testing in brain potential data. *Biometrika*, *74*(3), 525–533.

Phan, A.-H., Tichavsky, P., & Cichocki, A. (2013a). Low complexity damped Gauss–Newton algorithms for CANDECOMP/PARAFAC. *SIAM Journal on Matrix Analysis and Applications*, *34*(1), 126–147.

Phan, A.-H., Tichavsky, P., & Cichocki, A. (2013b). TENSORBOX: A MATLAB package for tensor decomposition. Software available at http://www.bsp.brain.riken.jp/~phan/tensorbox.php.

Pillow, J. W., Ahmadian, Y., & Paninski, L. (2011). Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains. *Neural Computation*, *23*(1), 1–45.

Pohlmeyer, E. A., Mahmoudi, B., Geng, S., Prins, N. W., & Sanchez, J. C. (2014). Using reinforcement learning to provide stable brain-machine interface control despite neural input reorganization. *PLOS ONE*, *9*(1), e87253.

Principe, J. C. (2010). *Information theoretic learning: Rényi's entropy and kernel perspectives*. Springer.

Pritchard, W. S. (1992). The brain in fractal time: 1/f-like power spectrum scaling of the human electroencephalogram. *International Journal of Neuroscience*, *66*(1-2), 119–129.

Radons, G., Becker, J., Dülfer, B., & Krüger, J. (1994). Analysis, classification, and coding of multielectrode spike trains with hidden Markov models. *Biological Cybernetics*, *71*(4), 359–373.

Ramoser, H., Muller-Gerking, J., & Pfurtscheller, G. (2000). Optimal spatial filtering of single trial EEG during imagined hand movement. *Rehabilitation Engineering, IEEE Transactions on*, *8*(4), 441–446.

Ranzato, M., Huang, F. J., Boureau, Y.-L., & Lecun, Y. (2007). Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, (pp. 1–8). IEEE.

Rao, M., Seth, S., Xu, J., Chen, Y., Tagare, H., & Príncipe, J. C. (2011). A test of independence based on a generalized correlation function. *Signal Processing*, *91*(1), 15–27.

Ray, W. J., & Cole, H. W. (1985). EEG alpha activity reflects attentional demands, and beta activity reflects emotional and cognitive processes. *Science*, *228*(4700), 750–752.

Rieke, F. (1999). *Spikes: Exploring the neural code*. The MIT Press.

Rivet, B., Souloumiac, A., Attina, V., & Gibert, G. (2009). xDAWN algorithm to enhance evoked potentials: Application to brain–computer interface. *Biomedical Engineering, IEEE Transactions on*, *56*(8), 2035–2043.

Romo, R., Hernndez, A., Zainos, A., Brody, C. D., & Lemus, L. (2000). Sensing without touching: Psychophysical performance based on cortical microstimulation. *Neuron*, *26*(1), 273–278.

Roux, J. L., Cheveigné, A. D., & Parra, L. C. (2009). Adaptive template matching with shift-invariant semi-NMF. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.) *Advances in Neural Information Processing Systems 21*, (pp. 921–928).

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Rusu, C. V., & Florian, R. V. (2013). A new class of metrics for spike trains. *Neural Computation*, *26*(2), 306–348.

Sammon Jr, J. (1969). A nonlinear mapping for data structure analysis. *Computers, IEEE Transactions on*, *100*(5), 401–409.

Sanchez Giraldo, L. G., & Principe, J. C. (2013). Information theoretic learning with infinitely divisible kernels. In *International Conference on Learning Representations*. arXiv:1301.3551v6.

Sanchez Giraldo, L. G., Rao, M., & Principe, J. C. (2012). Measures of entropy from data using infinitely divisible kernels. arXiv:1211.2459.

Santhanam, G., Ryu, S., Yu, B., Afshar, A., & Shenoy, K. (2006). A high-performance brain–computer interface. *Nature*, *442*(7099), 195–198.

Schalk, G., McFarland, D., Hinterberger, T., Birbaumer, N., & Wolpaw, J. (2004). BCI2000: a general-purpose brain-computer interface (BCI) system. *Biomedical Engineering, IEEE Transactions on*, *51*(6), 1034–1043.

Schmidt, E. M., Bak, M. J., Hambrecht, F. T., Kufta, C. V., O'Rourke, D. K., & Vallabhanath, P. (1996). Feasibility of a visual prosthesis for the blind based on intracortical microstimulation of the visual cortex. *Brain*, *119*(2), 507–522.

Schmidt, M. (2012). minFunc. Software available at http://www.di.ens.fr/~mschmidt/Software/minFunc.html.

Schoenberg, I. J. (1938). Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, *44*(3), 522–536.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*(5), 1299–1319.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. The MIT Press.

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

Seidemann, E., Meilijson, I., Abeles, M., Bergman, H., & Vaadia, E. (1996). Simultaneously recorded single units in the frontal cortex go through sequences of discrete and stable states in monkeys performing a delayed localization task. *Journal of Neuroscience*, *16*(2), 752–768.

Sejnowski, T., & Paulsen, O. (2006). Network oscillations: Emerging computational principles. *The Journal of Neuroscience*, *26*(6), 1673–1676.

Shalvi, O., & Weinstein, E. (1990). New criteria for blind deconvolution of nonminimum phase systems (channels). *Information Theory, IEEE Transactions on*, *36*(2), 312–321.

Sharpee, T., Rust, N. C., & Bialek, W. (2004). Analyzing neural responses to natural signals: Maximally informative dimensions. *Neural Computation*, *16*(2), 223–250.

Shenoy, K. V., Meeker, D., Cao, S., Kureshi, S. A., Pesaran, B., Buneo, C. A., Batista, A. P., Mitra, P. P., Burdick, J. W., & Andersen, R. A. (2003). Neural prosthetic control signals from plan activity. *Neuroreport*, *14*(4), 591–596.

Shimazaki, H., & Shinomoto, S. (2007). A method for selecting the bin size of a time histogram. *Neural Computation*, *19*(6), 1503–1527.

Shpigelman, L., Singer, Y., Paz, R., & Vaadia, E. (2005). Spikernels: Predicting arm movements by embedding population spike rate patterns in inner-product spaces. *Neural Computation*, *17*(3), 671–690.

Sinz, F., Stockl, A., Grewe, J., & Benda, J. (2013). Least informative dimensions. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Weinberger (Eds.) *Advances in Neural Information Processing Systems 26*, (pp. 413–421).

Smaragdis, P., Raj, B., & Shashanka, M. (2008). Sparse and shift-invariant feature extraction from non-negative data. In *Acoustics, Speech and Signal Processing, IEEE International Conference on*, (pp. 2069–2072).

Smith, E., & Lewicki, M. (2006). Efficient auditory coding. *Nature*, *439*(7079), 978–982.

Song, L., Bedo, J., Borgwardt, K. M., Gretton, A., & Smola, A. (2007). Gene selection via the BAHSIC family of algorithms. *Bioinformatics*, *23*(13), i490–i498.

Song, L., Smola, A., Gretton, A., Bedo, J., & Borgwardt, K. (2012). Feature selection via dependence maximization. *The Journal of Machine Learning Research*, *13*, 1393–1434.

Souloumiac, A., & Rivet, B. (2013). Improved estimation of EEG evoked potentials by jitter compensation and enhancing spatial filters. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, (pp. 1222–1226). IEEE.

Spielman, D. A., & Teng, S.-H. (2007). Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, *421*(2), 284–305.

Stegeman, A., & Comon, P. (2010). Subtracting a best rank-1 approximation may increase tensor rank. *Linear Algebra and its Applications*, *433*(7), 1276–1300.

Stoica, P., & Selén, Y. (2004). Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: A refresher. *Signal Processing Magazine, IEEE*, *21*(1), 112–114.

Stoica, P., & Selen, Y. (2004). Model-order selection: A review of information criterion rules. *Signal Processing Magazine, IEEE*, *21*(4), 36–47.

Stopfer, M., Jayaraman, V., & Laurent, G. (2003). Intensity versus identity coding in an olfactory system. *Neuron*, *39*(6), 991–1004.

Sugiyama, M. (2007). Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. *The Journal of Machine Learning Research*, *8*, 1027–1061.

Takeuchi, I., & Sugiyama, M. (2011). Target neighbor consistent feature weighting for nearest neighbor classification. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Weinberger (Eds.) *Advances in Neural Information Processing Systems 24*, (pp. 576–584).

Tenenbaum, J. B., De Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*(5500), 2319–2323.

Timmerman, M. E., & Kiers, H. A. L. (2000). Three-mode principal components analysis: Choosing the numbers of components and sensitivity to local optima. *British Journal of Mathematical and Statistical Psychology*, *53*(1), 1–16.

Tkačik, G., Granot-Atedgi, E., Segev, R., & Schneidman, E. (2013). Retinal metric: A stimulus distance measure derived from population neural responses. *Physical Review Letters*, *110*(5), 058104.

Torgerson, W. S. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*, *17*(4), 401–419.

Truccolo, W., Knuth, K. H., Shah, A., Bressler, S. L., Schroeder, C. E., & Ding, M. (2003). Estimation of single-trial multicomponent ERPs: Differentially variable component analysis (dVCA). *Biological Cybernetics*, *89*(6), 426–438.

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, *31*(3), 279–311.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

van Hateren, J. H., & van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *265*(1394), 359–366.

van Rossum, M. C. W. (2001). A novel spike distance. *Neural Computation*, *13*(4), 751–763.

Victor, J. D. (2005). Spike train metrics. *Current Opinion in Neurobiology*, *15*(5), 585–592.

Victor, J. D., & Purpura, K. P. (1996). Nature and precision of temporal coding in visual cortex: a metric-space analysis. *Journal of Neurophysiology*, *76*(2), 1310–1326.

Vuong, Q. H. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica: Journal of the Econometric Society*, (pp. 307–333).

Weeda, W. D., Grasman, R. P., Waldorp, L. J., van de Laar, M. C., Van der Molen, M. W., & Huizenga, H. M. (2012). A fast and reliable method for simultaneous waveform, amplitude and latency estimation of single-trial EEG/MEG data. *PLoS ONE*, *7*(6), e38292.

Weinberger, K., Blitzer, J., & Saul, L. (2006). Distance metric learning for large margin nearest neighbor classification. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.) *Advances in Neural Information Processing Systems 18*, (pp. 1473–1480). Cambridge, MA: MIT Press.

Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, *10*, 207–244.

Weinberger, K. Q., & Tesauro, G. (2007). Metric learning for kernel regression. In *International Conference on Artificial Intelligence and Statistics*, (pp. 612–619).

Wessberg, J., Stambaugh, C., Kralik, J., Beck, P., Laubach, M., Chapin, J., Kim, J., Biggs, S., Srinivasan, M., & Nicolelis, M. (2000). Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, *408*(6810), 361–365.

Woody, C. D. (1967). Characterization of an adaptive filter for the analysis of variable latency neuroelectric signals. *Medical and Biological Engineering*, *5*(6), 539–554.

Wu, W., Black, M., Mumford, D., Gao, Y., Bienenstock, E., & Donoghue, J. (2004). Modeling and decoding motor cortical activity using a switching Kalman filter. *Biomedical Engineering, IEEE Transactions on*, *51*(6), 933–942.

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2003). Distance metric learning with application to clustering with side-information. In S. Becker, S. Thrun, & K. Obermayer (Eds.) *Advances in Neural Information Processing Systems 15*, (pp. 505–512). Cambridge, MA: MIT Press.

Xu, Z., Weinberger, K. Q., & Chapelle, O. (2012). Distance metric learning for kernel machines. arXiv:1208.3422.

Xydas, D., Downes, J. H., Spencer, M. C., Hammond, M. W., Nasuto, S. J., Whalley, B. J., Becerra, V. M., & Warwick, K. (2011). Revealing ensemble state transition patterns in multi-electrode neuronal recordings using hidden markov models. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, *19*(4), 345–355.

Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., & Sugiyama, M. (2013). High-dimensional feature selection by feature-wise kernelized lasso. *Neural Computation*, *26*(1), 185–207.

Yu, B., Afshar, A., Santhanam, G., Ryu, S. I., Shenoy, K., & Sahani, M. (2006). Extracting dynamical structure embedded in neural activity. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.) *Advances in Neural Information Processing Systems 18*, (pp. 1545–1552). Cambridge, MA: MIT Press.

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., & Sahani, M. (2009). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.) *Advances in Neural Information Processing Systems 21*, (pp. 1881–1888).

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., & Sahani, M. (July 2009). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, *102*(1), 614–635.

Zhang, S., & Sim, T. (2007). Discriminant subspace analysis: A Fukunaga-Koontz approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *29*(10), 1732–1745.

Zhou, G., & Cichocki, A. (2012). Canonical polyadic decomposition based on a single mode blind source separation. *Signal Processing Letters, IEEE*, *19*(8), 523–526.

Zibulevsky, M., & Pearlmutter, B. A. (2001). Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation*, *13*(4), 863–882.

BIOGRAPHICAL SKETCH

Austin J. Brockmeier was born in western Nebraska in 1986. Austin graduated from Cozad High School in May 2005. He attended the University of Nebraska at Omaha where he was a Walter Scott, Jr. Scholar at the Peter Kiewit Institute. In May 2009, he received his Bachelor of Science degree in computer engineering from the University of Nebraska–Lincoln, via the Omaha campus, with a second major in mathematics and a minor in computer science. During his undergraduate studies, Austin held a two-year information technology internship with Union Pacific Railroad, was involved in undergraduate research, and worked with a technology start-up company developing neonatal health monitoring prototypes. He also served as a teaching assistant for digital logic and computer design.

In August 2009, Austin enrolled at the University of Florida to pursue a Ph.D. in the Electrical and Computer Engineering Department. He was a teaching assistant for microprocessor applications for two semesters before becoming a Research Assistant under Dr. José Príncipe. He had the opportunity to work with a collaborative team of researchers developing brain-machine interfaces. His role was developing novel statistical signal processing and machine learning algorithms and applying them to the analysis of neural data. Austin received his Ph.D. in Electrical and Computer Engineering in May 2014.

In 2012, Austin was awarded a NSF Fellowship for a summer research stay in Japan. He was hosted in the laboratory of Dr. Andrzej Cichocki at RIKEN Brain Science Center. During his stay he researched algorithms for brainwave analysis using tensor decompositions. Austin was also an active volunteer during his PhD studies, serving for multiple years as a middle school science fair judge, filling a graduate Student Senate seat, and working as a student volunteer at three international conferences.