Gaussian Process Kernels for Cross-Spectrum Analysis in Electrophysiological Time Series

by

Kyle Ulrich

Department of Electrical and Computer Engineering
Duke University

Date: ______________________
Approved:

Lawrence Carin, Supervisor

Guillermo Sapiro

Robert Calderbank

Katherine Heller

Kafui Dzirasa

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering in the Graduate School of Duke University
2016
ABSTRACT

Gaussian Process Kernels for Cross-Spectrum Analysis in Electrophysiological Time Series

by

Kyle Ulrich

Department of Electrical and Computer Engineering
Duke University

Date: __________________

Approved:

Lawrence Carin, Supervisor

Guillermo Sapiro

Robert Calderbank

Katherine Heller

Kafui Dzirasa

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering in the Graduate School of Duke University 2016
Abstract

Multi-output Gaussian processes provide a convenient framework for multi-task problems. An illustrative and motivating example of a multi-task problem is multi-region electrophysiological time-series data, where experimentalists are interested in both power and phase coherence between channels. Recently, the spectral mixture (SM) kernel was proposed to model the spectral density of a single task in a Gaussian process framework. This work develops a novel covariance kernel for multiple outputs, called the cross-spectral mixture (CSM) kernel. This new, flexible kernel represents both the power and phase relationship between multiple observation channels. The expressive capabilities of the CSM kernel are demonstrated through implementation of 1) a Bayesian hidden Markov model, where the emission distribution is a multi-output Gaussian process with a CSM covariance kernel, and 2) a Gaussian process factor analysis model, where factor scores represent the utilization of cross-spectral neural circuits. Results are presented for measured multi-region electrophysiological data.
To my loving and supportive wife, Devin.
Contents

Abstract iv
List of Tables ix
List of Figures x
Acknowledgements xi
1 Introduction 1

2 Gaussian Process Regression 5
  2.1 Predictive Distribution .................................................. 7
  2.2 Covariance Kernels ....................................................... 8
    2.2.1 The Squared Exponential Kernel ................................. 10
    2.2.2 The Spectral Gaussian Kernel .................................... 11
    2.2.3 The Spectral Mixture Kernel ..................................... 12
  2.3 Multi-Output Gaussian Process Regression ............................ 12
  2.4 Learning GP Hyperparameters ........................................ 14
    2.4.1 Notes on Gradient Complexity and Optimization ............. 15

3 Spectral Mixture Kernels for Multi-Output GPs 18
  3.1 The Cross-Spectral Mixture Kernel .................................. 19
  3.2 Discrete Fourier Transform Approximation .......................... 21
    3.2.1 Circulant Matrix Approximation ................................. 22
    3.2.2 Marginal Likelihood of DFT Coefficients ...................... 24
3.2.3 Stability Analysis of the DFT Approximation ... 25
3.3 Predictive Distribution ... 26
3.4 Model Comparison ... 28

4 GP Time-Frequency Applications ... 30
  4.1 Brain State Model ... 32
    4.1.1 Model ... 33
    4.1.2 Inference ... 37
    4.1.3 Results ... 41
  4.2 Including the CSM Kernel in a Bayesian Hierarchical Model ... 44
    4.2.1 Model ... 45
    4.2.2 Inference ... 47
    4.2.3 Results ... 48

5 Cross-Spectral Factor Analysis ... 51
  5.1 Model ... 51
  5.2 Inference ... 53
    5.2.1 Max-Margin Discrimination of Factor Scores ... 56
    5.2.2 Model Updates ... 57
    5.2.3 Discrimination Ensembles ... 58
  5.3 Chronic Stress Data ... 59

6 Conclusions and Future Work ... 65

A Kernel Derivations ... 67
  A.1 The Spectral Gaussian (SG) Kernel ... 67
  A.2 The Spectral Mixture (SM) Kernel ... 68
  A.3 The Cross-Spectral Mixture (CSM) Kernel ... 69

B Infinite Tensor Mixtures ... 71
List of Tables

3.1 Comparison of AIC values .............................................. 29
4.1 Average brain state held-out log predictive probability .............. 44
# List of Figures

3.1 Comparison prior draws from CSM and SM-LMC kernels ............ 20  
3.2 Stability analysis of DFT approximation of CSM kernel ............ 25  
3.3 CSM predictive distribution ........................................ 27  
4.1 Example of LFP time-series ........................................ 33  
4.2 Graphical representation of the state space model ............... 34  
4.3 Toy data results .................................................. 41  
4.4 Sleep data results .................................................. 43  
4.5 Novel environment data results .................................... 44  
4.6 Example LFP data and cross-amplitude/phase spectra ............ 45  
4.7 A subset of results from the Bayesian HMM analysis of brain states . 49  
5.1 Graphical model of supervised max-margin learning ............ 56  
5.2 Classification accuracy visualizations ............................. 60  
5.3 Correlations with interaction ratio ................................. 62  
5.4 Factor usages in prediction tasks ................................. 63  
5.5 Cross-spectra of selected factors ................................. 64
Acknowledgements

I would like to thank everyone who has impacted my education and guided me along the way. To all my fellow lab group members and graduate school peers, thanks for all of the thoughtful discussions. I would like to especially acknowledge David Carlson, who provided me with mentorship and assistance from my entry into the graduate program. I am grateful for my committee members, to include Kafui Dzirasa, Katherine Heller, Guillermo Sapiro, and Robert Calderbank, each of whom has supported and guided me through projects over the years. I cannot thank my advisor, Lawrence Carin, enough for providing me with many opportunities to work on flexible research avenues of particular interest to me.

Some of my largest gratitudes go out to my family. To my parents, for your unbelievable support. To my sister, Riane, for motivating me to go after what I want in life. And most of all, to my wife, Devin, for being there and believing in me throughout my studies.

I am also thankful for the institutions and universities who have graciously hosted me, to include The John’s Hopkins University Applied Physics Laboratory and University College London. Finally, much appreciation is due to the organizations who provided funding, including ARO, DARPA, DOE, DOT, NGA and ONR.
Gaussian process (GP) models have become an important component of the machine learning literature. They have provided a basis for non-linear multivariate regression and classification tasks, and have enjoyed much success in a wide variety of applications (Rasmussen and Williams, 2006).

A GP places a prior distribution over latent functions, rather than model parameters. In the sense that these functions are defined for any number of sample points and sample positions, as well as any general functional form, GPs are nonparametric. The properties of the latent functions are defined by a positive definite covariance kernel that controls the covariance between the function at any two sample points. For a given form of this covariance kernel, along with additive Gaussian observation noise, the posterior distribution of the function given data is analytic.

Recently, the spectral mixture (SM) kernel has been shown to encode the power spectral density of the latent function (Wilson and Adams, 2013). This flexible class of kernels is capable of recovering any composition of standard stationary kernels (Gönen and Alpaydın, 2011; Lloyd et al., 2014), and the kernel is easily interpreted through its spectral density, represented as a mixture of Gaussian compo-
nents. The SM kernel has been used for GP regression of a scalar output (i.e., single function, or observation “task”), achieving impressive results in extrapolating atmospheric CO\(_2\) concentrations (Wilson and Adams, 2013); image inpainting (Wilson et al., 2014); and feature extraction from electrophysiological signals (Ulrich et al., 2014).

However, the SM kernel is not defined for multiple outputs (multiple correlated functions). Multi-output GPs intersect with the field of multi-task learning (Caruana, 1997), where solving similar problems jointly allows for the transfer of statistical strength between problems, improving learning performance when compared to learning all tasks individually. This work considers neuroscience applications where low-frequency (< 200 Hz) extracellular potentials are simultaneously recorded from implanted electrodes in multiple brain regions of a mouse (Dzirasa et al., 2011). These signals are known as local field potentials (LFPs) and are often highly correlated between channels. A potential modeling decision would be to draw the data from each channel as an independent GP (Ulrich et al., 2014). This choice, presented in Section 4.1, is suboptimal because the LFP channels are highly interdependent, and inferring and understanding that interdependence is biologically significant. Therefore, jointly modeling multiple channels as a draw from a multi-output GP is desirable.

A multi-output GP can be thought of as a standard GP (all observations are jointly normal) where the covariance kernel is a function of both the input space and the output space (see Alvarez et al. (2011) and references therein for a comprehensive review); here “input space” means the points at which the functions are sampled (e.g., time), and the “output space” may correspond to different brain regions. A particular positive definite form of this multi-output covariance kernel is the sum of separable (SoS) kernels, or the linear model of coregionalization (LMC) in the geostatistics literature (Goovaerts, 1997), where a separable kernel is represented by the product of separate kernels for the input and output spaces.
This work extends the SM kernel to the multi-output setting via the LMC framework, which we term the SM-LMC kernel (Ulrich et al., 2015). The SM-LMC kernel represents each channel in the output space as a weighted sum of underlying latent functions drawn from GPs with SM covariance kernels. While this is a powerful framework for modeling the observations, the SM-LMC kernel does not provide an intuitive representation for the data it models. Specifically, the SM-LMC kernel encodes the cross-amplitude spectrum (square root of the cross power spectral density) between every pair of channels, but provides no information about the cross-phase spectrum. Together, the cross-amplitude and cross-phase spectra form the cross-spectrum, which is obtained through the Fourier transform of the cross-covariance between the pair of channels.

Motivated by the desire to encode the full cross-spectra into the covariance kernel, a novel kernel termed the cross-spectral mixture (CSM) kernel is designed, which provides an intuitive representation of the power and phase dependencies between multiple outputs. The need for embedding the full cross-spectrum into the covariance kernel is illustrated by a recent surge in neuroscience research discovering that LFP interdependencies between regions exhibit phase synchrony patterns that are dependent on frequency band (Gregoriou et al., 2009; Sauseng and Klimesch, 2008; Sweeney-Reed et al., 2014).

This document is organized as follows. Chapter 2 introduces Gaussian processes in the regression framework. The function-space view of Gaussian processes is considered, and GPs are formalized in a multi-output setting. In Chapter 3, the cross-spectral mixture kernel is introduced. Comparisons of the CSM kernel to other multi-output GP kernel frameworks are presented, motivating the power and flexibility of the CSM kernel as a cross-spectral feature extractor. Chapter 4 proceeds to discuss hierarchical models that include the SM and CSM kernels as a generative process to perform (non-classical) time-frequency analysis. Chapter 5 then introduces a latent
GP factor model that incorporates the CSM kernel as the spectral content associated with each factor. Conclusions and future extensions are presented in Chapter 6.
A traditional regression task takes an input covariate vector $\mathbf{x}_n \in \mathbb{R}^p$ and estimates an observation $y(\mathbf{x}_n) \in \mathbb{R}$ for that set of covariates. Often, an unobserved latent process $f(\mathbf{x})$ is responsible for generating the observation, i.e.,

$$y(\mathbf{x}_n) = f(\mathbf{x}_n) + \epsilon_n, \quad \epsilon_n \sim \mathcal{N}(0, \eta^{-1}),$$

such that each datum is represented as the estimate $f(\mathbf{x})$ plus independent Gaussian noise with precision $\eta$.

This formulation encompasses a broad class of models, including, for example, linear regression, where $f(\mathbf{x}) = \mathbf{a}^T \mathbf{x}$ is the inner product function; here, the function evaluations are linear transformations of their respective covariates. In such parametric models, prior knowledge is often imposed on the unknown parameters $\mathbf{a}$ via a prior distribution $p(\mathbf{a})$. To form a conjugate linear model, this example continues with $p(\mathbf{a}) = \mathcal{N}(\mathbf{0}, \Sigma)$. Inference proceeds by computing the posterior distribution $p(\mathbf{a}|\mathbf{X}, \mathbf{y})$ where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ is a concatenated matrix of covariates for respective observations $\mathbf{y} = [y_1, \ldots, y_N]$.

Observing the mean and covariance functions of this latent process reveals the
definitions $E[f(x)] = 0$ and $\text{cov}(f(x), f(x')) = x^T\Sigma x'$, respectively. By marginalizing out the parameters of the model $(\alpha)$, the mean and covariance functions may be viewed as a distribution over the latent function $f(x)$ with hyperparameters $\Sigma$. Given the hyperparameters and a set of observations, the posterior distribution of the latent function is analytic for the entire input domain. This differs from the original parametric model for $f(x)$, where the posterior distribution of the parameters is desired.

While such parametric models are often utilized for their computational convenience, the marginal likelihood (i.e., all model parameters integrated out of the model likelihood) must be obtained in order to form a predictive distribution for function evaluations at arbitrary locations in the input domain. This marginalization is often not analytic for models more complicated than linear models. Furthermore, directly providing a distribution over the latent function by defining a more general form of the mean/covariance functions may be more intuitive than formulating a complicated non-conjugate hierarchical model. For instance, one may wish to impose a smooth, stationary, or even periodic function. Such properties may be achieved through a careful design of the covariance function within a framework known as a Gaussian process.

**Definition 2.0.1.** Gaussian processes (GPs) place a prior distribution over the latent function (Rasmussen and Williams, 2006) according to

$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

where the mean function $m(x) = E[f(x)]$ is typically set to zero, and the covariance function (also termed the covariance kernel) $k(x, x') = \text{cov}(f(x), f(x'))$ creates dependencies between observations. The GP distribution is nonparametric in the sense that it is defined for an infinite number of functional evaluations, and general functional forms.
This chapter proceeds with Section 2.1 displaying the power of the GP predictive distribution. Section 2.2 introduces popular covariance kernels for scalar output Gaussian processes that are relevant to this work. An extension of Gaussian processes to multi-output observations is then presented in Section 2.3, as the remainder of this work addresses novel covariance kernels for multi-output GPs. Finally, section 2.4 details selecting the hyperparameters of the Gaussian process to maximize the marginal likelihood of the observed data.

2.1 Predictive Distribution

In practice, a dataset of observations \( y = [y(x_1), \ldots, y(x_N)]^T \in \mathbb{R}^N \) is provided, where each observation corresponds to a respective location \( X = [x_1, \ldots, x_N]^T \). Given a Gaussian process model, it is often of interest to obtain the distribution of the latent function at a new set of \( N^* \) input locations \( X^* = [x_1^*, \ldots, x_N^*]^T \). For instance, one may wish to interpolate a signal at non-observed input locations, or even extrapolate a signal to provide future forecasts for time-series data. In this GP framework, observations need not exist on an evenly spaced grid, allowing for richer forecasts than one-step-ahead predictions. These one-step-ahead forecasts are prevalent in Linear Dynamical Systems (Prado and West, 2010), where models are often restricted to observations on evenly spaced grids. This is one reason GPs provide a powerful framework for signal interpolation/extrapolation.

As stated in the introduction, the observations are represented by an underlying latent function plus additive Gaussian noise according to \( y = f(x) + \epsilon \), where \( \epsilon \) has precision \( \eta \). The covariance of the set of observations is given by

\[
\text{cov}(y) = K(X, X) + \eta^{-1} I_N, \tag{2.3}
\]

where the Gram matrix \( K(X, X) \) has deterministic entries \( (K(X, X))_{ij} = k(x_i, x_j) \). When \( k(x_i, x_j) \) is a valid positive semidefinite covariance function, the Gram matrix
is known as the covariance function.

The joint distribution between the observations $y$ and function values $f^*$ corresponding to the input locations of interest $X^*$ is the multivariate Gaussian

$$
\begin{bmatrix}
y \\
f^*
\end{bmatrix} \sim \mathcal{N}
\left(0,
\begin{bmatrix}
K(X, X) + \eta^{-1} I_N & K(X, X^*) \\
K(X^*, X) & K(X^*, X^*)
\end{bmatrix}
\right),
$$

(2.4)

where $K(X^*, X^*)$ is the $N^* \times N^*$ covariance matrix for the new input locations, and $K(X, X^*)$ is the $N \times N^*$ matrix of covariance function evaluations between the observation and test sets of locations, i.e., $(K(X, X^*))_{ij} = k(x_i, x_j^*)$. Using conditional multivariate Gaussian distribution theory, GP regression establishes a functional form for the distribution at test locations

$$
p(f^*|y, X, X^*) = \mathcal{N}({\bar{\mu}}^*, {\bar{K}}^*),
$$

(2.5)

where

$$
{\bar{\mu}}^* = K(X^*, X) \left[K(X, X) + \eta^{-1} I_N\right]^{-1} y
$$

(2.6)

$$
{\bar{K}}^* = K(X^*, X^*) - K(X^*, X) \left[K(X, X) + \eta^{-1} I_N\right]^{-1} K(X, X^*)
$$

(2.7)

The remainder of this work sometimes uses the notation $K \triangleq K(X, X)$ as shorthand for the covariance matrix.

### 2.2 Covariance Kernels

Covariance kernels are essential to Gaussian processes because they encode knowledge about the underlying function generating the observed data. A covariance kernel enforces similarity between data points. For instance, with a smooth function, a covariance kernel is desired such that nearby input locations map to similar function evaluations.

Similarities in the evaluations of a given function $f : \mathcal{X} \rightarrow \mathbb{R}$ between input locations $x \in \mathcal{X}$ and $x' \in \mathcal{X}$ is represented by the covariance $\text{cov}(f(x), f(x')) \in \mathbb{R}$. A
Gaussian process represents this cross-covariance by a *kernel*, defined as a function $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that $\text{cov}(f(x), f(x')) = k(x, x')$. In order to represent a true covariance function, the covariance kernel must be positive semidefinite. To satisfy this constraint, every possible Gram matrix $K(X, X)$ produced by the covariance kernel must result in a positive semidefinite matrix, which is a matrix characterized by all nonnegative eigenvalues.

There are many valid choices for covariance kernels that satisfy this definition. This work, however, is interested in examining a small subset of possible kernels. Therefore, several definitions are introduced here that restrict the structure of the covariance functions considered:

1. *Univariate input-space*: This work is mostly concerned with time-series analysis where $x \in \mathbb{R}$ corresponds to a univariate temporal dimension. Although every kernel presented in the subsequent text may be extended to a multivariate input space, this notation is suppressed for clarity.

2. *Stationarity*: A stationary covariance function is a function of $x - x'$, and therefore does not depend on input space location, making it invariant to translations in input space.

3. *Symmetry*: A symmetric kernel requires $k(x, x') = k(x', x)$.

4. *Isotropy*: Another term to indicate a symmetric, stationary kernel, an isotropic covariance function depends only on $\tau = |x - x'|$, implying radial symmetry.

5. *Infinite Divisibility*: This class of kernels produces smooth, infinitely divisible functions, containing kernels where $(k(\tau))^r$ is a valid covariance kernel $\forall r > 0$.

This section first introduces the popular squared exponential (SE) covariance kernel in Section 2.2.1. The spectral Gaussian (SG) kernel presented in Section 2.2.2
represents an SE kernel shifted in the frequency domain. In section 2.2.3, these SG kernels are combined to produce a class of covariance kernels that encode the power spectral density of the observation sequence, known as the spectral mixture (SM) kernel (Wilson and Adams, 2013).

2.2.1 The Squared Exponential Kernel

The isotropic squared exponential (SE) kernel is defined by

\[
k_{SE}(\tau; \theta) = \exp \left( -\frac{\tau^2}{2\ell^2} \right)
\]  

(2.8)

where the only kernel hyperparameter \( \ell \in \theta \) is known as the characteristic length-scale. A smaller length-scale implies more local input-space dependencies, resulting in latent functions with more local variations. The SE kernel is infinitely divisible, thereby producing infinitely divisible (i.e., smooth) functions. This property has made the SE kernel the default kernel used in many machine learning applications, even when the SE kernel is not necessarily the best possible choice.

Many generalizations of the SE kernel exist, though they are not introduced in detail here. For example, the rational quadratic (RQ) kernel results from integrating out a gamma prior on \( \ell^{-2} \). This produces a heavy-tailed kernel as a scale-mixture of SE kernels. The Matérn kernel is a finitely differentiable kernel that converges to the SE kernel as a degree-of-freedom term approaches infinity. This kernel is appropriate when the desired function is not smooth.

The Fourier transform of the SE kernel provides the angular spectral density

\[
S_{SE}(\omega; \theta) = \ell^{-\frac{1}{2}} \exp\left( -\frac{1}{2\ell^2} \omega^2 \right),
\]  

(2.9)

which is clearly a scaled Gaussian distribution centered at a mean frequency of 0. Because of this, the SE kernel is a powerful low-frequency smoothing function, but
has no ability to represent any interesting spectral densities. The spectral Gaussian kernel simply shifts this spectral density to have a non-zero mean.

2.2.2 The Spectral Gaussian Kernel

A spectral Gaussian (SG) kernel is defined by a spectral density function with a single Gaussian distribution reflected about the origin,

$$S_{SG}(\omega; \theta) = \frac{1}{2} [\mathcal{N}(\omega; -\mu, \nu) + \mathcal{N}(\omega; \mu, \nu)], \quad (2.10)$$

where $\theta = \{\mu, \nu\}$ are the kernel parameters, $\mu$ represents the peak frequency, and the variance $\nu$ is a scale parameter that controls the spread of the spectral density around $\mu$. This density is a function of angular frequency. The Fourier transform of (2.10) results in the stationary, positive definite auto-covariance function

$$k_{SG}(\tau; \theta) = \exp\left(-\frac{1}{2} \nu \tau^2\right) \cos(\mu \tau), \quad (2.11)$$

where stationarity implies dependence on input domain differences $k(\tau; \theta) = k(x, x'; \theta)$ with $\tau = x - x'$. The SG kernel may also be derived by considering a latent signal $f(x) = \sqrt{2} \cos(\omega(x + \phi))$ with frequency uncertainty $\omega \sim \mathcal{N}(\mu, \nu)$ and phase offset $\omega \phi$. The kernel is the auto-covariance function for $f(x)$, such that $k_{SG}(\tau; \theta) = \text{cov}(f(x), f(x+\tau))$. When computing the auto-covariance, the frequency $\omega$ is marginalized out, providing the kernel in (2.11) that includes all frequencies in the spectral domain with probability 1.
2.2.3 The Spectral Mixture Kernel

A weighted, linear combination of SG kernels gives the spectral mixture (SM) kernel Wilson and Adams (2013),

\[ k_{SM}(\tau; \theta) = \sum_{q=1}^{Q} a_q k_{SG}(\tau; \theta_q), \quad (2.12) \]

\[ S_{SM}(\omega; \theta) = \sum_{q=1}^{Q} a_q S_{SG}(\omega; \theta_q), \quad (2.13) \]

where \( \theta_q = \{a_q, \nu_q, \mu_q\} \) and \( \theta = \{\theta_q\} \) has 3\( Q \) degrees of freedom. The SM kernel may be derived as either the Fourier transform of the spectral density \( S_{SM}(\omega; \theta) \) or as the auto-covariance of latent functions \( f(x) = \sum_{q=1}^{Q} \sqrt{2a_q} \cos(\omega_q(x + \phi_q)) \) with uncertainty in angular frequency \( \omega_q \sim \mathcal{N}(\mu_q, \nu_q) \).

The moniker for the SM kernel in (2.13) reflects the mixture of Gaussian components that define the spectral density of the kernel. Since these components are allowed non-zero means, the SM kernel is able to represent any stationary covariance kernel given large enough \( Q \). To name a few, this includes any combination of squared exponential, Matèrn, rational quadratic, or periodic kernels (Gönen and Alpaydn, 2011; Rasmussen and Williams, 2006; Wilson and Adams, 2013).

2.3 Multi-Output Gaussian Process Regression

A multi-output regression task estimates samples from \( C \) output channels, \( y_n = [y_{n1}, \ldots, y_{nC}]^T \) corresponding to the \( n \)-th input point \( x_n \) (e.g., the \( n \)-th temporal sample). An unobserved latent function \( f(x) = [f_1(x), \ldots, f_C(x)]^T \) is responsible for generating the observations, such that

\[ y_n \sim \mathcal{N}(f(x_n), H^{-1}), \quad (2.14) \]

where \( H = \text{diag}(\eta_1, \ldots, \eta_C) \) and \( \eta_c \) is the precision of Gaussian noise for channel \( c \).
A GP prior on the latent function, for arbitrary input $x$, is formalized by

$$f(x) \sim \mathcal{GP}(m(x), K(x, x')),$$

where the mean function vector $m(x) \in \mathbb{R}^C$ is often set to equal 0, and the covariance function $(K(x, x'))_{c,c'} = k^{c,c'}(x, x') = \text{cov}(f_c(x), f_{c'}(x'))$ creates dependencies between observations at input points $x$ and $x'$, as observed on channels $c$ and $c'$. In general the input space $x$ could be vector valued, but for simplicity we here assume it to be scalar, consistent with our motivating neuroscience application in which $x$ corresponds to time.

A convenient representation for multi-output kernel functions is to separate the kernel into the product of a kernel for the input space and a kernel for the interactions between the outputs. This is known as a separable kernel. A \textit{sum of separable kernels} (SoS) representation Alvarez et al. (2011) is given by

$$k^{c,c'}(x, x') = \sum_{q=1}^{Q} b_q(c, c') k_q(x, x'), \quad \text{or} \quad K(x, x') = \sum_{q=1}^{Q} B_q k_q(x, x'),$$

where $k_q(x, x')$ is the input space kernel for component $q$, $b_q(c, c')$ is the $q$-th output interaction kernel, and $B_q \in \mathbb{R}^{C \times C}$ is a positive semi-definite output kernel matrix. Note that we have a discrete set of $C$ output spaces, $c \in \{1, \ldots, C\}$, where the input space $x$ is continuous, and discretely sampled arbitrarily in experiments. The SoS formulation is also known as the \textit{linear model of coregionalization} (LMC) Goovaerts (1997) and $B_q$ is termed the coregionalization matrix. When $Q = 1$, the LMC reduces to the \textit{intrinsic coregionalization model} (ICM) Alvarez et al. (2011), and when rank($B_q$) is restricted to equal 1, the LMC reduces to the \textit{semiparametric latent factor model} (SLFM) Teh et al. (2005).

Any finite number of latent functional evaluations $f = [f_1(x), \ldots, f_C(x)]^T$ at locations $x = [x_1, \ldots, x_N]^T$ has a multivariate normal distribution $\mathcal{N}(f; 0, K)$, such
that \( K \) is formed through the block partitioning

\[
K = \begin{bmatrix}
k^{1,1}(x, x) & \cdots & k^{1,C}(x, x) \\
\vdots & \ddots & \vdots \\
k^{C,1}(x, x) & \cdots & k^{C,C}(x, x)
\end{bmatrix} = \sum_{q=1}^{Q} B_q \otimes k_q(x, x), \tag{2.17}
\]

where each \( k^{c,d}(x, x) \) is an \( N \times N \) matrix and \( \otimes \) symbolizes the Kronecker product.

### 2.4 Learning GP Hyperparameters

A dataset of vector-valued data consists of observations \( y = [y^1, \ldots, y^C]^T \in \mathbb{R}^{CN} \), where observations from channel \( c \) are \( y^c = [y_{1c}, \ldots, y_{Nc}]^T \in \mathbb{R}^N \) at the respective locations \( x = [x_1, \ldots, x_N]^T \). Since both the likelihood \( p(y|f, x) \) and distribution over latent functions \( p(f|x) \) are Gaussian, the marginal likelihood is conveniently represented by

\[
p(y|x) = \int p(y|f, x)p(f|x)df = \mathcal{N}(0, \Gamma), \tag{2.18}
\]

\[
\Gamma = K + H^{-1} \otimes I_N, \tag{2.19}
\]

where all possible functions \( f \) have been marginalized out.

All probabilities in the previous discussion that involve the covariance kernel should be conditioned on a set of parameters \( \theta \). This conditioning was removed for notational simplicity, but will henceforth be included in the notation. For example, if the squared exponential kernel is used, then \( k_{\text{SE}}(x, x'; \theta) = \sigma^2 \exp(-\frac{1}{2}||x - x'||^2/\ell^2) \), where the parameters defining the latent functions are \( \theta = \{\sigma^2, \ell\} \). In this work, the observation precision terms \( \eta = [\eta_1, \ldots, \eta_C]^T \) that compromise \( H \) are optimized in the same manner as the kernel parameters. Therefore, a model is chosen by finding the optimal \( \Theta = \{\theta, \eta\} \).
To fit a GP to the dataset, the parameters $\Theta$ are typically chosen to maximize the marginal likelihood in (2.18) via gradient ascent. This is equivalent to maximizing the log marginal likelihood,

$$\log p(y|x) = -\frac{CN}{2} \log(2\pi) - \frac{1}{2} |\Gamma| - \frac{1}{2} x^T \Gamma^{-1} x.$$  

The gradient of the log marginal likelihood with respect to each parameter $\Theta_j$ is

$$\frac{\partial}{\partial \Theta_j} \log p(y|x) = \frac{1}{2} \text{tr} \left( (\alpha \alpha^T - \Gamma^{-1}) \frac{\partial \Gamma}{\partial \Theta_j} \right),$$  

(2.20)

where $\alpha = \Gamma^{-1} y$ and $\text{tr}(\cdot)$ is the trace operator. It is important to note that this gradient requires finding the inverse of $\Gamma$, which has complexity $O(N^3 C^3)$.

### Notes on Gradient Complexity and Optimization

The na"ive implementation of the gradients in (2.20) scales poorly with the number of observations. Several methods have been introduced in the literature to speed up Gaussian processes. In particular, sparse approximate Gaussian processes (Quiñonero-Candela and Rasmussen, 2005) introduce inducing points to form an effective prior.
that assumes conditional independence among all observations given the inducing points. Another popular approach is to reformulate multi-output Gaussian processes as multi-input Gaussian processes. For example, if all observations are located on a spatio-temporal grid, then separable covariance kernels may be used along each input dimensions, forming a covariance kernel with Kronecker structure (Saatçi, 2011; Gilboa et al., 2015; Flaxman et al., 2015). Due to the low spatial resolution of the data considered in this report, we prefer to consider low-rank free form coregionalization matrices over this multi-input Kronecker structure. Section 3.2 focuses on a transformation of the covariance kernel using the discrete Fourier transform (DFT) to improve storage and computation complexity as an alternative to the methods listed above.

Several methods for training Gaussian process hyperparameters are detailed in the literature (Rasmussen, 2004), such as conjugate gradient descent and resilient backpropagation (Blum and Riedmiller, 2013). Rprop is a gradient-based optimization technique that uses adaptive update steps that depend only on the sign of the gradient. The update rule is as follows:

\[ \Theta_j^{(t+1)} = \Theta_j^{(t)} - \text{sign} \left( -\frac{\partial}{\partial \Theta_j} \log p(y|x) \right) \delta_j^{(t)} . \]  

(2.21)

Within the last two consecutive iterations, if the sign of the partial derivative of the negative log likelihood remains the same, then \( \delta_j \) is increased by a factor \( \eta^+ > 1 \) to accelerate the parameter values to a local maxima. If the sign of the derivative does not remain the same between consecutive iterations, then \( \delta_j \) is decreased by a factor \( 0 < \eta^- < 1 \) to allow the parameter to converge to a local optima. The values of \( \eta^+ = 1.2 \) and \( \eta^- = 0.5 \) are empirically found to perform best.

Igel and Hüskens (2003) discuss several of the benefits of Rprop, to include its speed and accuracy, robustness to internal parameters, suitability to applications with noisy gradients (e.g., numerical estimation), and ease of use. In many Gaussian
process settings, Rprop performs just as well or better than conjugate gradient descent (Blum and Riedmiller, 2013). It is, however, a batch algorithm, so stochastic inference is theoretically not guaranteed to converge. Algorithm details an implementation of Rprop.
The SM kernel introduced by Wilson and Adams (2013) is considered a flexible feature extractor. The spectral components of the SM kernel are capable of representing a wide variety of patterns, and the GP framework allows for a convenient extrapolation of these patterns to unobserved locations in the input space.

While multi-output versions of the SM kernel may be developed by directly applying the LMC framework in Equation 2.16 to represent complex spectral relationships between channels, this straightforward SM-LMC kernel does not intuitively model the cross-phase spectrum between channels. We propose a novel kernel known as the *cross-spectral mixture* (CSM) kernel that provides both the cross-amplitude and cross-phase spectra of multi-channel observations. Detailed derivations of each of these kernels are found in Appendix A.

This chapter first introduces the CSM kernel in Section 3.1. A fast inference method is then presented in Section 3.2, which helps to alleviate the naive cubic complexity of Gaussian process regression. Finally, the power of the predictive distribution for the CSM kernel is shown in Section 3.3.
3.1 The Cross-Spectral Mixture Kernel

A multi-output version of the SM kernel uses the SG kernel directly within the LMC framework:

\[ K_{SM-LMC}(\tau; \theta) = \sum_{q=1}^{Q} B_q k_{SG}(\tau; \theta_q), \]  

(3.1)

where \( Q \) SG kernels are shared among the outputs via the coregionalization matrices \( \{B_q\}_{q=1}^{Q} \). A generalized, non-stationary version of this SM-LMC kernel was proposed in (Wilson et al., 2014) using the Gaussian process regression network (GPRN) (Wilson and Knowles, 2012). The marginal distribution for any single channel is simply a Gaussian process with a SM covariance kernel. While this formulation is capable of providing a full cross-amplitude spectrum between two channels, it contains no information about a cross-phase spectrum. Specifically, each channel is merely a weighted sum of \( \sum_q R_q \) latent functions where \( R_q = \text{rank}(B_q) \). Whereas these functions are shared exactly across channels, our novel CSM kernel shares phase-shifted versions of these latent functions across channels.

**Definition 3.1.1.** The cross-spectral mixture (CSM) kernel takes the form

\[ k_{CSM}^{c,c'}(\tau; \theta) = \sum_{q=1}^{Q} \sum_{r=1}^{R_q} \sqrt{a_{cq}^r a_{c'q}^r} \exp\left(-\frac{1}{2} \nu_q \tau^2\right) \cos \left( \mu_q (\tau + \phi_{cq}^r - \phi_{c'q}^r) \right), \]  

(3.2)

where \( \theta = \{\nu_q, \mu_q, \{a_{cq}^r, \phi_{cq}^r, \phi_{c'q}^r \neq 0\}_{r=1}^{R_q}\}_{q=1}^{Q} \) has \( 2Q + \sum_{q=1}^{Q} R_q (2C - 1) \) degrees of freedom, and \( a_{cq}^r \) and \( \phi_{cq}^r \) respectively represent the amplitude and shift in the input space for latent functions associated with channel \( c \). In the LMC framework, the CSM kernel is

\[ K_{CSM}(\tau; \theta) = \text{Re} \left\{ \sum_{q=1}^{Q} B_q \tilde{k}_{SG}(\tau; \theta_q) \right\}, \quad B_q = \sum_{r=1}^{R_q} \beta_q^r (\beta_q^r)^\dagger, \]  

(3.3)

\[ \tilde{k}_{SG}(\tau; \theta_q) = \exp\left(-\frac{1}{2} \nu_q \tau^2 + j \mu_q \tau\right), \quad \beta_{cq}^r = \sqrt{a_{cq}^r} \exp(-j \psi_{cq}^r), \]  

(3.4)
where \( \tilde{k}_{SG}(\tau, \theta_q) \) is phasor notation of the SG kernel, \( B_q \) is rank-\( R_q \), \( \{ \beta_{cq}^r \} \) are complex scalar coefficients encoding amplitude and phase, and \( \psi_{cq}^r \equiv \mu_q \phi_{cq}^r \) is an alternative phase representation. We use complex notation where \( j = \sqrt{-1} \), \( \text{Re}\{\cdot\} \) returns the real component of its argument, and \( \beta^\dagger \) represents the complex conjugate of \( \beta \).

Both the CSM and SM-LMC kernels force the marginal distribution of data from a single channel to be a Gaussian process with a SM covariance kernel. The CSM kernel is derived in the Supplemental Material by considering functions represented by phase-shifted sinusoidal signals,

\[
f_c(x) = \sum_{q=1}^{Q} \sum_{r=1}^{R_q} \sqrt{2\sigma_{cq}^r} \cos(\omega_{cq}^r(x + \phi_{cq}^r)),
\]

(3.5)

where each \( \omega_{cq}^r \overset{iid}{\sim} \mathcal{N}(\mu_q, \nu_q) \). Computing the cross-covariance function

\[
\text{cov}(f_c(x), f_c(x + \tau))
\]

(3.6)

provides the CSM kernel.

A comparison between draws from Gaussian processes with CSM and SM-LMC kernels is shown in Figure 3.1. The utility of the CSM kernel is clearly illustrated by the cross-amplitude and cross-phase spectra between \( f_1(x) \) and \( f_2(x) \) shown for the CSM kernel (solid) and SM-LMC kernel (dashed). The ability to tune phase relationships is beneficial for kernel design and interpretation.

**Figure 3.1**: Latent functions drawn for two channels \( f_1(x) \) (blue) and \( f_2(x) \) (red) using the CSM kernel (left) and rank-1 SM-LMC kernel (center). The functions are comprised of two SG components centered at 4 and 5 Hz. For the CSM kernel, we set the phase shift \( \psi_{c,2} = \pi \). Right: the cross-amplitude (purple) and cross-phase (green) spectra between \( f_1(x) \) and \( f_2(x) \) are shown for the CSM kernel (solid) and SM-LMC kernel (dashed). The ability to tune phase relationships is beneficial for kernel design and interpretation.
by its ability to encode phase information, as well as its powerful functional form of the full cross-spectrum (both amplitude and phase). The amplitude function $A_{c,c'}(\omega)$ and phase function $\Phi_{c,c'}(\omega)$ are obtained by representing the cross-spectrum in phasor notation, i.e.,

$$
\Gamma_{c,c'}(\omega; \Theta) = \sum_q (B_q)_{c,c'} S_{SG}(\omega; \theta_q)
= A_{c,c'}(\omega) \exp(j\Phi_{c,c'}(\omega)).
$$

(3.7)

Interestingly, while the CSM and SM-LMC kernels have identical marginal amplitude spectra for shared $\{\mu_q, \nu_q, a_q\}$, their cross-amplitude spectra differ due to the inherent destructive interference of the CSM kernel (see Figure 3.1, right).

### 3.2 Discrete Fourier Transform Approximation

A common trick for GPs with evenly spaced samples (e.g., a temporal grid) is to use the discrete Fourier transform (DFT) to approximate the inverse of the marginal likelihood covariance matrix ($\Gamma$ from Equation 2.19) by viewing this as an approximately circulant matrix (Dietrich and Newsam, 1997; Lázaro-Gredilla et al., 2010). These methods can speed up inference because circulant matrices are diagonalizable by the DFT coefficient matrix. Adjusting these methods to the multi-output formulation, we show how the DFT of the marginal covariance matrices retains the cross-spectrum information of the kernel.

**Proposition 3.2.1.** Let $y \sim \mathcal{N}(0, \Gamma)$ represent the marginal likelihood of circularly-symmetric (Gallager, 2008) real-valued observations, and denote the concatenation of the DFT of each channel as $z = (I_C \otimes U)^\dagger y$ where $U$ is the $N \times N$ unitary DFT
matrix. Then, $z$ has the complex normal distribution Gallager (2008):

$$z \sim \mathcal{CN}(0, 2S),$$

$$S = \delta^{-1} \sum_{q=1}^{Q} B_q \otimes W_q + H^{-1} \otimes I_N,$$

where $\delta = x_{i+1} - x_i$ for all $i = 2, \ldots, N$, and $W_q \approx \text{diag}(S_{SG}(\omega_q; \theta_q), 0)$ is approximately diagonal. The spectral density $S_{SG}(\omega; \theta) = [S_{SG}(\omega_1; \theta), \ldots, S_{SG}(\omega_{\lfloor N/2 \rfloor}; \theta)]$ is found via (2.10) at angular frequencies $\omega = \frac{2\pi}{N\delta} [0, 1, \ldots, \lfloor \frac{N}{2} \rfloor]$, and $0 = [0, \ldots, 0]$ is a row vector of $\lfloor \frac{N-1}{2} \rfloor$ zeros.

The parameters of the CSM kernels $\Theta$ may now be optimized from the expected marginal log-likelihood of $z$ instead of $y$. Conceptually, the only difference during the fitting process is that, with the latter, derivatives of the covariance kernel are used, while, with the former, derivatives of the power spectral density are used. Computationally, this method improves the naïve $O(N^3C^3)$ complexity of fitting the standard CSM kernel to $O(NC^3)$ complexity. Memory requirements are also reduced from $O(N^2C^2)$ to $O(NC^2)$. The reason for this improvement is that $S$ is now represented as $N$ independent $C \times C$ blocks, reducing the inversion of $S$ to inverting a permuted block-diagonal matrix.

The remainder of this section discusses details of Proposition 3.2.1. Section 3.2.1 provides details on the circulant matrix approximation used on each $K_q$, and Section 3.2.2 discusses the complex normal distribution. Finally, Section 3.2.3 performs a stability analysis on recovering a true CSM kernel with this DFT approximation.

### 3.2.1 Circulant Matrix Approximation

The marginal likelihood variance of the CSM kernel is defined as $\Gamma = \text{Re}\{\hat{\Gamma}\}$ where

$$\hat{\Gamma} = \sum_q B_q \otimes K_q + H^{-1} \otimes I_N$$

(3.10)
and $K_q = \tilde{k}_{SG}(x, x; \theta)$ for a set of input locations $x$.

By definition, the covariance matrix $\Sigma = K_q$, for any $q$, is a symmetric Toeplitz matrix. There are several properties of symmetric Toeplitz matrices that make the form of $\tilde{\Gamma}$ particularly convenient. One of these properties is that a matrix of this form is uniquely identified by the first column of the matrix, denoted here as $\sigma$. Furthermore, a symmetric Toeplitz matrix is closely related to a circulant matrix. Specifically, reflecting the first $\frac{N}{2} + 1$ elements of $c$ to the last $\frac{N}{2} + 1$ elements will produce a circulant approximation, $\tilde{\sigma}$, to $\sigma$. The resulting matrix is also Toeplitz, $\tilde{\Sigma} = \text{toeplitz}(\tilde{\sigma})$, and is closely related to the discrete Fourier transform (DFT) in the following way. Given the eigen-decomposition $\tilde{\Sigma} = V_{\tilde{\Sigma}} \Lambda_{\tilde{\Sigma}} V_{\tilde{\Sigma}}^{-1}$, the vector of eigenvalues $\lambda_{\tilde{\Sigma}} = \text{diag}(\Lambda_{\tilde{\Sigma}})$ equals the complex DFT of $\tilde{\sigma}$. Mathematically,

$$U \tilde{\sigma} = \lambda_{\tilde{\Sigma}},$$  \hspace{1cm} (3.11)

where $U$ is the $N \times N$ unitary DFT matrix (Healy and White, 2005).

This reveals that the columns of $U$ are the eigenvectors of $\tilde{\Sigma}$, thereby implying that (a) $\tilde{\Sigma}$ is diagonalizable by the DFT coefficient matrix, and (b) the elements along the diagonal, $\lambda_{\tilde{\Sigma}}$, refer to the power spectral density of $\tilde{\sigma}$, i.e.,

$$U^\dagger \tilde{\Sigma} U = \Lambda_{\tilde{\Sigma}} = \text{diag} \left( \delta^{-1} S(\omega) \right),$$ \hspace{1cm} (3.12)

where the spectral density values are scaled by $\delta^{-1}$ and evaluated independently $S(\omega) = [S(\omega_1), \ldots, S(\omega_N)]$ at angular frequency locations

$$\omega = \frac{2\pi}{N \delta} \left[ 0, 1, \ldots, \left\lfloor \frac{N}{2} \right\rfloor, -\left\lfloor \frac{N-1}{2} \right\rfloor, \ldots, -1 \right].$$ \hspace{1cm} (3.13)

Because of this, the full matrix $\Sigma$ need not ever be stored; rather, the spectral density vector $S(\omega)$ is sufficient.

As $\delta \to 0$ and $x_N \to \infty$, this spectral density is exact and there is no error in approximating $\Sigma$ with $\tilde{\Sigma}$. This is due to the increasing resolution of the DFT
frequency bins. When the resolution is poor, however, the elements of $\Sigma$ have small, but non-negligible, negative correlations.

### 3.2.2 Marginal Likelihood of DFT Coefficients

The marginal likelihood formulation $y \sim \mathcal{N}(0, \Gamma)$ is particularly useful if the DFT approximation from the previous section is applied. Specifically, the marginal likelihood of the linear transformations $\{U^\dagger y_c\}_{c=1}^C$ is of interest. The following provides details about this distribution.

Although only the real portion of $y = y^r + jy^i$ is observed, the fact that $y^r$ is a sum of latent sinusoidal signals allows the complex vector $y$ to be represented as a circularly symmetric complex normal random variable, such that $y \sim \mathcal{CN}(0, 2\Gamma)$. This formulation implies that the distribution of $y$ is invariant to any real phase shift in the underlying latent signals (Gallager, 2008). To avoid complex numbers, the joint distribution of the real and imaginary components of $y$ is equivalent to the multivariate Gaussian distribution,

$$
\begin{bmatrix}
  y^r \\
  y^i
\end{bmatrix} \sim \mathcal{N}
\left(
\begin{bmatrix}
  0 \\
  0
\end{bmatrix},
\begin{bmatrix}
  \text{Re}\{\Gamma\} & \text{Im}\{-\Gamma\} \\
  \text{Im}\{\Gamma\} & \text{Re}\{\Gamma\}
\end{bmatrix}
\right),
\tag{3.14}
$$

where the marginal distribution for the real component is $y^r \sim \mathcal{N}(0, \text{Re}\{\Gamma\})$, the exact form given above.

Denoting the DFT of each channel as $z = (I_C \otimes U)^\dagger y$, and using the circulant approximation for each matrix $\{K_q\}_{q=1}^Q$ along with (3.12) provides the equivalent marginal distribution for the DFT coefficients of $z \sim \mathcal{CN}(0, 2S)$, where

$$
S = (I_C \otimes U)^\dagger \Gamma (I_C \otimes U)
$$

$$
\approx H^{-1} \otimes I_N + \delta^{-1} \sum_{q=1}^Q (\beta_q^l \beta_q^\dagger) \otimes \Lambda_{K_q}.
\tag{3.15}
$$
Figure 3.2: Time-series data is drawn from a Gaussian process with a known CSM covariance kernel, where the domain restricted to a fixed number of seconds. A Gaussian process is then fitted to this data using the DFT approximation. The KL-divergence of the fitted marginal likelihood from the true marginal likelihood is shown.

To obtain this result, the property of Kronecker products

$$\left( A_1 \otimes B_1 \right) \left( A_2 \otimes B_2 \right) = \left( A_1 A_2 \right) \otimes \left( B_1 B_2 \right)$$

was used, leaving the additive Gaussian noise term $H^{-1} \otimes I_N$ untouched (since $U^T U = I_N$) and replacing each $K_q$ with its spectral density, $\delta^{-1}A_{\tilde{K}_q}$.

One important comment pertains to the power spectral density along the diagonal of $A_{\tilde{K}_q}$. While the Fourier transform of any real data $y^r$ results in a symmetric spectral density, the Fourier transform of circularly symmetric data $y$ results in a spectral density that equals zero for half of the spectrum. Therefore, elements along the diagonal of $A_{\tilde{K}_q}$ will equal zero for half the spectrum, i.e., the spectral density will contain only white noise for $\omega = \frac{2\pi}{N} \left[-\left\lfloor \frac{N-1}{2} \right\rfloor, \ldots, -1 \right]$.

3.2.3 Stability Analysis of the DFT Approximation

In this section, numerical results provide conditions required when using the DFT approximation in (3.15). This allows for one to define details of a particular application in order to determine if the DFT approximation to the CSM kernel is appropriate.
A CSM kernel is defined for two outputs with a single Gaussian component, \( Q = 1 \). The mean frequency and variance for this component are set to push the limits of the application. For example, with LFP data, low frequency content is of interest, namely greater than \( 1 \) Hz; therefore, we test values of \( \tilde{\mu}_1 \in \{ \frac{1}{2}, 1, 3 \} \) Hz. We anticipate variances at these frequencies to be around \( \tilde{\nu}_1 = 1 \) Hz. A conversion to angular frequency gives \( \mu_1 = 2\pi \tilde{\mu}_1 \) and \( \nu_1 = 4\pi^2 \tilde{\nu}_1 \). The covariance matrix \( \Gamma \) in (2.18) is formed using these parameters, a fixed noise variance, and \( N \) observations on a fixed time grid with sampling rate of 200 Hz. Data \( y \) are drawn from the marginal likelihood with this covariance matrix.

A new CSM kernel is fit to \( y \) using the DFT approximation, providing an estimate \( \hat{\Gamma} \). The KL divergence of the fitted marginal likelihood from the true marginal likelihood is

\[
\text{KL}(p(y|\hat{\Gamma})||p(y|\Gamma)) = \frac{1}{2} \left[ \log \frac{\det \Gamma}{\det \hat{\Gamma}} - N + \text{tr}(\Gamma^{-1}\hat{\Gamma}) \right],
\]

where \( | \cdot | \) and \( \text{tr}(\cdot) \) are the determinant and trace operators, respectively. Computing \( \frac{1}{N} \text{KL}(p(y|\hat{\Gamma})||p(y|\Gamma)) \) for various values of \( \tilde{\mu}_1 \) and \( N \) provides the results in Figure 3.2. This plot shows that the DFT approximation struggles to resolve low frequency components unless the series length is sufficiently long. Due to the approximation error, when using the DFT approximation on LFP data we a priori filter out frequencies below \( 1.5 \) Hz and perform analyses with a series length of 3 seconds. This ensures the DFT approximation represents the true covariance matrix. The following application of the CSM kernel uses these settings.

### 3.3 Predictive Distribution

Local field potentials (LFP) are low-frequency (< 200 Hz) electrophysiological signals that are generated by electrical currents from many neurons in the local region of the
Figure 3.3: The predictive distribution of an unobserved channel (the LFP of the Frontal Association Cortex) conditioned on LFPs from four other brain regions.

In this analysis, time-series are collected according to the methods of Dzirasa et al. (2011), and five regions are considered: the Basal Amygdala (BasalAmy), Dorsolateral Striatum (DLS), Dorsomedial Striatum (DMS), Dorsal Hippocampus (DHipp), and Frontal Association Cortex (FrA). The data were collected at 1000 Hz, but subsampled to 200 Hz for analysis. The CSM kernel in (A.6) was fit to a two second window of this 5-channel LFP data. This section explores the predictive distribution of unobserved data given the fitted CSM kernel.

The ability of the CSM kernel to extract cross-spectrum patterns and predict an unobserved channel given observations from other channels results from the kernels expressive nature. Although this prediction is displayed merely to convey the utility of the CSM kernel, experimentalists are often interested in this prediction task as a method of denoising channels. For example, the signal-to-noise ratio may be improved through principal component analysis (PCA) or linear models, where a channel is predicted by the optimal linear regression of the other observed channels (Musial et al., 2002).

In Figure 3.3, the predictive distribution of the Dorsal Hippocampus given data from the other four channels is shown. As a direct comparison, the simple linear
regression method of (Musial et al., 2002) is also shown. The mean function of the predictive distribution obtains an 8% better MSE performance than the linear regression method. The nonparametric nature of Gaussian processes allow the predictive distribution to be computed at any point along the entire input domain. This is an interpolation/extrapolation feature that is non-trivial in the broad field of dynamic linear models (Prado and West, 2010). Furthermore, treating each channel as an independent draw from a Gaussian process with an SM kernel is unable to extract useful channel information after a tenth of a second. This is due to the broad spread of frequencies (length-scale) present in the LFP signal.

These comparisons illustrate the expressiveness of the CSM kernel. Not only does the cross-spectrum provide significant knowledge about unobserved data, the kernel allows for desired non-parametric feature extraction.

3.4 Model Comparison

This section compares the model fit of the novel CSM kernel, the SM-LMC kernel, and the SE-LMC (squared exponential) kernel. For each of these models we allow Q=20, and vary the rank of the coregionalization matrices from rank-1 to rank-3. For a given rank, the CSM kernel always obtains the largest marginal likelihood for a window of LFP data, and the marginal likelihood always increases for increasing rank. To penalize the number of kernel parameters (e.g., a rank-3, Q=20 CSM kernel for 7 channels has 827 free parameters to optimize), the Akaike information criterion (AIC) is used for model selection (Akaike, 1974). Table 3.1 shows that a rank-2 CSM kernel is selected using this criterion, followed by a rank-1 CSM kernel. To show the rank-2 CSM kernel is consistently selected as the preferred model we report means and standard deviations of AIC value differences across 30 different randomly selected 3-second windows of LFP data.
Table 3.1: The mean and standard deviation of the difference between the AIC value of a given model and the AIC value of the rank-2 CSM model. Lower values are better.

<table>
<thead>
<tr>
<th>rank</th>
<th>SE-LMC</th>
<th>SM-LMC</th>
<th>CSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4770 (993)</td>
<td>512 (190)</td>
<td>109 (110)</td>
</tr>
<tr>
<td>2</td>
<td>5180 (1120)</td>
<td>325 (167)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>3</td>
<td>5550 (1240)</td>
<td>412 (184)</td>
<td>204 (71.7)</td>
</tr>
</tbody>
</table>
Neuroscience has made significant progress in learning how activity in specific neurons or brain areas correlates with behavior. One of the remaining mysteries is how to best represent and understand the way whole-brain activity relates to cognition: in other words, how to describe brain states (Gilbert and Sigman, 2007). Although different brain regions have different functions, neural activity across brain regions is often highly correlated. It has been proposed that the specific way brain regions are correlated at any given time may represent a “state” designed specifically to optimize neural computations relevant to the behavioral context an organism is in (Kohn et al., 2009). Unfortunately, although there is great interest in the concept of global brain states, little progress has been made towards developing methods to identify or characterize them.

The study of arousal is an important area of research relating to brain states. Arousal is a hotly debated topic that generally refers to the way the brain dynamically responds to varying levels of stimulation (Pfaff et al., 2008). One continuum of arousal used in the neuroscience literature is sleep (low arousal) to wakefulness (higher arousal). Another is calm (low arousal) to excited or stressed (high arousal).
arousal) (Lang and Bradley, 2010). A common electrophysiological measurement used to determine arousal levels is local field potentials (LFPs), or low-frequency (< 200 Hz) extracellular neural oscillations that represent coordinated neural activity across distributed spatial and temporal scales. LFPs are useful for describing overall brain states since they reflect activity across many neural networks. We examine brain states under different levels of arousal by recording LFPs simultaneously in multiple regions of the mouse brain, first, as mice pass through different stages of sleep, and second, as mice are moved from a familiar environment to a novel environment to induce interest and exploration.

In neuroscience, the analysis of electrophysiological time-series data is largely centered around dynamic causal modeling (DCM) (Friston et al., 2003), where continuous state-space models are formulated based on differential equations that are specifically crafted around knowledge of underlying neurobiological processes. However, DCM is not suitable for exploratory analysis of data, such as inferring unknown arousal levels, for two reasons: because the differential equations are driven by inputs of experimental conditions, and the analysis is dependent on a priori hypotheses about which neuronal populations and interactions are important. This work focuses on methods suitable for exploratory analysis.

Previously published neuroscience studies distinguished between slow-wave sleep (SWS), rapid-eye-movement (REM), and wake (WK) using proportions of high-frequency (33-55 Hz) gamma oscillations and lower frequency theta (4-9 Hz) oscillations in a brain area called the Hippocampus (Dzirasa et al., 2006; Gervasoni et al., 2004). As an alternative approach, recent statistical methods for tensor factorization (Rai et al., 2014) can be applied to short time Fourier transform (STFT) coefficients by factorizing a 3–way LFP tensor, with dimensions of brain region, frequency band and time. Distinct sleep states may then be revealed by clustering the inferred sequence of time-varying score vectors.
Although good first steps, the above two methods have several shortcomings: 1) They do not consider the time dependency of brain activity, and therefore cannot capture state-transition properties. 2) They cannot directly work on raw data, but require preprocessing that only considers spectral content in predefined frequency bins, thus leading to information loss. 3) They do not allow for individual brain regions to take on their own set of sub-state characteristics within a given global brain state. 4) Finally, they cannot leverage the shared information of LFP data across multiple animals.

In this chapter, several applications of Gaussian processes are presented to model such electrophysiological time-series measurements. Section 4.1 presents a Bayesian tensor framework for brain states, where each channel is modeled independently using the SM covariance kernel. In this work, spectral content is weakly coupled between brain regions, and the cross-phase spectrum is completely ignored. Section 4.2 addresses these issues by using the convenient structure of the CSM kernel within multi-output Gaussian processes.

4.1 Brain State Model

The model presented in this section overcomes the shortcomings of previously published brain-state methods by defining a sequence of brain states over a sliding window of raw, filtered LFP data. An infinite hidden Markov model (iHMM) (Beal et al., 2002) is imposed on these state assignments. Conditioned on this brain state, each brain region is assigned to a cluster in a mixture model. Each cluster is associated with a specific spectral content (or density) pattern, manifested through a spectral mixture kernel (Wilson and Adams, 2013) of a Gaussian process. Each window of LFP data is generated as a draw from this mixture of Gaussian processes. Thus, all animals share an underlying brain state space, of which, all brain regions share the underlying components of the mixture model.
4.1.1 Model

For each animal \( a \in \{1, \ldots, A\} \), the LFP time-series is recorded in \( R \) different regions, measured simultaneously. These time-series are split into sequential, sliding windows, \( y_{w}^{(ar)} \in \mathbb{R}^{N} \) for \( w \in \{1, \ldots, W\} \), such that windows are common across regions. These windows are chosen to be overlapping, thereby sharing data points between consecutive windows; nonoverlapping windows may also be used. Each window is considered as a single observation vector, and we wish to model the generative process of these observations, \( \{y_{w}^{(ar)}\} \).

The proposed model aims to describe the spectral content in each of these LFP signals, as a function of brain region and time. This is done by first assigning a joint “brain state” to each time window, \( \{s_{1}^{(a)}, \ldots, s_{W}^{(a)}\} \), shared across all brain regions \( \{1, \ldots, R\} \). The brain state is assumed to evolve in time as a latent Markov process. The LFP data from a particular brain region is assumed drawn from a mixture of Gaussian processes. The characteristics of each mixture component are shared across brain states and brain regions, with mixture weights that are dependent on these two entities.
Figure 4.2: Graphical representation of the state space model. The model first assigns a sequence of brain states, \( \{ s_{w}^{(a)} \}_{1}^{W} \), with Markovian dynamics to animal \( a \). Given state \( s_{w}^{(a)} \), each region is assigned to a cluster, \( z_{w}^{(ar)} = \ell \in \{1, \ldots, L\} \), and the data \( y_{w}^{(ar)} \) is generated from a Gaussian process with covariance function \( k(\tau; \theta_{\ell}, \gamma) \).

Brain state assignment

Within the generative process, each animal has a latent brain state for every time window, \( w \). This brain state is represented through a categorical latent variable \( s_{w}^{(a)} \), and an infinite hidden Markov model (iHMM) is placed on the state dynamics (Beal et al., 2002; Teh et al., 2005). This process is formulated as

\[
s_{w}^{(a)} \sim \text{Categorical}(\lambda_{s_{w-1}^{(a)}}), \quad \lambda_{g}^{(a)} \sim \text{DP}(\alpha_{0}\beta), \quad \beta \sim \text{GEM}(\gamma_{0}), \quad (4.1)
\]

where GEM is the stick-breaking process

\[
\beta_{h} = \beta_{h}' \prod_{i=1}^{h-1} (1 - \beta_{i}'), \quad \beta_{h}' \sim \text{Beta}(1, \gamma_{0}).
\]

Here, \( \{ \beta_{h} \}_{h=1}^{H} \) represents global transition probabilities to each state in a potentially infinite state space. For the stick-breaking process, \( H \to \infty \), but in a finite collection of data only a finite number of state transitions will be used and \( H \) can be efficiently truncated. Since the state space is shared across animals, we cannot predefine initial state assignments, \( s_{1}^{(a)} \). To remedy this, allow \( s_{1}^{(a)} \sim \text{Categorical}(\psi^{(a)}) \) and place a discrete uniform prior on \( \psi^{(a)} \) over the truncated state space.
Each animal is given a transition matrix $\Lambda^{(a)}$, where each row of this matrix is a transition probability vector $\lambda^{(a)}_{gh}$ such that the transition from state $g$ to state $h$ for animal $a$ is $\lambda^{(a)}_{gh}$, each centered around the global transition vector $\beta$. Because each animal’s brain can be structured differently (e.g., as an extreme case, consider a central nervous system disorder), $\Lambda^{(a)}$ is allowed to vary from animal to animal.

Assigning brain regions to clusters

For each brain state, mixture weights are drawn to define the distribution over clusters independently for each region $r$, centered around a global mixture $\eta$ using a hierarchical Dirichlet Process (Teh et al., 2005):

$$
\phi^{(r)}_{h} \sim \text{DP}(\alpha_{1}\eta), \quad \eta \sim \text{GEM}(\gamma_{1}),
$$

(4.2)

where $\phi^{(r)}_{h}$ is the probability of assigning region $r$ of a window with brain state $h$ to cluster $\ell$. This cluster assignment can be written as

$$
z_{w}^{(ar)} | z_{w}^{(a)} \sim \text{Categorical}(\phi^{(r)}_{s_{w}^{(a)}}).
$$

(4.3)

For each cluster $\ell$ there is a set of parameters, $\theta_{\ell}$, describing a Gaussian process (GP), detailed in the next section. One could consider the joint probability over cluster assignments for all brain regions as an extension of a latent nonnegative PARAFAC tensor decomposition (Murray and Dunson, 2013; Harshman, 1970). Refer to Appendix B for details on this tensor model.

Infinite Mixture of Gaussian Processes

Combining the clustering model with the SM kernel of Section 2.2.3, each cluster $\ell$ is associated with a distinct set of kernel parameters $\theta_{\ell}$. To generate the observations $\{y_{w}^{(ar)}\}$, where each $y_{w}^{(ar)} \in \mathbb{R}^{N}$ has observation times $t = \{t_{1}, \ldots, t_{N}\}$ such that $|t_{i} - t_{j}| = |i - j|\tau$ for all $i$ and $j$, consider a draw from the multivariate normal
distribution:

\[ \mathbf{y}_w^{(ar)} \sim \mathcal{N}(0, \Sigma_{z_w^{(ar)}}), \quad (\Sigma_{\ell})_{ij} = k(|t_i - t_j|; \theta_{\ell}, \gamma), \quad (4.4) \]

where each observation is generated from the cluster indicated by \( z_w^{(ar)} \), and each cluster is represented uniquely by a covariance matrix, \( \Sigma_{\ell} \), whose elements are defined through the covariance kernel \( k(\tau; \theta_{\ell}, \gamma) \). Therefore, the parameters \( \theta_{z_w^{(ar)}} \) describe the auto-correlation content associated with each \( \mathbf{y}_w^{(ar)} \).

This GP model contains the hyperparameters \( \{\theta_{\ell} = \{\omega^{\ell}, \mu^{\ell}, \nu^{\ell}\}\} \) and \( \gamma \). Priors may be placed on these parameters; for example, we use the uninformative priors \( \mu_q \sim \text{Uniform}(\mu_{\min}, \mu_{\max}) \), \( \nu_q \sim \text{Uniform}(0, \nu_{\max}) \) and \( \omega_q \sim \text{Gamma}(e_0, f_0) \). A bandpass filter is applied to the LFP signal from \( \mu_{\min} \) to \( \mu_{\max} \) Hz as a preprocessing step, so this prior knowledge is justified. Also, \( \nu_{\max} \) is set to prevent overfitting, and \( e_0 \) and \( f_0 \) are set to manifest a broad prior. Finally, we set the prior \( \gamma \sim \text{Gamma}(e_1, f_1) \) where the hyperparameters \( e_1 \) and \( f_1 \) are chosen to manifest a broad prior.

There are a few primary concerns with this formulation. First, this observation model ignores complex cross-covariance functions between regions. Although LFP measurements exhibit coherence patterns across regions, the generative model in (4.4) only weakly couples the spectral densities of each region through the brain state. In principle, the generative model could be extended to incorporate this coherence information. Second, (4.4) does not model the time-series itself as a stochastic process, but rather the preprocessed, ‘independent’ observation vectors. This shortcoming is not ideal, but the windowing process allows for efficient computation via the mixture of Gaussian processes.
4.1.2 Inference

In the following, latent model variables are represented by $\Omega = \{Z, S, \Phi, \eta, \Lambda, \beta, \Psi\}$, the kernel parameters to be optimized are $\Theta = \{\theta_i\}_{i=1}^L$, and $H$ and $L$ are upper limit truncations on the number of brain states and clusters, respectively. As described throughout this section, the proposed algorithm adaptively adjusts the truncation levels on the number of brain states, $H$, and clusters, $L$, through a series of split-merge moves. The joint probability of the proposed model is

$$p(Y, \Omega, \Theta) = p(Y|Z, \Theta)p(Z|\Phi, \Lambda, \Psi)p(\Phi|\eta)p(\eta)p(\Lambda|\beta)p(\beta)p(\Psi)p(\Theta)$$

$$= \left[\prod_{a,r,w} p(y_w^{(ar)}|z_w^{(ar)}, \Theta)p(z_w^{(ar)}|s_w^{(a)}, \Phi)\right] \left[p(\eta|\gamma_1) \prod_{r,h} p(\phi_h^(r)|\eta, \alpha_1)\right]$$

$$\left[\prod_a p(\zeta_1(a) | \psi(a))p(\psi(a)) \prod_{w=2}^W p(s_w^{(a)}|s_{w-1}^{(a)}, \Lambda^{(a)}) \right] \left[p(\beta|\gamma_0) \prod_{a,g} p(\lambda_y^{(a)}|\beta, \alpha_0)\right]$$

$$\left[p(\gamma|\epsilon_1, f_1) \prod_{q=1}^Q p(\omega_q|\epsilon_0, f_0)p(\mu_q|\mu^{min}, \mu^{max})p(\nu_q|\nu^{max})\right]. \quad (4.5)$$

A variational inference scheme is developed to update $\Omega$ and $\Theta$.

**Variational inference**

With variational inference, an approximate variational posterior distribution is sought that is similar to the true posterior distribution, $q(\Omega, \Theta) \approx p(\Omega, \Theta|Y)$. This variational posterior is assumed to have a factorization into simpler distributions, where

$$q(\Omega, \Theta) = q(Z)q(S)q(\Phi)q(\eta)q(\Lambda)q(\beta)q(\Psi)q(\Theta),$$

with further factorization

$$q(Z) = \prod_{a,r,w} \text{Cat}(z_w^{(ar)}, \phi_w^{(ar)}), \quad q(\Phi) = \prod_{h,r} \text{Dir}(\phi_h^{(r)}; \nu_h^{(r)}), \quad q(\eta) = \delta_{\eta*}(\eta),$$

$$q(S) = \prod_a q(s_w^{(a)}|s_{w-1}^{(a)}), \quad q(\Lambda) = \prod_{g,a} \text{Dir}(\lambda_y^{(a)}; \kappa_g^{(a)}), \quad q(\beta) = \delta_{\beta*}(\beta),$$

$$q(\Psi) = \prod_a \delta_{\psi(a)*}(\psi^{(a)}), \quad q(\Theta) = \prod_j \delta_{\Theta_j*}(\Theta_j), \quad (4.6)$$

where only necessary sufficient statistics of the latent factors $q(s_w^{(a)}|s_{w-1}^{(a)})$ are required, and the approximate posteriors of $\eta$, $\beta$, $\{\psi^{(a)}\}$ and $\{\Theta_j\}$ are represented by point estimates at $\eta^*$, $\beta^*$, $\{\psi^{(a)*}\}$ and $\{\Theta_j^*\}$, respectively.
The degenerate distributions $\delta_{\eta^*}(\eta)$ and $\delta_{\beta^*}(\beta)$ are described in previous work on variational inference for HDPs (Bryant and Sudderth, 2012; Liang et al., 2007). The idea is that the point estimates of the stick-breaking processes simplify the derivation of the variational posterior, and the authors of (Liang et al., 2007) show that obtaining a full posterior distribution on the stick-breaking weights has little impact on model fitting since the variational lower bound is not heavily influenced by the terms dependent on $\eta$ and $\beta$. Furthermore, the Dirichlet process is truncated for both the number of states and the number of clusters such that $q(z_w^{(ar)} = \ell) = 0$ for $\ell > L$ and $q(s_w^{(a)} = h) = 0$ for $h > H$. This truncation method (see Teh et al. (2007) for details) is notably different than other common truncation methods of the DP (e.g., Blei and Jordan (2004) and Kurihara et al. (2007)), and is primarily important for facilitating the split-merge inference techniques described in the following section.

In mean-field variational inference, the variational distribution $q(\Omega, \Theta)$ is chosen such that the Kullback-Leibler divergence of $p(\Omega, \Theta | Y)$ from $q(\Omega, \Theta)$,

$$D_{KL}(q(\Omega, \Theta)||p(\Omega, \Theta|Y)),$$

is minimized. This is equivalent to maximizing the evidence lower bound (also known as the variational free energy in the DCM literature),

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(Y, \Omega, \Theta)] - \mathbb{E}_q[\log q(\Omega, \Theta)],$$

where both expectations are taken with respect to the variational distribution. The resulting lower bound is

$$\mathcal{L}(q) = \mathbb{E}[\ln p(Y|Z, \Theta)] + \mathbb{E}[\ln p(Z, S|\Phi, \Lambda, \Psi)] + \mathbb{E}[\ln p(\Phi|\eta)] + \mathbb{E}[\ln p(\eta)] + \mathbb{E}[\ln p(\Lambda|\beta)] + \mathbb{E}[\ln p(\beta)] + \mathbb{E}[\ln p(\Psi)] + \mathbb{E}[\ln p(\Theta)] + \mathbb{H}[q(Z)] + \mathbb{H}[q(S)] + \mathbb{H}[q(\Phi)] + \mathbb{H}[q(\Lambda)],$$

(4.7)

where all expectations are with respect to the variational distribution, the hyperparameters are excluded for notational simplicity, and we define $\mathbb{H}[q(\cdot)]$ as the sum over
the entropies of the individual factors of \( q(\cdot) \). Due to the degenerate approximations for \( q(\eta), q(\beta), q(\Psi) \) and \( q(\Theta) \), these full posterior distributions are not obtained, and, therefore, the terms \( \mathbb{H}[q(\eta)], \mathbb{H}[q(\beta)], \mathbb{H}[q(\Psi)] \) and \( \mathbb{H}[q(\Theta)] \) are set to zero in the lower bound.

The updates for \( \zeta_w^{(ar)} \) and \( \nu_h^{(r)} \) are standard. Variational inference for the HDP-HMM is detailed in other work (e.g., see Beal (2003); Paisley and Carin (2009)). Details on variational inference for Bayesian HMM’s is provided in Appendix C. Using these methods, updates for \( \kappa_g^{(a)}, \psi^{(a)} \) and the necessary expected sufficient statistics of the factors of \( q(S) \) are realized. To note, provided the extension of Bayesian HMM’s to Bayesian hidden semi-Markov models (HSMM) in Appendix D, an HSMM is easily supplanted in place of the HMM.

Finally, updates for \( \beta^*, \eta^* \) and \( \{\Theta_j\} \) are non-conjugate, so a gradient-ascent method is performed to optimize these values. We use a simple resilient backpropagation (Rprop), though most line-search methods should suffice. Details on all updates and taking the gradient of \( \mathcal{L}(q) \) with respect to \( \beta, \eta \) and \( \{\Theta_j\} \) are found in Appendix E.

**Split-merge moves**

During inference, a series of split and merge operations are used to help the algorithm jump out of local optima (Jain and Neal, 2007). This work takes the viewpoint that two clusters (or states) should merge only if the variational lower bound increases, and, when a split is proposed for a cluster (or state), it should always be accepted, whether or not the split increases the variational lower bound. If the split is not appropriate, a future merge step is expected to undo this operation. In this way, the opportunity is provided for cluster and state assignments to jump out of local optima, allowing the inference algorithm to readjust assignments as desired.
Merge states: To merge states $h'$ and $h''$ into a new state $h$, new parameters are initialized as:

$$
\rho^{(a)}_{wh} = \rho^{(a)}_{wh'} + \rho^{(a)}_{wh''},
\kappa^{(a)}_{gh} = \kappa^{(a)}_{gh'} + \kappa^{(a)}_{gh''},
\beta^*_{h'} = \beta^*_{h''} + \beta^*_{h},
\nu^{(a)}_h = \nu^{(a)}_{h'} + \nu^{(a)}_{h''},
$$

such that the model now has a truncation at $H^{new} = H - 1$ states. In order to account for problems with merging two states in an HMM, a single restricted iteration is allowed, where only the state-dependent variational parameters in $\Omega^{new}$ are updated, producing a new distribution $q(\Omega^{new})$. The merge is accepted (i.e., $\Omega = \Omega^{new}$) if 

$$
L(q(\Omega^{new})) > L(q(\Omega)).
$$

Since these computations are not excessive, all possible state merges are computed and a small number of merges are accepted per iteration.

Merge clusters: To merge clusters $\ell'$ and $\ell''$ into a new cluster $\ell$, new parameters are initialized as:

$$
\zeta^{(ar)}_{w\ell} = \zeta^{(ar)}_{w\ell'} + \zeta^{(ar)}_{w\ell''},
\nu^{(r)}_{h\ell} = \nu^{(r)}_{h\ell'} + \nu^{(r)}_{h\ell''},
\eta^*_{\ell} = \eta^*_{\ell'} + \eta^*_{\ell''},
\theta^{new}_{\ell} = \theta^*,
$$

such that there is a truncation at $L^{new} = L - 1$ clusters. We set $\theta^* = \theta_{\ell'}$ for simplicity, and allow a restricted iteration of updates to $\Omega^{new}$ and $\theta^{new}_{\ell}$. The merge is accepted (i.e., $\Omega = \Omega^{new}$ and $\Theta = \Theta^{new}$) if the lower bound is improved, 

$$
L(q(\Omega^{new},\Theta^{new})) > L(q(\Omega,\Theta)).
$$

Since the restricted iteration for $\theta^{new}_{\ell}$ is expensive, only a few cluster merges may be proposed at a time. Therefore, merges are proposed for clusters with the smallest earth mover’s distance Rubner et al. (2000) between their spectral densities.

Split step: When splitting states and clusters, the opposite process to the initialization of the merging procedures described above is performed. For clusters, data points within a cluster $\ell$ are randomly chosen to stay in cluster $\ell$ or split to a new cluster $\ell'$. For splitting state $h$, the cluster assignment vector $\phi^{(r)}_h$ is replicated and windows within state $h$ are randomly chosen to stay in state $h$ or split to a new cluster $h'$. Regardless of how this effects the lower bound, a split step is always accepted.

For implementation details, we allow the model to accept 3 state merges every third iteration, propose 5 cluster merges every third iteration, and split one state and one cluster every third iteration. Therefore, every iteration may affect the truncation
level of either the number of states or clusters. A ‘burn-in’ period is allowed before
starting the proposing of splits/merges, and a ‘burn-out’ period is employed in which
split proposals cease. In this way, the algorithm has guarantees of improving the
lower bound only during iterations when a split is not proposed, and convergence
tests are only considered during the burn-out period.

4.1.3 Results

For all results, we set $Q = 10$, $H = 15$, $L = 25$, stop the ‘burn-in’ period after
iteration 6, and start the subsequent computation period after iteration 25. Hyper-
parameters were set to $\gamma_0 = \gamma_1 = .01$, $\alpha_0 = \alpha_1 = 1$, $\mu^{\min} = 0$, $\mu^{\max} = 50$, $\nu^{\max} = 10$,
and $e_0 = f_0 = 10^{-6}$. In all results, the model was seen to converge to a local optima
after 30 iterations, and each iteration took on the order of 20 seconds using Matlab
code on a PC with a 2.30GHz quad-core CPU and 8GB RAM.

Figure 4.3 shows results on the toy data. The model correctly recovers exactly 3

![Figure 4.3: Toy data results. Top row shows the generated toy data. From left to
right: the five spectral functions, each associated with a component in the mixture
model; the probability of each of these five components occurring for all five regions
in each brain state; the generated brain state assignments from a 3-state HMM along
with the generated cluster assignments for the five simulated regions. The bottom
row shows the results of our model. On the left, a comparison of the recovered state
vs. the true state for all time; on the right, an alignment of the five recovered kernels
to the spectral density ground truth.](image-url)
states and 5 clusters, and, as seen in the figure, the state assignments and spectral densities of each cluster component are recovered almost perfectly. The model was implemented for different values of the noise variance, $\gamma^{-1}$, and, though not shown, in all cases the noise variance was recovered accurately during inference, implying the spectral mixture kernels are not overfitting the noise. In this way, we confirm that the inference scheme recovers a ground truth. For further model verification, ten-fold cross-validation was used to compute predictive probabilities for held-out data (reported in Table 4.1), where we compare to two simpler versions of our model: 1) the HDP-HMM on brain states in (4.1) is replaced with an HDP, and 2) a single brain state. For the HDP-HMM, the hold-out data was considered as ‘missing data’ in the training data and the window index was used to assign time-dependent probabilities over clusters, whereas in the HDP and Single State models it was simply withheld from the training data. We see large predictive performance gains when considering multiple brain states, and even more improvement on average (though modest) when considering an HDP-HMM.

The sleep and novel environment results are presented in Figures 4.4 and 4.5, respectively. With the sleep dataset, our results are compared with the two methods discussed in the Introduction: that of (Dzirasa et al., 2006; Gervasoni et al., 2004), and the tensor method of Rai et al. (2014). We refer to the Supplemental Material for exact specifications of the tensor method.

For each of these datasets, we infer the intended arousal states. In the novel environment data, we observe broad arousal changes at 9–minutes for all animals, as expected. In the sleep data, we successfully uncover at least as many states as the simple approach of (Dzirasa et al., 2006; Gervasoni et al., 2004), to include SWS, REM and WK states. Thus far neuroscientists have focused primarily on 2 stages of sleep (NREM and REM), but as many as 5 have been discussed (4 different stages of NREM sleep, and 1 stage of REM). Different stages of sleep affect memory and
behavior in different ways (e.g., see Tucker et al. (2006)), as does the number of times animals transition between these states (Rolls et al., 2011). Our results suggest that there may be even more levels of sleep that should be considered (e.g., transition states and sub states). This is very interesting and important for neuroscientists to know, because it is possible that each of our newly observed states could affect memory and behavior in different ways. There is no other published method that has provided evidence of these other states.

In addition to brain states, we infer spectral information for each brain region through cluster assignments. Though not the primary focus of this work, it is interesting that groups of brain regions tend to share similar attributes. In Figure 4.4, we have sorted brain regions into groups based on cluster assignment similarity, essentially recovering a ‘network’ of the brain. This underscores the power of the proposed method: not only do we develop unsupervised methods to classify whole-brain activity into states, we infer the cross-region/animal relationships within these

**Figure 4.4:** Sleep data results. **Top:** A comparison of brain state assignments from our method to two other methods. **Bottom Left:** Spectral density of the 7 inferred clusters. **Middle Left:** Cluster assignments over time for 16 different brain regions, sorted by similarity. **Middle Right:** Given brain states 1, 2 and 3, we show cluster assignment probabilities for 4 different brain regions: the hippocampus (D_Hipp), nucleus accumbens core (NAc_core), orbitofrontal cortex (OFC) and ventral tegmental area (VTA) from left to right, respectively. **Right:** State assignments of our method and the tensor method conditioned on the method of Dzirasa et al. (2006).
**Figure 4.5:** Novel environment data results. **Left:** The log spectral density of the 6 inferred clusters. **Middle:** State assignments for all 9 animals over a 30 minute period. There are 7 inferred states, and each state has a distribution over clusters for each region, as seen on the **right**.

### 4.2 Including the CSM Kernel in a Bayesian Hierarchical Model

The model presented in Section 4.1 is admittedly complicated. Dependencies between brain regions are imposed through an infinite tensor factorization on cluster assignments, and the definition of a ‘brain state’ is loose with limited interpretability.

Although this model is useful in providing a much-needed framework for analyzing LFP signals (Ulrich *et al.*, 2014), it lacks the explicit cross-channel dependencies provided by the novel CSM kernel presented in Chapter 3.

In the model provided in this section, we cluster segments of the LFP signal into discrete “brain states” (Ulrich *et al.*, 2014). Each brain state is represented by a unique cross-spectrum provided by the CSM kernel. The use of the full cross-spectrum allows for a more detailed analysis of the brain’s activity over time.

#### Table 4.1: Average held-out log predictive probability for different priors on brain states: HDP-HMM, HDP, and a single state. The data consists of $W$ time-series windows for $R$ regions of $A$ animals; at random, 10% of these time-series windows were held-out, and the predictive distribution was used to determine their likelihood.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>HDP-HMM</th>
<th>HDP</th>
<th>Single State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toy ($\times 10^5$)</td>
<td>$-1.686 \pm 0.053$</td>
<td>$-1.688 \pm 0.053$</td>
<td>$-1.718 \pm 0.054$</td>
</tr>
<tr>
<td>Sleep ($\times 10^6$)</td>
<td>$-1.677 \pm 0.030$</td>
<td>$-1.682 \pm 0.020$</td>
<td>$-1.874 \pm 0.019$</td>
</tr>
<tr>
<td>Novel ($\times 10^9$)</td>
<td>$-5.932 \pm 0.040$</td>
<td>$-5.973 \pm 0.034$</td>
<td>$-6.962 \pm 0.063$</td>
</tr>
</tbody>
</table>

44
spectrum to define brain states is supported by previous work discovering that 1) the power spectral density of LFP signals indicate various levels of arousal states in mice (Dzirasa et al., 2006; Ulrich et al., 2014), and 2) frequency-dependent phase synchrony patterns change as animals undergo different conditions in a task (Gregoriou et al., 2009; Sauseng and Klimesch, 2008; Sweeney-Reed et al., 2014).

4.2.1 Model

The vector-valued observations from $C$ channels are segmented into $W$ contiguous, non-overlapping windows. The windows are common across channels, such that the $C$-channel data for window $w \in \{1, \ldots, W\}$ are represented by $y_n^w = [y_{n1}^w, \ldots, y_{nC}^w]^T$ at sample location $x_n^w$. Given data, each window consists of $N_w$ temporal samples, but the model is defined for any set of sample locations in the input space.

We model the observations $\{y_n^w\}$ as emissions from a hidden Markov model (HMM) with $L$ hidden, discrete states. State assignments are represented by latent variables $\zeta_w \in \{1, \ldots, L\}$ for each window $w \in \{1, \ldots, W\}$. In general, $L$ is a set upper bound of the number of states (brain states (Ulrich et al., 2014), or “clusters”), but the model can shrink down and infer the number of states needed to fit the data. This is achieved by defining the dynamics of the latent states $\{\zeta_w^w\}_{w=1}^W$ according to

![Figure 4.6: A short segment of LFP data recorded from the basolateral amygdala and infralimbic cortex is shown on the left. The cross-amplitude and phase spectra are produced using Welch’s averaged periodogram method Welch (1967) for several consecutive 5 second windows of LFP data. Frequency dependent phase synchrony lags are consistently present in the cross-phase spectrum, motivating the CSM kernel. This frequency dependency aligns with preconcieved notions of bands, or brain waves (e.g., 8-12 Hz alpha waves).](image-url)
a Bayesian HMM (MacKay, 1997):

\[
\zeta_1 \sim \text{Categorical}(\rho_0),
\]

\[
\zeta_w \sim \text{Categorical}(\rho_{\zeta_{w-1}}) \ \forall w \geq 2,
\]

\[
\rho_0, \rho_{\ell} \sim \text{Dirichlet}(\nu),
\]

where the initial state assignment is drawn from a categorical distribution with probability vector \(\rho_0\) and all subsequent states assignments are drawn from the transition vector \(\rho_{\zeta_{w-1}}\). Here, \(\rho_{\ell h}\) is the probability of transitioning from state \(\ell\) to state \(h\). The vectors \(\{\rho_0, \rho_1, \ldots, \rho_L\}\) are independently drawn from symmetric Dirichlet distributions centered around \(\nu = [1/L, \ldots, 1/L]\) to impose sparsity on transition probabilities. In effect, this allows the model to learn the number of states needed for the data (i.e., fewer than \(L\)) (Beal, 2003).

Each cluster \(\ell \in \{1, \ldots, L\}\) is assigned GP parameters \(\theta_{\ell}\). The latent cluster assignment \(\zeta_w\) for window \(w\) indicates which set of GP parameters control the emission distribution of the HMM:

\[
y_n^w \sim \mathcal{N}(f_w(x_n^w), H_{\zeta_w}^{-1}), \quad f_w(x) \sim \mathcal{GP}(0, K(x, x'; \theta_{\zeta_w})),
\]

where \((K(x, x'; \theta_{\ell}))_{c,c'} = k_{\text{CSM}}^{c,c'}(x, x'; \theta_{\ell})\) is the CSM kernel, and the cluster-dependent precision \(H_{\zeta_w} = \text{diag}(\eta_{\zeta_w})\) generates independent Gaussian observation noise. In this way, each window \(w\) is modeled as a stochastic process with a multi-channel cross-spectrum defined by \(\theta_{\zeta_w}\).

While this section introduces a Bayesian HMM to model the evolution of brain states (see Appendix C for details), this is merely a modeling decision. If the duration that a brain state persists for is of importance, an experimentalist could easily supplant the Bayesian HMM described here with a more general Bayesian hidden semi-Markov model (HSMM). The HSMM as described in Appendix D, has the ability to place any duration distribution on the length of time an animal stays within a state, rather than the assumed geometric distribution of an HMM model.
4.2.2 Inference

A convenient notation vectorizes all observations within a window, 

\[ \mathbf{y}^w = \text{vec}([\mathbf{y}_1^w, \ldots, \mathbf{y}_{N_w}^w]^T), \]

where \( \text{vec}(\mathbf{A}) \) is the vectorization of matrix \( \mathbf{A} \); i.e., the first \( N_w \) elements of \( \mathbf{y}^w \) are observations from channel 1, up to the last \( N_w \) elements of \( \mathbf{y}^w \) belonging to channel \( C \). Because samples are obtained on an evenly spaced temporal grid, we fix \( N_w = N \) and align relative sample locations within a window to an oracle \( \mathbf{x}^w = \mathbf{x} = [x_1, \ldots, x_N]^T \) for all \( w \).

The model in Section 4.2.1 generates the set of observations \( \mathbf{Y} = \{\mathbf{y}^w\}_{w=1}^W \) at aligned sample locations \( \mathbf{x} \) given kernel parameters \( \Theta = \{\theta_\ell, \eta_\ell\}_{\ell=1}^L \) and model variables \( \Omega = \{\{\rho_\ell\}_{\ell=0}^L, \{\zeta_w\}_{w=1}^W\} \). The joint probability of the proposed model is

\[
p(\mathbf{Y}, \Omega | \Theta, \mathbf{x}) = p(\mathbf{y}^1 | \Theta, \zeta_1, \mathbf{x}) p(\zeta_1 | \rho_0) \left[ \prod_{w=2}^W \prod_{\ell=1}^L p(\mathbf{y}_w^w | \Theta, \zeta_w, \mathbf{x}) p(\zeta_w | \rho_{\zeta_{w-1}}) \right] \left[ \prod_{\ell=0}^L p(\rho_\ell | \nu) \right]. \tag{4.9} \]

The latent variables \( \Omega \) are inverted using mean-field variational inference (Beal, 2003), obtaining an approximate posterior distribution \( q(\Omega) \) that is assumed to factorize among it’s coordinates, such that

\[
q(\Omega) = q(\zeta_{1,W}) \prod_{\ell=0}^L \text{Dir}(\rho_\ell; \alpha_\ell).
\]

The approximate posterior is chosen to minimize the KL divergence to the true posterior distribution \( p(\Omega | \mathbf{Y}, \Theta, \mathbf{x}) \) using the standard variational EM method detailed in Appendix C. During each iteration of the variational EM algorithm, the kernel parameters \( \Theta \) are chosen to maximize the expected marginal log-likelihood 

\[
Q = \sum_{w=1}^W \sum_{\ell=1}^L q(\zeta_w = \ell) \log \mathcal{N}(\mathbf{y}_w^w; \mathbf{0}, \Gamma_\ell).
\]
via gradient ascent, where \( q(\zeta_w = \ell) \) is the marginal posterior probability that window \( w \) is assigned to brain state \( \ell \), and \( \Gamma_\ell = \text{Re}(\tilde{\Gamma}_\ell) \) is the CSM kernel matrix for state \( \ell \) with the complex form \( \tilde{\Gamma}_\ell = \sum_q B^\ell_q \otimes \tilde{k}_{SG}(x, x; \theta_\ell) + H^\ell_q \otimes I_N \). Performing gradient ascent requires the derivatives \( \frac{\partial q}{\partial \theta_j} = \frac{1}{2} \sum_{w,\ell} \text{tr}((\alpha_{\ell w} \alpha_{\ell w}^T - \Gamma^{-1}_\ell) \frac{\partial \Gamma}{\partial \theta_j}) \) where \( \alpha_{\ell w} = \Gamma^{-1}_\ell y^w \) (Rasmussen and Williams, 2006). A naïve implementation of this gradient requires the inversion of \( \Gamma_\ell \), which has complexity \( O(N^3 C^3) \) and storage requirements \( O(N^2 C^2) \) since a simple method to invert a sum of Kronecker products does not exist. During inference, we use the DFT approximation presented in Section 3.2 to alleviate the naïve cost of a Gaussian process.

4.2.3 Results

We analyze 12 hours of LFP data of a mouse transitioning between different stages of sleep (Dzirasa et al., 2006; Ulrich et al., 2014). Observations were recorded simultaneously from 4 channels (Dzirasa et al., 2011), high-pass filtered at 1.5 Hz, and subsampled to 200 Hz. Using 3 second windows provides \( N = 600 \) and \( W = 14,400 \). The HMM was implemented with the number of kernel components \( Q = 15 \) and the number of states \( L = 7 \). This was chosen because sleep staging tasks categorize as many as seven states: various levels of rapid eye movement, slow wave sleep, and wake (Tucker et al., 2006). Although rigorous model selection on \( L \) is necessary to draw scientific conclusions from the results, the purpose of this experiment is to illustrate the utility of the CSM kernel in this application.

An illustrative subset of the results are shown in Figure 4.7. The full cross-spectrum is shown for a single state (state 7), and the cross-spectrum between the Dorsomedial Striatum and the Dorsal Hippocampus are shown for all states. Furthermore, we show the progression of these brain state assignments over 3 hours and compare them to states from the method of (Dzirasa et al., 2006), where statistics of the Hippocampus spectral density were clustered in an ad hoc fashion. To the
Figure 4.7: A subset of results from the Bayesian HMM analysis of brain states. In the upper left, the full cross-spectrum for an arbitrary state (state 7) is plotted. In the upper right, the amplitude (top) and phase (bottom) functions for the cross-spectrum between the Dorsomedial Striatum (DMS) and Hippocampus (DHipp) are shown for all seven states. On the bottom, the maximum likelihood state assignments are shown and compared to the state assignments from Dzirasa et al. (2006). The same colors between the CSM state assignments and the phase and amplitude functions correspond to the same state. These colors are aligned to the Dzirasa et al. (2006) states, but there is no explicit relationship between the colors of the two state sequences.

best of our knowledge, this method represents the most relevant and accurate results for sleep staging from LFP signals in the neuroscience literature. From these results, it is apparent that our clusters pick up sub-states of (Dzirasa et al., 2006). For instance, states 3, 6, and 7 all appear with high probability when (Dzirasa et al., 2006) classifies state 3. Observing the cross-phase function of sub-state 7 reveals striking differences from other states in the theta wave (4-7 Hz) and the alpha wave (8-15 Hz). This cross-phase function is nearly identical for states 2 and 5, implying that significant differences in the cross-amplitude spectrum may have played a role in identifying the difference between these two brain states.

Many more of these interesting details exist due to the expressive nature of the
CSM kernel. As a full interpretation of the cross-spectrum results is not the focus of this work, we contend that the CSM kernel has the potential to have a tremendous impact in fields such as neuroscience, where the dynamics of cross-spectrum relationships of LFP signals are of great interest.
The definition of discrete brain states that evolve according to a hidden Markov model does not represent reality. While discrete brain states were successfully applied in the previous sections to classify various levels of arousal (e.g., sleep levels), brain activity is truly dynamic. A more logical application describes the ‘brain state’ as evolving along a continuous low-dimensional manifold. In this section, the CSM kernel is extended to a factor analysis model such that the factor scores may represent an instantaneous brain state. This method resembles that of Titsias and Lázaro-Gredilla (2011), where each time window is represented as a linear combination of latent functions with distinct cross-spectral content. However, this work marginalizes out the latent factors during inference, thereby sharing only the spectral content across windows and not latent functions.

5.1 Model

Consider a slight modification to the previous application,
\[
y^w_n \sim \mathcal{N}(f_w(x^w_n), H^{-1}_w),
\]
(5.1)
where the precision \( H_w \) is now no longer cluster dependent, and the latent function \( f_w(x_n^w) \) at input location \( x_n^w \) represents the underlying process generating \( y_n^w \). Instead of clustering the latent functions across time windows, a factor model is constructed according to

\[
f_w(x) = \sum_{\ell=1}^L s_{w\ell} f^\ell_w(x),
\]

(5.2)

where a linear combination of \( L \) latent functions represents \( f_w(x) \). The parameters \( s_w = [s_{w1}, \ldots, s_{wL}]^T \) are here referred to as the vector of factor scores, and are unique to each time window.

For each time window, the \( L \) latent factors are independent draws from a Gaussian process

\[
f^\ell_w(x) \sim \mathcal{GP}(0, K_{CSM}(x, x'; \theta_\ell)),
\]

(5.3)

where each factor is allowed its own set of kernel parameters \( \theta_\ell \). In this way, a latent factor is described by the multi-channel cross-spectrum of the CSM kernel. However, each window independently draws a latent function for each factor. Although the functions themselves are not shared across windows (as in Titsias and Lázaro-Gredilla (2011)), their spectral properties are shared. For identifiability, the CSM kernels are restricted to their correlation function, such that

\[
\max (\text{diag}(K_{CSM}(0, 0; \theta_\ell))) = 1 \quad \forall \ell,
\]

(5.4)

where the maximum correlation between all input/output locations is forced to equal to 1. Importantly, the factor score \( s_{w\ell} \) may now be interpreted as the variance associated with factor \( \ell \) in window \( w \).

Marginalizing out each factor \( f^\ell_w(x) \), the latent function representing the observed data is characterized by the Gaussian process

\[
f_w(x) \sim \mathcal{GP}
\left(0, \sum_{\ell=1}^L s_{w\ell}^2 \sum_{q=1}^Q \tilde{B}_q \otimes K_{SG}(x, x'; \theta^\ell_q)\right).
\]

(5.5)
The set of parameters to optimize is now $\Theta = \left\{ \{\tilde{B}_q^\ell, \theta_q^\ell\}_{q, \ell}, \{H_w, s_w\}_{w=1}^W \right\}$, which contains a large number of parameters to optimize in the training set. However, for new test windows when $\tilde{B}_q^\ell$ and $\theta_q^\ell$ are given, each window is identified by only $(L + C)$ parameters.

The factor scores may be considered as a non-linear embedding of the data onto a low-dimensional manifold. Due to the construction of an interpretable generative process, these factor scores are extremely powerful. That is, we understand how changes in the factor scores impact the cross-spectral properties of the generative process. This interpretation does not exist for certain discriminative models that provide non-linear projections onto low-dimensional manifolds, e.g., many non-generative deep neural networks. However, learning the manifold representation of a new datum requires solving an expensive inverse problem. While feed-forward neural networks are extremely fast at test, their inability to easily interpret the meaning of hidden network structures is a major deterrent. This method may be seen as a generative model with a dictionary of Gabor kernels in an analogous fashion to the convolutional dictionaries (Mairal et al., 2008; Pu et al., 2015) in dictionary learning.

5.2 Inference

Inference proceeds by maximizing the marginal likelihood:

$$
\ln p(Y|\Theta) = -\frac{1}{2} \sum_{w=1}^W \left( \log |\Sigma_w| + y_w^T \Sigma_w^{-1} y_w \right) - \frac{NCW}{2} \log(2\pi) \tag{5.6}
$$

where $\Sigma_w = (s_w^T \otimes I)K(s_w \otimes I) + H_w^{-1}$ and $K = \text{blkdiag}((K^{(\ell)})_{\ell=1}^L) \in \mathbb{R}^{NCL \times NCL}$ such that each

$$
K^{(\ell)} = \sum_{q=1}^Q \tilde{B}_q^\ell \otimes K_{SG}(x, x'; \theta_q^\ell) \in \mathbb{R}^{NC \times NC} \tag{5.7}
$$

is the Gram matrix associated with the $N$ input locations and $C$ output channels.
The inverse
\[ \Sigma_w^{-1} = \left[ (s_w^T \otimes I) K (s_w \otimes I) + H_w^{-1} \right]^{-1} \] (5.8)
is required for evaluation of partial derivatives of the log marginal likelihood. In particular, \( \Sigma_w^{-1} \) is indirectly required for computation of the model fit term, and directly required for the computation of the complexity term, as illustrated in Section 2.4.

At first appearance, this inverse looks to be a good candidate for use under the Woodbury identity, but the result provides
\[ H_w - H_w (s_w^T \otimes I) \left( K^{-1} + (s_w \otimes I) H_w (s_w^T \otimes I) \right)^{-1} (s_w \otimes I) H_w, \] (5.9)
which requires the inversion of an \( NCL \times NCL \) matrix instead of the original \( NC \times NC \) matrix. With smaller dimension, the original formulation is computationally more feasible. However, the latter formulation illustrates the difficulty in obtaining the inverse of a sum of Kronecker products. Specifically, while \( K \) is a block diagonal matrix, the term \((s_w \otimes I) H_w (s_w^T \otimes I)\) establishes dependencies between the block structures.

Another consideration, as an alternative to use of the Woodbury identity, is to write the eigendecomposition \( K = QAQ^T \), such that the eigenvectors and eigenvalues are respectively represented by
\[ Q = \text{blkdiag}\left(\{Q^{(\ell)}\}_{\ell=1}^L\right) \quad \text{and} \quad \Lambda = \text{blkdiag}\left(\{\Lambda^{(\ell)}\}_{\ell=1}^L\right), \] (5.10)
where each \( Q^{(\ell)} \) and \( \Lambda^{(\ell)} \) respectively contain the eigenvectors and eigenvalues of \( K^{(\ell)} \). Given \( S_w = \frac{s_w^T}{||s_w||} \otimes I \), it is then possible to write
\[ \Sigma_w = S_w Q \left( ||s_w||^2 \Lambda + \sigma^2 I \right) Q^T S_w^T. \] (5.11)
However, because \( S_w \) is a fat matrix, a left pseudo-inverse such that \( S_w^T S_w = I \) does not exist, thereby making a direct inversion according to the factorization in
unattainable. This again confirms that decoupling the factor scores from the inverse only exists for the case where $L = 1$. In other words, we cannot obtain a simplification for the inverse of a weighted sum of matrices.

The above has shown that the inverse in (5.8) must be preformed independently for each window. This implies that the most expensive step in obtaining the gradients of the marginal likelihood will scale linearly with the number of windows. Luckily, with the spectral density approximation method introduced in Chapter 3, the inversion in (5.8) may be performed in $O(NC^3)$ computations per window. The overall per-iteration cost of obtaining a gradient for each CSM kernel parameter is $O(WNC^3)$. Importantly, when $C = 1$, this method will scale linear in both $N$ and $W$.

The best way to avoid the computational complexity of the matrix inversion in (5.8) is to treat the latent functions $f_w^n(x)$ as model parameters to invert. This is often the preferred method with Gaussian processes (e.g., Titsias and Lázaro-Gredilla (2011)), and Lázaro-Gredilla and Titsias (2011) introduce a non-standard variational inference approach to obtain more accurate variational posterior representations in these scenarios. However, this approach would introduce a total of $NLW$ additional parameters to infer, and, as a result, a large number of local optima are introduced into the solution space. Thus, we avoid this undesirable approach.

Zhang (2007) outline an expectation maximization (EM) method to obtain maximum likelihood estimates for the coregionalization matrices in the linear model of coregionalization (LMC). This method could be extended to this scenario, where obtaining maximum likelihood estimates for both the coregionalization matrices and factor scores for all $L$ factors is of interest. However, Zhang (2007) still requires the explicit inversion of $\Sigma_w$. Because the covariance kernels of interest extract meaningful features of the data (i.e., spectral content), we find that marginalizing out the latent functions $f_w^n(x)$ allows for kernel and coregionalization parameters to be
Task label: \( z_w \)

Latent states:

Observation \( y_w \):

**Figure 5.1:** Graphical model of supervised max-margin learning of latent states. Latent states are inferred to both fit observations \( y_w \) and predict task labels \( z_w \).

jointly optimized via gradient methods.

### 5.2.1 Max-Margin Discrimination of Factor Scores

In this latent factor setting, one may desire the factors to both uncover data-fitting latent structure and inherit strong predictive power. Often, side information is recorded, and a (semi-)supervised formulation allows the latent features to take this information into account during training. During testing, the model may then be predictive of this side information, as illustrated in Figure 5.1. While discrimination is often done in a downstream processing component (e.g., fit a latent factor model, then train an SVM independently on the factor scores), the ability to jointly infer model parameters with a max-margin predictive model (Jiang et al., 2012) improves predictive power while simultaneously uncovering latent structure.

Each window is provided a class label \( z_w \in \{-1, 1\} \). We desire the factor scores \( \tilde{s}_w = s_w \circ s_w \in \mathbb{R}^L \) in the factor analysis model to be predictive of the class label \( z_w \) for each window \( w \). A max-margin optimization problem is designed to find the optimal hyperplane that separates the two classes (Cortes and Vapnik, 1995).

In particular, classification parameters \( \Psi \) are introduced, and a linear discriminant function is defined by \( g(\tilde{s}_w; \Psi) = \beta^T \tilde{s}_w + b \), with \( \Psi = \{\beta, b\} \). The classification rule \( \hat{z}_w = \text{sign}(g(\tilde{s}_w; \Psi)) \) is used to form a prediction of the class label.
For separable classes, the optimization problem
\[
\arg\min_{\Theta, \Psi} \sum_{w=1}^{W} - \log p(Y^w|\Theta) + \frac{1}{2}||\beta||^2
\]
(5.12)

s.t.: \(z_w g(\tilde{s}_w; \Psi) \geq 1 \quad \forall w\)
enforces maximum separability between classes by satisfying the hinge loss function \((1 - z_w g(\tilde{s}_w; \Psi))_+ = 0\) for all windows and placing an \(\ell_2\)-regularization penalty on \(\beta\) for identifiability. The hinge loss function is defined as \((x)_+ = \max(0, x)\).

In the case that a hyperplane cannot exactly separate all data examples, a more generalized soft-margin version includes a slack variable \(\xi_w\) for each window,
\[
\arg\min_{\Theta, \Psi} \sum_{w=1}^{W} - \log p(Y^w|\Theta) + \frac{1}{2}||\beta||^2 + \sum_{w=1}^{W} \lambda \xi_w
\]
(5.13)

s.t.: \(z_w g(\tilde{s}_w; \Psi) \geq 1 - \xi_w \quad \forall w\)
\(\xi_w \geq 0 \quad \forall w\)
allowing for the discriminant function to separate the majority of examples correctly by penalizing for the \(\ell_1\)-norm of the slack variables. Here, the set of classification parameters are \(\Psi = \{\beta, b, \xi\}\).

5.2.2 Model Updates

An alternating minimization procedure is used to obtain a maximum likelihood estimate for the parameters \(\Theta\) and \(\Psi\). The model fit parameters \(\Theta\) are chosen to minimize the Lagrangian dual
\[
\arg\min_{\Theta} \sum_{w=1}^{W} (- \log p(Y^w|\Theta) + c(1 - z_w g(\tilde{s}_w; \Psi))_+)
\]
(5.14)

s.t.: \(\max(\text{diag}(K_{CSM}(0, 0; \theta_\ell))) = 1 \quad \forall \ell\)
where \(c\) is a tuning parameter that must be set a priori. Without a closed form solution, any gradient descent method may be used to optimize the parameters \(\Theta\).
We implement resilient backpropagation for robust gradient descent, which is detailed in Algorithm 2.4.1. To satisfy the correlation constraint \(\max_{p} \text{diag}(\mathbf{K}_{\text{CSM}}(0, 0; \theta_{\ell})) = 1\), kernel parameter values are projected onto the feasible set after each gradient step. This amounts to normalizing the coregionalization matrices such that the maximum autocovariance function for all regions equals 1 at \(\tau = 0\) for all factors.

To optimize the classification parameters \(\Psi\), the primal problem is

\[
\arg\min_{\Psi} \frac{1}{2} ||\beta||^2 + \lambda \sum_{w=1}^{W} \xi_w
\]

s.t.: \(z_w g(\tilde{s}_w; \Psi) \geq 1 - \xi_w \quad \forall w\)
\(\xi_w \geq 0 \quad \forall w\)

which is the primal form of a standard support vector machine (SVM) with a soft-margin hyperplane (Cortes and Vapnik, 1995). Many software packages include implementations to solve the dual form of (5.15).

Due to the computational expense of solving (5.14) via gradient descent, an SVM is trained according to (5.15) every iteration of the gradient descent algorithm that solves (5.14). This process is summarized in Algorithm 2.

### 5.2.3 Discrimination Ensembles

This section describes a potential extension to the max-margin factor model detailed in this section. In the described model above, a single classifier is trained
for each classification task. Alternatively, it is possible to train an ensemble of models and allow the loss function and decision rule to average over the ensemble mixture (Jaakkola et al., 1999; Zhu et al., 2008). Known as maximum entropy discrimination, these methods have been extended to supervised Bayesian generative processes (Zhu et al., 2009, 2013; Jiang et al., 2012), and even for online scenarios Shi and Zhu (2014).

Instead of optimizing for point estimates of $\beta$ and $s_w$, prior distributions may be defined, e.g., $p(\beta) = \mathcal{N}(0, I)$ and $p(s_w) = \mathcal{N}(0, I)$, and a mean-field variational posterior

$$q(\beta, S) = q(\beta) \prod_{w=1}^{W} q(s_w)$$

may be optimized instead. While updates for $q(\beta)$ are detailed in the literature (Zhu et al., 2009), obtaining the approximate posterior distributions $q(s_w)$ remains difficult due to the kernel embeddings of the variational parameters in the Gaussian processes. Recent Monte Carlo variational inference techniques would provide a possible solution to obtaining noisy gradient steps on the variational parameters (Kingma and Welling, 2014). Furthermore, the potentially high variance of the naïve gradient estimator may be reduced using methods such as control variates (Paisley et al., 2012).

An important consideration is that the RPROP gradient descent algorithm we use to optimize kernel parameters (detailed in Algorithm 2.4.1) is extremely well suited for instances where the gradient is noisy due to numerical estimation (Igel and Hüsken, 2003).

5.3 Chronic Stress Data

Many neuropsychiatric disorders are triggered by or are directly related to psychological stress. In this application, we look at murine models of social defeat stress in
order to identify neural circuits that are involved with stress (Kumar et al., 2014). The phenotype of chronically stressed mice is of interest.

A total of 44 mice underwent electrode implantation surgery to record local field potentials from seven distinct brain regions. Specifically, the basal lateral amygdala (BLA), central amygdala (Ce), infralimbic cortex (IL), nucleus accumbens (NAc), prelimbic cortex (PrL), ventral subiculum (VSub), and ventral tegmental area (VTA).

Each of these mice was recorded for 10 minutes in its homecage, and then underwent a forced interaction task (FIT). Within the forced interaction task, the mouse was paced in a wire cage within a chamber, and electrophysiological activity was
recorded for 5-minutes. After this period, a CD1 aggressor was introduced to the chamber, and neurophysiological data was recorded for another 5 minutes.

The original group of animals were then split into a stress group and a control group. The stress group was subjected to 15 days of chronic social defeat stress. During this time period, each mouse was introduced to a novel aggressive CD1 mouse every day, forced to interact in the same cage for 5 minutes, and the same aggressive mouse was housed in an adjacent cage for 24 hours.

Following chronic stress, the stress group was introduced to a social interaction test, in which mice were recorded in a novel chamber for 2.5 minutes. After this time, a novel CD1 aggressor was placed in a small cage within the chamber, and the mouse was recorded for another period of time. During these recordings, the total amount of time the mouse interacted close to the small cage was documented. An interaction ratio was calculated according to (interaction time, CD1 present)/(interaction time, CD1 absent). The 50% of mice with the largest interaction ratios are here termed the resilient group, while those with the smaller interaction ratios are termed the non-resilient group.

Finally, both the control group and the stress group were subjected to another round of the forced interaction task detailed above.

The CSM factor model was setup to distinguish between the stress group and the control group, between the resilient group and the non-resilient group, and between different stages of the forced interaction task. Specifically, 25 factors were used where only 6 of those factors were used in the max-margin discrimination tasks. All other settings were set as previously discussed in this work. The choice of 25 factors came from leave-one-animal-out cross-validation, where if 30 or more factors were used, factors started to become specialized towards single animals, rather than generically shared neural circuits. Because four different classification tasks were imposed on the system, the use of 6 factors for the discrimination tasks allows for networks to
be shared between the tasks as well as some flexibility in factor usage. This choice is justifiable because not all 6 factors wind up playing a large role in any of the classification tasks.

The model was trained using 75% of all 5-second windows. We treat the windows as independent, using the 25% of held-out windows on test. During testing, the factor scores for each window in the factor analysis model were inverted, keeping all the kernel parameters fixed from training. After inversion, the factor scores were fed into the trained classification tasks.

Figure 5.2 illustrates the accuracy of the classification tasks on the held out data. The inverted, held-out SVM scores for each animal were averaged across all time windows associated with each task. In all cases, the trained classifiers are able to distinguish between the conditions \( p < 0.001 \). Figure 5.3 illustrates how the resiliency classification scores correlate with the recorded interaction ratios for the

![Interaction Ratio vs. Classification Score](image)

**Figure 5.3:** Regression of interaction ratios versus average heldout classification scores for resiliency classifier. The green regions represent zones of correct prediction. A single mouse was left out because it was an outlier; however, this mouse received the correct classification.
The bar plots in Figure 5.4 depict the relative usage of each of the 6 predictive factors in all 4 of the trained prediction tasks. It is clear that information is shared between these tasks. In particular, factor 4 is a phenotype of both the stress group and the non-resilient group. Since these tasks both relate to stress, this result is expected. Similarly, in predicting the FIT and FIT+CD1 task conditions, factors 3 and 6 are used to positively indicate the FIT+CD1 condition and, therefore, negatively indicate the FIT condition. The cross-spectra most associated with these predictions are shown in Figure 5.5. While factors 2 and 4 implicate whole-brain low frequency signatures as a phenotype of stress and non-resiliency, factors 3 and 6 illustrate that more interesting spectral activity is indicative of the presence of a CD1 aggressor.
Figure 5.5: Cross-spectra of selected factors. Factors 2 and 4 are used in the prediction of resiliency to stress, while factors 3 and 6 are used to predict whether the animal is under the FIT+CD1 task condition.
Conclusions and Future Work

This work has introduced the cross-spectral mixture kernel, and made the argument that the CSM kernel should be a consideration for time-series applications where multiple correlated, quasi-periodic channels are observed. The CSM kernels has the ability to encode cross-spectral information about multi-channel data, producing a convenient, smoothed statistical summary of power and phase synchrony across all frequency bands. The CSM kernel complements the extrapolation capabilities of the spectral mixture kernel in the multi-output setting. A fast inference technique is presented, allowing for feasibility of implementation for many types of quasi-periodic, multi-channel observations. Applications to electrophysiological data have shown the ability of the CSM kernel to extract meaningful information neural circuits, and these circuits may be trained to be predictive of environmental phenotypes, such as social defeat stress.

There are several future directions applicable to this work. Extensions to applications with higher-dimensional input-spaces would be interesting to examine. Particularly, given higher resolution spatial data, the spatio-temporal applications considered in this work could be constructed with Kronecker structured covariance
kernels (Saatçi, 2011) instead of the proposed linear models of coregionalization. Another application with a higher-dimensional input-space is climate data. Importantly, continual developments on inference techniques for Gaussian processes are necessary in order to make these methods widely available.

With respect to the max-margin classification models in Section 5.2.1, one potential future direction extends this work to an ensemble of max-margin classification models and allow the loss function and decision rule to average over the ensemble mixture (Jaakkola et al., 1999; Zhu et al., 2008). This is known as maximum entropy discrimination. In order to obtain a posterior distribution on factor scores, the use of Monte Carlo variational inference techniques would provide a possible solution to obtaining noisy gradient steps on the variational parameters (Kingma and Welling, 2014).

Regarding analysis of brain states, the latent factor model provides a significant practical improvement over the former implementation of a discrete brain states model. While inversion of new data in a test set is still possible with this model, the complexity scales $O(NWC^3)$, which turns out to be very slow. A potential solution is the use of variational auto-encoders (Kingma and Welling, 2014), where an ‘encoder’ allows for a discriminative mapping from a new observation to feature space. This has the convenient solution of avoiding the inversion of the ‘decoder’ by approximating it with a non-linear function.

There are many more applications associated with local field potential, electroencephalogram, and electrocorticography recordings that may benefit from these proposed methods, let alone observations aside from neural recordings.
Appendix A
Kernel Derivations

In this section, we detail derivations of the SG, SM, and CSM kernels by the method of the auto-covariance of periodic functions with unknown frequency.

A.1 The Spectral Gaussian (SG) Kernel

Consider a periodic signal \( f(x) = \sqrt{2} \cos(\omega(x + \phi)) \) with the prior \( \omega \sim \mathcal{N}(\mu, \nu) \) on the angular frequency. Letting \( \theta = \{\mu, \nu\} \), the stationary, positive definite auto-covariance function for \( f(x) \) is derived according to

\[
k_{SG}(\tau; \theta) = \text{cov}(f(x), f(x + \tau))
\]

\[
= \mathbb{E} [2 \cos(\omega(x + \phi)) \cos(\omega(x + \phi + \tau))] \\
= \mathbb{E} [\cos(\omega \tau) - \cos(\omega(2x + 2\phi + \tau))] \\
= \mathbb{E}_{p(\omega)} [\cos(\omega \tau)]
\]

(A.1)

where the expectation can be written as an integral in phasor notation,

\[
k_{SG}(\tau; \theta) = \int_{-\infty}^{\infty} \cos(\omega \tau) \frac{1}{\sqrt{2\pi\nu}} \exp \left( -\frac{1}{2\nu}(\omega - \mu)^2 \right) d\omega
\]

\[
= \frac{1}{\sqrt{2\pi\nu}} \text{Re} \left\{ \int_{-\infty}^{\infty} \exp \left( j\omega \tau - \frac{1}{2\nu}(\omega - \mu)^2 \right) d\omega \right\}
\]

(A.2)
Letting $a = \frac{1}{2\nu}$, $b = -\frac{1}{2}(j\tau + \frac{q}{p})$, and $c = -\frac{\nu^2}{2\nu}$, then

$$k_{SG}(\tau; \theta) = \frac{1}{\sqrt{2\pi\nu}} \text{Re} \left\{ \int_{-\infty}^{\infty} \exp(-a\omega^2 - 2b\omega + c) d\omega \right\}$$

$$= \frac{1}{\sqrt{2\pi\nu}} \text{Re} \left\{ \sqrt{\frac{\pi}{a}} \exp \left( \frac{b^2}{a} + c \right) \right\}$$

$$= \text{Re} \left\{ \exp(j\mu\tau - \frac{1}{2}\nu\tau^2) \right\}$$

$$= \exp(-\frac{1}{2}\nu\tau^2) \cos(\mu\tau). \quad (A.3)$$

This results in an interpretable auto-covariance function, where $\mu$ represents the primary frequency and the variance $\nu$ controls how quickly this frequency dependency decays with $\tau$. Used as a stationary kernel in a Gaussian process, the SG kernel discovers a single Gaussian component in the spectral density.

A.2 The Spectral Mixture (SM) Kernel

Consider a periodic signal $f(x) = \sum_{q=1}^{Q} \sqrt{2a_q} \cos(\omega_q^T(x + \phi_q))$, where $Q$ latent functions are superimposed, each with unique input-space offset $\phi_q \in \mathbb{R}^P$ and angular frequency vector $\omega_q \in \mathbb{R}^P$. Each angular frequency component is assigned the prior $\omega_q^{(p)} \sim \mathcal{N}(\mu_q^{(p)}, \nu_q^{(p)})$. Letting $\theta = (\mu_q, \nu_q)^{Q}_{q=1}$ and $p(\Omega) = \prod_q \prod_p p(\omega_q^{(p)})$, the auto-covariance function for $f(x)$ is derived according to

$$k_{SM}(\tau; \theta) = \text{cov}(f(x), f(x + \tau))$$

$$= \mathbb{E} \left[ 2 \sum_{q=1}^{Q} \sum_{r=1}^{Q} \sqrt{a_q a_r} \cos(\omega_q^T(x + \phi_q)) \cos(\omega_r^T(x + \phi_r + \tau)) \right]$$

$$= \sum_{q=1}^{Q} a_q \mathbb{E}_{p(\Omega)} \left[ \cos(\omega_q^T \tau) \right],$$

68
where the expectation over the input domain eliminates cross terms between all \( q \neq r \) since each \( \omega_q^{(p)} \pm \omega_r^{(p)} \) with probability 1. Proceeding in the same fashion as the SG kernel, \( k_{SM}(\tau; \theta) = \sum_{q=1}^{Q} a_q \int_{-\infty}^{\infty} \cos(\omega^T \tau) \prod_{p=1}^{P} \frac{1}{\sqrt{2 \pi \nu_q^{(p)}}} \exp \left( -\frac{1}{2 \nu_q^{(p)}} (\omega_q^{(p)} - \mu_q^{(p)})^2 \right) d\omega \)  

\[
= \sum_{q=1}^{Q} a_q \prod_{p=1}^{P} \frac{1}{\sqrt{2 \pi \nu_q^{(p)}}} \text{Re} \left\{ \int_{-\infty}^{\infty} \exp \left( j \omega_q^{(p)} \tau_p - \frac{1}{2 \nu_q^{(p)}} (\omega_q^{(p)} - \mu_q^{(p)})^2 \right) d\omega_q^{(p)} \right\} 
\]

\[
= \sum_{q=1}^{Q} a_q \prod_{p=1}^{P} \exp \left( -\frac{1}{2 \nu_q^{(p)}} \tau_p^2 \right) \cos(\mu_q^{(p)} \tau_p) \quad \text{(A.4)}
\]

where the final step proceeds by recognizing that each integral in the product is an SG component from (A.2). This is known as the spectral mixture (SM) kernel.

### A.3 The Cross-Spectral Mixture (CSM) Kernel

We now consider \( C \) different channels and allow observations from each channel \( \{f_c(x)\}_{c=1}^{C} \) to be represented as a linear combination of latent signals, \( f_c(x) = \sum_{q=1}^{Q} \sqrt{2a_{cq}} \cos(\omega_q^T (x + \phi_{cq})) \). While the angular frequency components are still assigned the prior \( \omega_q^{(p)} \sim \mathcal{N}(\mu_q^{(p)}, \nu_q^{(p)}) \), each channel is assigned channel-specific input-space shifts \( \phi_{cq} \). When computing the cross-covariance function, these shifts will not cancel when \( c_1 \neq c_2 \). The cross-covariance function is derived according to  

\[
k_{CSM}^{c_1,c_2}(\tau; \theta) = \text{cov}(f_{c_1}(x), f_{c_2}(x + \tau)) 
\]

\[
= E \left[ 2 \sum_{q=1}^{Q} \sum_{r=1}^{Q} \sqrt{a_{c_1 q} a_{c_2 r}} \cos(\omega_q^T (x + \phi_{c_1 q})) \cos(\omega_r^T (x + \phi_{c_2 r} + \tau)) \right] 
\]

\[
= \sum_{q=1}^{Q} \sqrt{a_{c_1 q} a_{c_2 q}} \mathbb{E}_p(\Omega) \left[ \cos(\omega_q^T \tau + \phi_{c_2 q} - \phi_{c_1 q}) \right] \quad \text{(A.5)}
\]
where again proceeding with expanding the expectation as an integral over phasor notation,

\[ k_{\text{CSM}}^{c_1,c_2}(\tau; \theta) = \sum_{q=1}^{Q} \sqrt{a_{c_1,q} a_{c_2,q}} \prod_{p=1}^{P} \frac{1}{\sqrt{2\pi \nu_q^{(p)}}} \Re \left\{ \int_{-\infty}^{\infty} \exp \left( j\omega_q^{(p)} \tau_p + j(\phi_{c_2,q}^{(p)} - \phi_{c_1,q}^{(p)}) \right) \right. \\
- \frac{1}{2\nu_q^{(p)}} \left( \omega_q^{(p)} - \mu_q^{(p)} \right)^2 \left. \right\} d\omega_q^{(p)} \right\} \\
= \sum_{q=1}^{Q} \sqrt{a_{c_1,q} a_{c_2,q}} \prod_{p=1}^{P} \exp \left( -\frac{1}{2} \nu_q^{(p)} \tau_p^2 \right) \cos \left( \mu_q^{(p)} (\tau_p + \phi_{c_2,q}^{(p)} - \phi_{c_1,q}^{(p)}) \right), \quad (A.6) \]

where the final step proceeds by recognizing that each integral in the product is simply an SG component from (A.2). The phase difference term \( \phi_{c_2,q}^{(p)} - \phi_{c_1,q}^{(p)} \) in this last step simply gets added to the constant \( c \) to produce the final result. We have termed this novel kernel the cross-spectral mixture (CSM) kernel.
Appendix B

Infinite Tensor Mixtures

The proposed model in Section 4.1.1 has similar properties to the infinite tensor mixture (ITM) model in (Murray and Dunson, 2013), which uses a nonnegative PARAFAC tensor decomposition (Harshman, 1970), when observing the stationary distribution over brain states. Though similar in many ways, the PARAFAC decomposition is fundamentally different than the higher order SVD (HOSVD) decomposition (Lathauwer et al., 2000), which is used in the simplex factor model of (Bhattacharya and Dunson, 2012). To reformulate the model in Section 4.1.1 as a nonnegative PARAFAC decomposition, we follow (Murray and Dunson, 2013) and define probabilities over clusters \( \pi^{(aw)} \) according to an infinite tensor factorization (ITF)

\[
\text{pr}(z^{(a)}_w = \ell_1, \ldots, z^{(a)}_w = \ell_R) = \pi^{(aw)}_{\ell_1, \ldots, \ell_R},
\]

\[
\pi^{(aw)} = \sum_{h=1}^{\infty} \text{pr}(s^{(a)}_w = h) \otimes_{r=1}^{R} \Phi_h^{(r)},
\]

where \( \otimes \) represents the tensor product. Murray and Dunson (2013) draws \( \text{pr}(s^{(a)}_w) \) as a GEM distribution, whereas, in this work, \( \text{pr}(s^{(a)}_w) \) is represented by the stationary distribution of the iHMM for animal \( a \). In this infinite tensor reformulation, each
brain state is represented as a rank-1, $R$-way probability tensor. Given brain state $h$, the distribution over cluster assignments for region $r$ is given by the probability vector $\phi_h^{(r)}$. In our model, the prior distributions on $\phi_h^{(r)}$ are still equivalent to their definitions in the paper, whereas Murray and Dunson (2013) impose stick-breaking processes. The resulting probability tensor $\pi^{(aw)}$ represents the stationary joint distribution of cluster assignments for all regions.

To conclude, this clustering model differs from this ITF model of Murray and Dunson (2013) in three significant ways: we place Markovian dynamics on state assignments for each animal, we model separate draws from the prior jointly for each animal, and we share cluster atoms across all regions through use of an HDP. These are significant differences that change the dynamics and interpretation of our model. For example, a primary difference is that, in our model, two regions assigned to the same index cluster are drawn $iid$ from the same atom in the mixture model.
Appendix C

Bayesian Hidden Markov Models

C.1 The hidden Markov model

A hidden Markov model is comprised of a sequence of observations \( \{y_t\}_{t=1}^T \), where each observation is generated from an emission distribution \( p(y_t|x_t) \). Each hidden variable \( x_t \in \{1, \ldots, K\} \) is assigned to one of \( K \) discrete states, and the state sequence \( \{x_t\}_{t=1}^T \) follows a Markov-chain, i.e., \( p(x_{1:T}) = p(x_1) \prod_{t=2}^T p(x_t|x_{t-1}) \). This process is described by

\[
y_t \sim f(\theta_{x_t}), \quad x_1 \sim \text{Categorical}(\pi), \quad x_t \sim \text{Categorical}(a_{x_{t-1}}) \quad \forall \ t \geq 2,
\]

(C.1)

where \( \theta_i \) are parameters for the emission distribution \( f \) associated with state \( i \), and the initial state probability vector and transition matrix are defined as follows:

\[
\pi_i \triangleq p(x_1 = i), \quad A \triangleq \begin{pmatrix}
a_{11} & \cdots & a_{1K} \\
\vdots & \ddots & \vdots \\
a_{K1} & \cdots & a_{KK}
\end{pmatrix}.
\]

The parameters \( \Theta = \{\theta_i\} \), \( A \) and \( \pi \) are model parameters to be optimized. In this section, the maximum likelihood updates are provided. In Section C.2, prior distributions are placed on the parameters, allowing for inference of the posterior distribution.
Inference for the HMM is typically performed through an expectation-maximization (EM) algorithm. The E-step obtains expected marginal statistics of state assignments. As shorthand, we say that the E-step computes the necessary statistics of the marginal posterior distribution \( p(x_{1:T}|y_{1:T}, \Theta) = \text{HMM}(A, \pi, L) \), where \( \{A, \pi, \Theta\} \) are as defined above, and \( L \in \mathbb{R}^{K \times T} \) holds the data likelihood potentials. For notation purposes, we define \( \ell_t(i) \equiv p(y_t|\theta_i) \), and the \( t \)-th column of \( L \) is given by \( \ell_t \).

The M-step then maximizes the model parameters \( A, \pi \) and \( \Theta \) given these expected marginal statistics.

### C.1.1 E-step

In order to perform inference in this model, the joint posterior density \( p(x_{1:T}|y_{1:T}, \Theta) = \text{HMM}(A, \pi, L) \) is desired. However, only the following marginal statistics of this density are required for the updates of \( \Theta, \pi \) and \( A \):

\[
\gamma_t(i) \equiv p(x_t = i|y_{1:T}, \Theta), \quad \xi_t(i, j) \equiv p(x_t = i, x_{t+1} = j|y_{1:T}, \Theta) \quad (C.2)
\]

These statistics are often computed through the forwards-backward algorithm. During forwards propagation, the filtered marginals, \( p(x_t|y_{1:t}) \), are computed. Here, state assignments are only conditioned on previous data; the forwards-backward algorithm smoothes the state sequence by conditioning on future observations, i.e., \( p(x_t|y_{1:T}) \) is computed. This is done by separating the smoothed marginals into two parts:

\[
p(x_t = i|y_{1:T}) \propto p(x_t = i|y_{1:t}) p(y_{t+1:T}|x_t = i) \propto \alpha_t(i) \beta_t(i) \quad (C.3)
\]
and computing these two parts iteratively through a forwards propagation, $\alpha_t$, and a backwards propagation, $\beta_t$. The forwards propagation is computed as

$$\alpha_t(j) = p(x_t = j|y_{1:t})$$

$$= \frac{1}{\zeta_t} p(y_t|x_t = j)p(x_t = j|y_{1:t-1})$$

$$= \frac{1}{\zeta_t} \ell_t(j) \sum_{i=1}^{K} a_{ij} \alpha_{t-1}(i)$$

$$\alpha_t \preceq \ell_t \odot (A^T \alpha_{t-1}) \quad (C.4)$$

where $\odot$ represents the Hadamard (element-wise) product, and the propagation is initialized with $\alpha_1 \preceq \ell_1 \odot \pi$. During the forwards propagation, the normalizing constant $\zeta_t$ is useful to store during computation because $\zeta_t$ represents the predictive likelihood of the observation $y_t$ given all previous data in the time-series:

$$\zeta_t = p(y_t|y_{1:t-1}) = \sum_{i=1}^{K} p(y_t|x_t = i)p(x_t = i|y_{1:t-1}). \quad (C.5)$$

Storing these values allows for an inexpensive computation of the data likelihood through the chain equation

$$p(y_{1:T}|\Theta) = \prod_{t=1}^{T} p(y_t|y_{1:t-1}) = \prod_{t=1}^{T} \zeta_t. \quad (C.6)$$

This normalization constant will be useful when computing the model evidence or the variational evidence lower bound.
The backwards propagation is computed as
\[
\beta_{t-1}(i) = p(y_{t:T}|x_{t-1} = i)
\]
\[
= \sum_{j=1}^{K} p(y_{t+1:T}|x_t = j)p(x_t = j, y_t|x_{t-1} = i)
\]
\[
= \sum_{j=1}^{K} \beta_t(j)\ell_t(j)a_{ij}
\]
\[
\beta_{t-1} \propto A(\ell_t \odot \beta_t)
\]  
(C.7)

where the propagation is initialized with $\beta_T(i) = 1$.

It is important to note that both the forwards and backwards propagation steps have complexity $O(K^2T)$ without any further assumptions imposed on the updates in the forwards-backwards algorithm. For small $K$, this complexity is reasonable; for large $K$, one may desire a sparse transition matrix to speed up computation.

To avoid numerical underflow during backwards propagation, each $\beta_t$ may be normalized without an effect on the performance of the algorithm. The updates in (C.4), (C.7) and (C.6) may also be computed in log-space to avoid numerical underflow if necessary.

With the updates in (C.4) and (C.7), the marginal statistics from (C.2) are simply computed by
\[
\gamma_t(i) \propto \alpha_t(i)\beta_t(i) \quad \quad \quad \gamma_t \propto \alpha_t \odot \beta_t
\]  
(C.8)
\[
\xi_{t,t+1}(i,j) \propto \alpha_t(i)a_{ij}\ell_{t+1}(j)\beta_{t+1}(j) \quad \quad \quad \xi_{t,t+1} \propto A \odot (\alpha_t(\ell_{t+1} \odot \beta_{t+1})^T
\]  
(C.9)

where we show element-wise updates on the left and matrix form on the right. Equation (C.8) is known as the *smoothed* marginal distribution, representing the expected probability of time $t$ belonging to state $i$. Equation (C.9) is known as the *two-slice* smoothed marginal distribution, representing the expected probability of a transition from state $i$ to state $j$ at time $t$. These marginal distributions are used in the M-step of the EM algorithm for HMMs.
C.1.2 M-step

The maximum likelihood estimate for $A$ is given by

$$
\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{t,t+1}(i,j)}{\sum_{t=1}^{T-1} \sum_{j'=1}^{K} \xi_{t,t+1}(i,j')}
$$

which can be thought of as a sum of the expected number of transitions from state $i$ to state $j$, normalized by the total expected transitions out of state $i$.

The maximum likelihood estimate for $\pi$ is

$$
\hat{\pi}_i = \gamma_1(i)
$$

by definition.

Updating the parameters $\Theta$ depends on the emission distribution $f$. Assuming $f$ is a Gaussian distribution then $\theta_i = \{\mu_i, \Sigma_i\}$ and the maximum likelihood estimates of these parameters are

$$
\hat{\mu}_i = \frac{1}{N_i} \sum_{t=1}^{T} \gamma_t(i) y_t, \quad \hat{\Sigma}_i = \frac{1}{N_i} \sum_{t=1}^{T} \gamma_t(i) y_t^T y_t^T - \hat{\mu}_i \hat{\mu}_i^T
$$

where $N_i = \sum_{t=1}^{T} \gamma_t(i)$.

C.2 The Bayesian hidden Markov model

The generative process for a Bayesian hidden Markov model (HMM) extends (C.1) into

$$
y_t \sim f(\theta_{x_t}), \quad \theta_i \sim H, \\
x_1 \sim \text{Categorical}(\pi), \quad x_t \sim \text{Categorical}(a_{x_{t-1}}) \quad \forall \ t \geq 2, \\
\pi \sim \text{Dirichlet}(\kappa), \quad a_i \sim \text{Dirichlet}(\nu),
$$

where $\kappa$ and $\nu$ are hyperparameters. These parameters are updated using the Bayes rule and the prior distribution specified earlier.
by placing a prior distribution over the parameters $\Theta$, $\pi$ and $A$. We note the Dirichlet distribution is a conjugate prior to the Categorical distribution, and the prior $H$ is chosen to be conjugate to the emission distribution $f$.

In this document, inference is performed by seeking a mean-field variational posterior distribution $q$ that is close to the true posterior distribution $p$. The variational distribution is assumed factorizable: $q(x_{1:T}, \pi, A, \Theta) = q(x_{1:T})q(\pi)\prod_i q(a_i)q(\theta_i)$

C.2.1 VB E-step

With mean-field variational inference, the expected sufficient statistics of the state sequence are obtained through $q(x_{1:T}) = \text{HMM}(\tilde{A}, \tilde{\pi}, \tilde{L})$ where

$$\tilde{a}_{ij} \triangleq \exp\{E_{q(a_i)}[\ln a_{ij}]\},$$
$$\tilde{\pi}_i \triangleq \exp\{E_{q(\pi)}[\ln \pi_i]\},$$
$$\tilde{\ell}_i(i) \triangleq \exp\{E_{q(\theta_i)}[\ln p(y_t|\theta_i)]\}.$$ (C.14)

These sub-normalized probabilities are treated as input into the standard HMM. For $A$ and $\pi$, these inputs are the geometric means of the posterior distribution. For $L$, this is the exponentiation of the expected log likelihoods.

The normalizing constants from (C.5) should still be stored throughout this procedure since the data likelihood captured in (C.6) is important during the computation of the evidence lower bound.

C.2.2 VB M-step

Update transition matrix

The variational posterior distribution for each transition vector is

$$q(a_i) = \text{Dirichlet}(\hat{\alpha}^a_i)$$ (C.15)

$$\hat{\alpha}^a_{ij} = v_j + \sum_{t=1}^{T-1} \xi_{t,t+1}(i, j),$$ (C.16)
where the E-step requires the expectation \( E[\ln a_{ij}] = \psi(\hat{\alpha}_{i}^{\pi}) - \psi(\sum_{j'=1}^{K} \hat{\alpha}_{j'}^{\pi}) \) and \( \psi(\cdot) \) is the digamma function.

**Update initial distribution**

The variational posterior of the initial distribution is

\[
q(\pi) = \text{Dirichlet}(\hat{\alpha}^{\pi}) \quad \text{(C.17)}
\]

\[
\hat{\alpha}_{i}^{\pi} = \kappa_{i} + \gamma_{i}(i), \quad \text{(C.18)}
\]

where the E-step requires the expectation \( E[\ln \pi_{i}] = \psi(\hat{\alpha}_{i}^{\pi}) - \psi(\sum_{j'=1}^{K} \hat{\alpha}_{j'}^{\pi}) \) and \( \psi(\cdot) \) is the digamma function.

**Update emission distribution**

Updates for the emission distribution vary according to the distribution chosen in the modeling framework.

**C.2.3 Variational lower bound**

The variational lower bound is computed as follows:

\[
\mathcal{L} = \mathbb{E}_{q(\pi)} \left[ \ln \frac{p(\pi)}{q(\pi)} \right] + \sum_{i=1}^{K} \mathbb{E}_{q(\alpha_{i})} \left[ \ln \frac{p(\alpha_{i})}{q(\alpha_{i})} \right] + \sum_{i=1}^{K} \mathbb{E}_{q(\theta_{i})} \left[ \ln \frac{p(\theta_{i})}{q(\theta_{i})} \right] + \sum_{t=1}^{T} \log \zeta_{t} \quad \text{(C.19)}
\]
Appendix D

Bayesian hidden semi-Markov models

A primary problem with hidden Markov models is the inherent Geometric distribution placed on the duration of a state. To see this, we look at the prior probability that a recently-entered hidden state $i$ will persist for a duration $d$; this quantity is given by

$$p(x_{t+1} = i, x_{k+d} \neq i | x_t = i, x_{t-1} \neq i) \propto \rho_i(d) = (a_{ii})^{d-1}(1 - a_{ii}), \quad (D.1)$$

where $a_{ii}$ is the probability of a self-transition. We can see that

$$\rho_i(d) = \text{Geometric}(d; a_{ii}). \quad (D.2)$$

If we have prior knowledge that states persist for a variable duration, it is useful to represent $\rho_i(d)$ by a more appropriate distribution (e.g., Poisson, negative binomial, etc.).

A hidden semi-Markov model (HSMM) allows for an explicit definition of a duration distribution $\rho_i(d) \triangleq \rho(d; \omega_i)$, parametrized by $\omega_i$ (Yu, 2010). By setting $\rho(\omega_i)$ to a Geometric distribution, an HMM is recovered; however, the HSMM framework allows for the ability to define a variety of parametric duration distributions, often
dependent on the application. In the generative process for a Bayesian HSMM, if state $i$ is entered at time $t$, a *residential time* parameter is drawn $\tau_t \sim \rho(\omega_i)$, representing the time remaining in the current state. For each consecutive observation, the residential time deterministically counts down (i.e., $\tau_{t+1} = \tau_t - 1$). Once $\tau_t = 1$ for some $t' \geq t$, the model transitions into a new state $j \neq i$, and the duration parameter is redrawn as $\tau_{t'+1} \sim \rho(\omega_j)$. This generative process is formalized as follows:

$$y_t \sim f(\theta_{x_t})$$

$$p(x_{t+1} = j|x_t = i, \tau_t = d, t \geq 1) = \begin{cases} a_{ij} : d = 1, \ i \neq j \\ \mathbb{1}_{(i=j)} : d > 1 \\ 0 : \text{otherwise} \end{cases}$$

$$p(\tau_{t+1} = d'|x_{t+1} = j, \tau_t = d, t \geq 1) = \begin{cases} \rho(d'; \omega_j) : d = 1 \\ \mathbb{1}_{(d'=d-1)} : d > 1 \\ 0 : \text{otherwise} \end{cases}$$

$$(x_1, \tau_1) \sim \pi$$

$$(\theta_i, \omega_i, \pi) \sim H \times G \times E$$

$$a_{i\setminus i} \sim \text{Dirichlet}(\nu), \ a_{ii} = 0 \quad (D.3)$$

where observations $\{y_t\}_{t=1}^T$ are generated from an emission distribution $f$, parametrized by $\theta_{x_t}$. Each $\theta_i$ is drawn independently from the distribution $H$. A hidden state sequence with semi-Markovian dependencies is represented by $\{x_t\}_{t=1}^T$. The probability of transition from $x_t = i$ to $x_{t+1} = j$ is represented by $a_{ij}$. Since the HSMM does not permit self-transitions, $a_{ii}$ is set to 0, and we adopt the notation $a_{i\setminus i} = \{a_{i,1}, \ldots, a_{i,i-1}, a_{i,i+1}, \ldots, a_{i,K}\}$ in order to place a Dirichlet prior on the transition vector from state $i$ to any state excluding state $i$. This transition is only permitted when the residential time has run out (i.e., $\tau_t = 1$). When this transition occurs, the residential time is reset as a draw from the duration distribution $\rho(\omega_j)$, where each set of parameters $\omega_i$ are independent draws from $G$. The initial state and residential time ($x_1$ and $\tau_1$, respectively) are drawn jointly from $\pi$, which has a
prior $E$.

D.1 Embedding the HSMM in a stationary HMM

The hidden semi-Markov model may be embedded within a stationary HMM. This embedding models a Markov chain on the joint process $(x_t, \tau_t)$. A state in this Markov chain is uniquely defined by the state assignment in the semi-Markov chain $x_t$ jointly with the residual time of this semi-Markov state $\tau_t$.

As such, a transition kernel of this Markovian process is defined by an injective mapping from the semi-Markovian process,

$$
\phi_{(i,d)(j,d')} = \begin{cases} 
1 & : j = i, \ d > 1, \ d' = d - 1 \\
\alpha_{ij} \rho_j(d') & : d = 1 \\
0 & : \text{otherwise}
\end{cases}
$$

(D.4)

where $\phi_{(i,d)(j,d')}$ is the probability of transitioning from state $i$ with residual time $d$ to state $j$ with residual time $d'$. Furthermore, the initial condition implies $\pi_{(i,d)} = p((x_1, \tau_1) = (i, d))$, data likelihood potentials are still represented as $\ell_t(i) = p(y_t | \theta_i)$, and a matrix of residual time potentials $R \in \mathbb{R}^{K\times D}$ is constructed such that $r_{id} = \rho_i(d)$. In this formulation, the state duration is truncated at $D$; this truncation allows for computational improvements, but care should be taken to set $D$ large enough to capture the true residual time distribution of a state. Thus, the embedded stationary HMM has a marginal distribution $p((x_t, \tau_t)_{t=1}^T | y_{1:T}, \pi, \Theta) = \text{HMM}(\Phi, \pi, L) = \text{HSMM}(A, R, \pi, L)$. The same EM algorithm used for the HMM may be used to obtain the expected sufficient statistics of this injective map.

D.2 Variational inference

With mean-field variational inference, an approximate posterior distribution $q$ is sought that is close to the true posterior $p$. In the E-step of this algorithm, expected statistics from the marginal distribution $p((x_t, \tau_t)_{t=1}^T | y_{1:T}, \pi, \Theta) = \text{HSMM}(A, R, \pi, L)$
are obtained, where the parameters with a tilde indicate the geometric mean of their marginal posteriors:

$$
\tilde{a}_{ij} \triangleq \exp\{\mathbb{E}_{q(a_i)}[\ln a_{ij}]\}, \quad \tilde{r}_{id} \triangleq \exp\{\mathbb{E}_{q(\omega_i)}[\ln \rho(d|\omega_i)]\}, \\
\tilde{\pi}_i \triangleq \exp\{\mathbb{E}_{q(\pi)}[\ln \pi_i]\}, \quad \tilde{\ell}_t(i) \triangleq \exp\{\mathbb{E}_{q(\theta_i)}[\ln p(y_t|\theta_i)]\}. \quad (D.5)
$$

Section D.2.1 presents the E-step updates for HSMM($A, R, \pi, L$). With mean-field inference, the only change will be using the sub-normalized probabilities from (D.5) as inputs into the algorithm. The variational M-step presented in Section D.2.2 optimizes the distributions $\{q(a_i)\}$, $q(\pi)$, $\{q(\theta_i)\}$ and $\{q(\omega_i)\}$ given $q((x_t, \tau_t)_{t=1}^T)$. The E-step and M-step are iteratively computed until convergence of the variational lower bound in Section D.2.3.

### D.2.1 E-step

Although we have just mentioned that the HMM algorithm may be used on the injective map to perform E-step updates for an HSMM, these computations will have complexity $\mathcal{O}(K^2D^2T)$, which may be very expensive if $D$ is on the order of magnitude of $T$. Fortunately, the transition kernel of the injective map is primarily composed of zeros, allowing for some adjustments to be made that reduces this complexity to $\mathcal{O}(K^2T + KDT)$ (Yu and Kobayashi, 2003; Hudson, 2009). In the following derivations, any occurrence of $\phi_{i,d}(j,d')$ is replaced by

$$
[a_{ij}\rho_j(d')\mathbb{I}_{(d=1, j\neq i)} + \mathbb{I}_{(j=i, d>1, d'=d-1)}]
$$

and simplified. Equations (D.6)–(D.12) are computed during the update procedure HSMM($A, R, \pi, L$) in order to obtain the necessary expected marginal statistics of the posterior distribution.
Forwards propagation

These updates are computed according to
\[
\alpha_t(j, d') \triangleq p(x_t = j, \tau_t = d' | y_{1:t}) = \frac{1}{\zeta_t} p(y_t | x_t = j) p(x_t = j, \tau_t = d' | y_{1:t-1}) = \frac{1}{\zeta_t} \ell_t(j) \sum_{i=1}^{K} \sum_{d=1}^{D} \phi(i, d)(j, d') \alpha_{t-1}(i, d) = \frac{1}{\zeta_t} \ell_t(j) \left[ \sum_{i=1}^{K} \alpha_{t-1}(i, 1) a_{ij} + \alpha_{t-1}(j, d' + 1) \right]
\] (D.6)

where we define \( \alpha_t(i, D + 1) = \alpha_t(i, D) \) and continue to store the important normalization constant:
\[
\zeta_t \triangleq p(y_t | y_{1:t-1}) = \sum_{j=1}^{K} \sum_{d'=1}^{D} \ell_t(j) \left[ \sum_{i=1}^{K} \alpha_{t-1}(i, 1) a_{ij} + \alpha_{t-1}(j, d' + 1) \right]
\] (D.7)

Backwards propagation

These updates are computed according to
\[
\beta_{t-1}(i, d) \triangleq p(y_{t:T} | x_{t-1} = i, \tau_{t-1} = d) = \sum_{j=1}^{K} \sum_{d'=1}^{D} p(y_{t+1:T} | x_t = j, \tau_t = d') p(x_t = j, \tau_t = d', y_t | x_{t-1} = i, \tau_{t-1} = d) = \sum_{j=1}^{K} \sum_{d'=1}^{D} \beta_t(j, d') \ell_t(j) \phi(i, d)(j, d') = \left\{ \begin{array}{ll}
\sum_{j=1, j \neq i}^{K} a_{ij} \ell_t(j) \sum_{d'=1}^{D} \rho_j(d') \beta_t(j, d') & : d = 1 \\
\beta_t(i, d - 1) \ell_t(i) & : d > 1
\end{array} \right.
\] (D.8)
Smoothed marginal distribution

The smoothed marginal distribution is represented by
\[ \gamma_t(i, d) \doteq p(x_t = i, \tau_t = d | \mathbf{y}_{1:T}, \Theta) \]
\[ \propto \alpha_t(i, d) \beta_t(i, d) \] \hspace{1cm} (D.9)

Two-slice smoothed marginal distribution

The entire two-slice smoothed marginal distribution is given by
\[ \xi_t(i, d, j, d') \doteq p(x_t = i, \tau_t = d, x_{t+1} = j, \tau_{t+1} = d' | \mathbf{y}_{1:T}, \Theta) \]
\[ = \frac{1}{c_{\xi_t}} \alpha_t(i, d) \phi(i, d)(j, d') \ell_{t+1}(j) \beta_{t+1}(j, d') \]

where \( \frac{1}{c_{\xi_t}} \) is a normalizing constant at time \( t \). Fortunately, we do not need to store all of \( \xi_t(i, d, j, d') \), which may be prohibitive for large \( T, D \) and \( K \). Instead, the normalization constant and two marginal statistics are computed in it’s place. These two statistics are used in the M-step updates for the HSMM.

Normalization constant

\[ c_{\xi_t} \doteq \sum_{i=1}^{K} \sum_{d=1}^{D} \sum_{j=1}^{K} \sum_{d'=1}^{D} \xi_t(i, d, j, d') \]
\[ = \sum_{i=1}^{K} \sum_{j=1}^{K} \sum_{d'=1}^{D} \alpha_t(i, 1) a_{ij} \rho_j(d') \ell_{t+1}(j) \beta_{t+1}(j, d') \]
\[ + \sum_{i=1}^{K} \sum_{d=2}^{D} \alpha_t(i, d) \ell_{t+1}(i) \beta_{t+1}(i, d-1) \] \hspace{1cm} (D.10)

Two-slice smoothed marginal transition distribution

\[ \chi_t(i, j) \doteq p(x_t = i, \tau_t = 1, x_{t+1} = j | \mathbf{y}_{1:T}, \Theta) = \sum_{d'=1}^{D} \xi_t(i, 1, j, d') \]
\[ = \frac{1}{c_{\xi_t}} \alpha_t(i, 1) a_{ij} \ell_{t+1}(j) \sum_{d'=1}^{D} \rho_j(d') \beta_{t+1}(j, d') \] \hspace{1cm} (D.11)
Two-slice smoothed marginal residual time distribution

\[ \eta_t(j, d') \triangleq p(\tau_t = 1, x_{t+1} = j, \tau_{t+1} = d'|y_{1:T}, \Theta) = \sum_{i=1, i \neq j}^{K} \xi_t(i, 1, j, d') \]

\[ = \frac{1}{c_{\xi_t}} \ell_{t+1}(j)\rho_j(d')\beta_t(j, d') \sum_{i=1, i \neq j}^{K} \alpha_t(i, 1) a_{ij} \quad (D.12) \]

D.2.2 Variational M-step

Update transition matrix

In this section, we detail updates to the transition matrix. The variational posterior distribution for each transition vector is

\[ q(a_{i,j}) = \text{Dirichlet}(\hat{\alpha}_{i1}^a, \ldots, \hat{\alpha}_{i,i-1}^a, \hat{\alpha}_{i,i+1}^a, \ldots, \hat{\alpha}_{iK}^a) \quad (D.13) \]

\[ \hat{\alpha}_{ij}^a = \nu_j + \sum_{t=1}^{T-1} \chi_t(i, j), \quad \text{for } j \neq i \quad (D.14) \]

where \( a_{ij} = 0 \)

Update duration distribution

Here, the updates for various duration distributions are explored, to include a categorical distribution with a Dirichlet prior on \( \omega_i \), a Poisson distribution with a Gamma prior on \( \omega_i \), and a Geometric distribution with a Beta prior on \( \omega_i \).

Categorical duration With a categorical duration distribution, the model is specified by

\[ \rho_i(d) = \text{Categorical}(d; \omega_i), \quad \omega_i \sim G = \text{Dirichlet}(\alpha_0), \quad (D.15) \]

and updating the parameters of this duration distribution results in the variational posterior distribution

\[ q(\omega_i) = \text{Dirichlet}(\hat{\alpha}_i), \quad (D.16) \]

\[ \hat{\alpha}_{id} = \alpha_0 + \sum_{t=1}^{T} \eta_t(i, d), \quad (D.17) \]
such that the expected log likelihood of a duration $d$ (used as input to the HSMM) is

$$\mathbb{E}[\ln \rho_i(d)] = \mathbb{E}[\ln \omega_{id}] \quad (D.18)$$

where $\mathbb{E}[\ln \omega_{id}] = \psi(\hat{\alpha}_{id}) - \psi(\sum_d \hat{\alpha}_{id})$, and $\psi(\cdot)$ is the digamma function.

**Poisson duration** With a Poisson duration distribution, the model is specified by

$$\rho_i(d) = \text{Poisson}(d; \omega_i), \quad \omega_i \sim \text{Gamma}(\alpha_0, \beta_0), \quad (D.19)$$

and updating the parameters of this duration distribution results in the variational posterior distribution

$$q(\omega_i) = \text{Gamma}(\hat{\alpha}_i, \hat{\beta}_i), \quad (D.20)$$

\begin{align*}
\hat{\alpha}_i &= \alpha_0 + \frac{1}{2} \sum_{t=1}^{T} \sum_{d=1}^{D} \eta_t(i, d), \\
\hat{\beta}_i &= \beta_0 + \frac{1}{2} \sum_{d=1}^{D} d \sum_{t=1}^{T} \eta_t(i, d),
\end{align*} \quad (D.21) \quad (D.22)

such that the expected log likelihood of a duration $d$ (used as input to the HSMM) is

$$\mathbb{E}[\ln \rho_i(d)] = -\ln d! + d \mathbb{E}[\ln \omega_i] - \mathbb{E}[\omega_i] \quad (D.23)$$

where $\mathbb{E}[\omega_i] = \frac{\alpha}{\hat{\beta}_i}$, $\mathbb{E}[\ln \omega_i] = \psi(\hat{\alpha}_i) - \ln \hat{\beta}_i$, and $\psi(\cdot)$ is the digamma function.

**Geometric duration** With a geometric duration distribution, the model is specified by

$$\rho_i(d) = \text{Geometric}(d; \omega_i), \quad \omega_i \sim \text{Gamma}(\alpha_0, \beta_0), \quad (D.24)$$
and updating the parameters of this duration distribution results in the variational posterior distribution

\[ q(\omega_i) = \text{Beta}(\hat{\alpha}_i, \hat{\beta}_i), \]  

(D.25)

\[ \hat{\alpha}_i = \alpha_0 + \sum_{t=1}^{T} \sum_{d=1}^{D} \eta_t(i, d), \]  

(D.26)

\[ \hat{\beta}_i = \beta_0 + \sum_{d=1}^{D} \sum_{t=1}^{T} \eta_t(i, d), \]  

(D.27)

such that the expected log likelihood of a duration \( d \) (used as input to the HSMM) is

\[ \mathbb{E}[\ln \rho_i(d)] = (d - 1)\mathbb{E}[\ln(1 - \omega_i)] + \mathbb{E}[\ln \omega_i] \]  

(D.28)

where \( \mathbb{E}[\ln \omega_i] = \psi(\hat{\alpha}_i) - \psi(\hat{\alpha}_i + \hat{\beta}_i), \mathbb{E}[\ln(1 - \omega_i)] = \psi(\hat{\beta}_i) - \psi(\hat{\alpha}_i + \hat{\beta}_i), \) and \( \psi(\cdot) \) is the digamma function.

It is important to note that this recovers the HMM, or the sticky-HMM with a strong prior biasing \( \omega_i \) closer to 1.

\[ q(\alpha_i) = \text{Dirichlet}(\eta_i) \]  

(D.29)

### D.2.3 Variational lower bound

The variational lower bound is computed as follows:

\[ \mathcal{L} = \mathbb{E}_{q(\pi)} \left[ \ln \frac{p(\pi)}{q(\pi)} \right] + \sum_{i=1}^{K} \left( \mathbb{E}_{q(\alpha_i)} \left[ \ln \frac{p(\alpha_i)}{q(\alpha_i)} \right] + \mathbb{E}_{q(\theta_i)} \left[ \ln \frac{p(\theta_i)}{q(\theta_i)} \right] + \mathbb{E}_{q(\rho_i)} \left[ \ln \frac{p(\rho_i)}{q(\rho_i)} \right] \right) \]

\[ + \sum_{t=1}^{T} \log \zeta_t \]  

(D.30)
Appendix E

Update Derivations for Brain State Model

As listed in the paper, the variational lower bound of our model is

\[ \mathcal{L}(q) = \mathbb{E}[\ln p(Y|Z, \Theta)] + \mathbb{E}[\ln p(Z, S|\Phi, \Lambda, \Psi)] + \mathbb{E}[\ln p(\Phi|\eta)] + \mathbb{E}[\ln p(\eta)] \]

\[ + \mathbb{E}[\ln p(\Lambda|\beta)] + \mathbb{E}[\ln p(\beta)] + \mathbb{E}[\ln p(\Psi)] + \mathbb{E}[\ln p(\Theta)] \]

\[ + \mathbb{H}[q(Z)] + \mathbb{H}[q(S)] + \mathbb{H}[q(\Phi)] + \mathbb{H}[q(\Lambda)] , \quad (E.1) \]

with a variational posterior factorization over model parameters

\[ q(Z) = \prod_{a,r,w} \text{Cat}(z_{w}^{(ar)}, z_{w}^{(ar)}) \], \quad q(\Phi) = \prod_{h,r} \text{Dir}(\phi_{h}^{(r)}, \nu_{h}^{(r)}) \], \quad q(\eta) = \delta_{\eta*}(\eta), \]

\[ q(S) = \prod_{a} q(\{s_{w}^{(a)}\}_{w=1}^{W}) \], \quad q(\Lambda) = \prod_{g,a} \text{Dir}(\lambda_{g}^{(a)}, \kappa_{g}^{(a)}) \], \quad q(\beta) = \delta_{\beta*}(\beta), \]

\[ q(\Psi) = \prod_{a} \delta_{\psi(a)}^*(\psi^{(a)}) \], \quad q(\Theta) = \prod_{j} \delta_{\Theta_j^*}(\Theta_j) . \quad (E.2) \]

The following subsections detail the updates in this model.
E.1 Updates for cluster parameters

The variational updates for $\zeta_{s_{w\ell}}^{(ar)}$ and $\nu_{h\ell}^{(r)}$ are standard:

$$
\zeta_{s_{w\ell}}^{(ar)} \propto \sum_{h=1}^H \rho_{wh}^{(a)} \mathbb{E} \left[ \ln \phi_{h\ell}^{(r)} \right] - \frac{1}{2} \ln |\Sigma_{\ell}| - \frac{1}{2} y_w^{(ar)T} \Sigma_{\ell} y_w^{(ar)}
$$

(E.3)

$$
\nu_{h\ell}^{(r)} = \alpha_1 \eta^* + \sum_{a=1}^A \sum_{w=1}^W \zeta_{s_{w\ell}}^{(ar)} \rho_{wh}^{(a)}
$$

(E.4)

where $\mathbb{E} \left[ \ln \phi_{h\ell}^{(r)} \right] = \psi \left( \nu_{h\ell}^{(r)} \right) - \psi \left( \sum_{i=1}^L \nu_{hi}^{(r)} \right)$ and $\psi(\cdot)$ is the digamma function.

E.2 Updates for state parameters

States are assigned according to a hidden Markov model (HMM). The VBM step for this procedure is:

$$
\psi_{h}^{(a)*} \propto 1 + \mathbb{E} \left[ q(s_1^{(a)} = h) \right]
$$

(E.5)

$$
\kappa_{gh}^{(a)} = \alpha_0 \beta^* + \sum_{w=1}^W \mathbb{E} \left[ q(s_{w-1}^{(a)} = g, s_w^{(a)} = h) \right]
$$

(E.6)

where $\rho_{wh}^{(a)} = \mathbb{E} \left[ q(s_w^{(a)} = h) \right]$ and $\mathbb{E} \left[ q(s_{w-1}^{(a)} = g, s_w^{(a)} = h) \right]$ are obtained in a VBE step through the forwards backwards algorithm for variational inference. See (Beal, 2003) for more details.

E.3 Updates for global probability vectors

For both $\eta$ and $\beta$ we have a non-conjugate update. We generalize these two updates to be within the framework of the following update for $\beta$:

$$
\pi_c \sim DP(\alpha_0 \beta) \quad \beta \sim GEM(\gamma_0)
$$

(E.7)

where GEM($\gamma_0$) is the stick-breaking construction for the atom weights, defined by

$$
\beta_k = \beta_k' \prod_{i=1}^{k-1} (1 - \beta_i') \quad \beta_k' \sim Beta(1, \gamma_0)
$$

(E.8)
We define an objective function $\ell(\beta')$ for $\beta'$ as the portions of the variational log-posterior distribution that depends on $\beta'$. The goal is to obtain a point estimate for $\beta'$ that maximizes the objective. This log-posterior distribution is found as:

$$\ell(\beta') = \log \left( \prod_{k=1}^{K} \text{Beta}(\beta'_k; 1, \gamma_0) \right) + \mathbb{E} \left[ \log \left( \prod_{c=1}^{C} \text{Dir}(\pi_c|\alpha_0\beta) \right) \right] + \text{const} \quad \quad (E.9)$$

$$= \sum_{k=1}^{K} (\gamma_0 - 1) \log(1 - \beta'_k) + \sum_{c=1}^{C} \mathbb{E} \left[ - \log(B(\alpha_0\beta)) + \sum_{k=1}^{K} (\alpha_0\beta_k - 1) \log(\pi_{ck}) \right] + \text{const} \quad \quad (E.10)$$

$$= \sum_{k=1}^{K} (\gamma_0 - 1) \log(1 - \beta'_k) + \sum_{k=1}^{K} (\alpha_0\beta_k - 1) \sum_{c=1}^{C} \mathbb{E} \left[ \log(\pi_{ck}) \right] - C \log(B(\alpha_0\beta)) + \text{const} \quad \quad (E.11)$$

where $B(\cdot \cdot)$ is the multivariate beta function. In order to maximize this log-posterior objective, we take the derivative with respect of $\beta'$ and perform a first-order conjugate gradient optimization algorithm. The derivatives of each component ($\ell_1, \ell_2,$ and $\ell_3$) are found as follows:

- For $\ell_1(\beta')$,

$$\frac{\partial \ell_1}{\partial \beta'_k} = \frac{1 - \gamma_0}{1 - \beta'_k} \quad \quad \quad \quad (E.12)$$

- For $\ell_2(\beta')$, we first begin by expanding the stick-breaking representation of $\beta_k$

$$\ell_2 = \sum_{k=1}^{K} \left( \alpha_0\beta'_k \prod_{i=1}^{k-1} (1 - \beta'_i) - 1 \right) \sum_{c=1}^{C} \mathbb{E} \left[ \log(\pi_{ck}) \right] \quad \quad (E.13)$$

$$\frac{\partial \ell_2}{\partial \beta'_k} = \alpha_0 \prod_{i=1}^{k-1} (1 - \beta'_i) \sum_{c=1}^{C} \mathbb{E} \left[ \log(\pi_{ck}) \right] - \sum_{l=k+1}^{K} \alpha_0\beta'_l \prod_{i=1}^{l-1} (1 - \beta'_i) \sum_{c=1}^{C} \mathbb{E} \left[ \log(\pi_{cl}) \right]$$

$$= d_k \zeta_k - \sum_{l=k+1}^{K} e_{kl} \zeta_l \quad \quad (E.14)$$
• For \( \ell_3(\beta') \), we first begin by expanding the stick-breaking representation of \( \beta_k \)

\[
\ell_3 = -C \log(B(\alpha_0, \beta))
\]

(E.16)

\[
= -C \left[ \sum_{k=1}^{K} \log \Gamma(\alpha_0 \beta_k) - \log \Gamma \left( \alpha_0 \sum_{k=1}^{K} \beta_k \right) \right]
\]

(E.17)

\[
= -C \left[ \sum_{k=1}^{K} \log \Gamma(a_k) - \log \Gamma(b) \right]
\]

(E.18)

where the following equalities hold:

\[
a_k = \alpha_0 \beta_k
\]

(E.19)

\[
b = \alpha_0 \sum_{k=1}^{K} \beta_k
\]

(E.20)

\[
\frac{\partial a_{\ell}}{\partial \beta'_k} = \begin{cases} 
0, & \text{if } k > \ell \\
d_k, & \text{if } k = \ell \\
e_{k\ell}, & \text{if } k < \ell 
\end{cases}
\]

(E.21)

\[
\frac{\partial b}{\partial \beta'_k} = \sum_{\ell=1}^{K} \frac{\partial a_{\ell}}{\partial \beta'_k}
\]

(E.22)

We now take the derivative of \( \ell_3 \) with respect to \( \beta'_k \). In this derivation, we take the derivative of a gamma function where \( \frac{\partial \Gamma(x)}{\partial x} = \Gamma(x) \psi_0(x) \) where \( \psi_0(\cdot) \) is termed the polygamma function.

\[
\frac{\partial \ell_3}{\partial \beta'_k} = -C \left[ \sum_{\ell=1}^{K} \psi_0(a_{\ell}) \frac{\partial a_{\ell}}{\partial \beta'_k} - \psi_0(b) \frac{\partial b}{\partial \beta'_k} \right]
\]

(E.23)

\[
= -C \left[ d_k f_k - \sum_{\ell=k+1}^{K} e_{k\ell} f_\ell \right]
\]

(E.24)

where \( f_\ell = \psi_0(a_{\ell}) - \psi_0(b) \)

Combining equations E.12, E.15 and E.23, we are able to obtain the first deriva-
tive of the objective function $\ell(\beta')$ with respect to $\beta'_k$,

$$
\frac{\partial \ell}{\partial \beta'_k} = \frac{\partial \ell_1}{\partial \beta'_k} + \frac{\partial \ell_2}{\partial \beta'_k} + \frac{\partial \ell_3}{\partial \beta'_k}
$$

(E.25)

$$
= \frac{1 - \gamma_0}{1 - \beta'_k} \left[ d_k \zeta_k - \sum_{\ell=k+1}^{K} e_{k\ell} \zeta_{\ell} \right] - C \left[ d_k f_k - \sum_{\ell=k+1}^{K} e_{k\ell} f_\ell \right]
$$

(E.26)

$$
= \frac{1 - \gamma_0}{1 - \beta'_k} + d_k (\zeta_k - C f_k) + \sum_{\ell=k+1}^{K} e_{k\ell} (C f_\ell - \zeta_\ell)
$$

(E.27)

$$
= \frac{1 - \gamma_0}{1 - \beta'_k} + \alpha_0 \sum_{\ell=k}^{K} \frac{\partial \beta'_k}{\partial \beta'_k} \left[ \left( \sum_c \mathbb{E} [\log \pi_{c\ell}] \right) - C (\psi_0(\alpha_0 \beta_k) - \psi_0(\alpha_0)) \right]
$$

(E.28)

$$
= \frac{1 - \gamma_0}{1 - \beta'_k} + \alpha_0 \sum_{\ell=k}^{K} \frac{\partial \beta'_k}{\partial \beta'_k} \left[ \left( \sum_c \mathbb{E} [\log \pi_{c\ell}] \right) - C \mathbb{E} [\log \pi_{c\ell}] \right]
$$

(E.29)

Most line search methods should suffice to update $\beta$ using this derivative. We use a resilient back propagation method; see (Toussaint, 2012) for more details on this particular method.

E.4 Updates for kernel parameters

To update the parameters in $\Theta$, we take the derivative of the variational lower bound with respect to each parameter $\Theta_j$.

$$
\frac{\partial}{\partial \Theta_j} \mathbb{E} [\ln p(Y|Z, \Theta)] = \frac{1}{2} \sum_{\ell=1}^{L} \zeta_{w\ell}^{(ar)} \left[ y_w^{(ar)} \Sigma_{\ell}^{-1} \frac{\partial \Sigma_{\ell}}{\partial \Theta_j} \Sigma_{\ell}^{-1} y_w^{(ar)} - \text{tr} \left( \Sigma_{\ell}^{-1} \frac{\partial \Sigma_{\ell}}{\partial \Theta_j} \right) \right]
$$

(E.30)

where expectations are removed from the kernel parameters for notational simplicity and the derivative of the covariance matrix $\Sigma_{\ell}$ with respect to the parameter $\Theta_j$ is represented by the symmetric toeplitz matrix, $\frac{\partial \Sigma_{\ell}}{\partial \Theta_j} = \text{toeplitz}(\frac{\partial}{\partial \Theta_j} k(\tau; \theta_{\ell}, \gamma))$. The
derivatives of the kernel are computed to be

\[
\frac{\partial}{\partial w_q} k(\tau; \theta, \gamma) = \exp(-2\pi^2 \tau^2 \nu_q) \cos(2\pi \tau \mu_q) \tag{E.31}
\]

\[
\frac{\partial}{\partial \nu_q} k(\tau; \theta, \gamma) = -2\pi^2 \tau^2 w_q \exp(-2\pi^2 \tau^2 \nu_q) \cos(2\pi \tau \mu_q) \tag{E.32}
\]

\[
\frac{\partial}{\partial \mu_q} k(\tau; \theta, \gamma) = -2\pi \tau w_q \exp(-2\pi^2 \tau^2 \nu_q) \sin(2\pi \tau \mu_q) \tag{E.33}
\]

\[
\frac{\partial}{\partial \gamma} k(\tau; \theta, \gamma) = -\frac{1}{\gamma^2} \delta_r \tag{E.34}
\]

Using equation E.30, any simple gradient descent algorithm can be used to optimize the marginal likelihood. We use a resilient back propagation method in order to not rely heavily on the size of the gradients.
Bibliography


95


97


100
Biography

Kyle Ulrich was born in Monterey, California in 1990. He received his Bachelor of Science in Engineering in Electrical and Computer Engineering from Duke University and commissioned as an officer in the United States Air Force in May 2012. He received a Master of Science in Electrical and Computer Engineering in May 2015 and expects to receive a Ph.D. in Electrical and Computer Engineering in September 2016. He is a member of Eta Kappa Nu.