# Variational Inference for Nonlinear Regression Using Dimension Reduced Mixtures of Generalized Linear Models with Application to Neural Data

by

Vivek Subramanian

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
Sayan Mukherjee, Co-supervisor

_____
Jeffrey Beck, Co-supervisor

_____
Katherine Heller

_____
Miguel A. L. Nicolelis

Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Department of Statistical Science
in the Graduate School of Duke University
2015

# Abstract

## Variational Inference for Nonlinear Regression Using Dimension Reduced Mixtures of Generalized Linear Models with Application to Neural Data

by

Vivek Subramanian

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
Sayan Mukherjee, Co-supervisor

_____
Jeffrey Beck, Co-supervisor

_____
Katherine Heller

_____
Miguel A. L. Nicolelis

An abstract of a thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the Department of Statistical Science in the Graduate School of Duke University
2015

# Abstract

Brain-machine interfaces (BMIs) are devices that transform neural activity into commands executed by a robotic actuator. For paraplegics who have suffered spinal cord injury and for amputees, BMIs provide an avenue to regain lost limb mobility by providing a direct connection between the brain and an actuator. One of the most important aspects of a BMI is the decoding algorithm, which interprets patterns of neural activity and issues an appropriate kinematic action. The decoding algorithm relies heavily on a neural tuning function for each neuron which describes the response of that neuron to an external stimulus or upcoming motor action. Modern BMI decoders assume a simple parametric form for this tuning function such as cosine, linear, or quadratic, and fit parameters of the chosen function to a training data set. While this may be appropriate for some neurons, tuning curves for all neurons may not all take the same parametric form; hence, performance of BMI decoding may suffer because of an inappropriate mapping from firing rate to kinematic. In this work, we develop a non-parametric model for the identification of non-linear tuning curves with arbitrary shape. We also develop an associated variational Bayesian (VB) inference scheme which provides a fast, big data-friendly method to obtain approximate posterior distributions on model parameters. We demonstrate our model's capabilities on both simulated and experimental datasets.

# Contents

# List of Tables

# List of Figures

# List of Abbreviations and Symbols

## Abbreviations

| | |
|---|---|
| BMI | brain-machine interface |
| DP | Dirichlet process |
| DP-GLM | Dirichlet Process Mixture of Generalized Linear Models |
| EM | Expectation Maximization |
| GLM | generalized linear model |
| GMM | Gaussian mixture model |
| GP | Gaussian process |
| KL | Kullback-Leibler |
| LNP | linear-nonlinear-Poisson |
| LN-LNP | linear-nonlinear-linear-nonlinear-Poisson |
| M1 | primary motor cortex |
| MAP | maximum a posteriori |
| ML | maximum likelihood |
| MLE | maximum likelihood estimate |
| VB | variational Bayes |

## Symbols

| | |
|---|---|
| $x$ | A scalar (lowercase) |
| $\mathbf{x}$ | A vector (lowercase, bold) |

| | |
|---|---|
| $\mathbf{X}$ | A matrix (uppercase, bold) |
| $^\top$ | Transpose |
| $\text{vec}(\mathbf{X})$ | Vectorization of matrix $\mathbf{X}$ by stacking columns |

# Acknowledgements

This thesis represents a culmination of knowledge and expertise in Statistical Science that I have gained throughout my first three-and-a-half years as a graduate student. Completing this thesis took much more than just knowledge of statistics. This has been an incredible learning experience, and there are so many people for whose advice, mentorship, and friendship I am grateful.

I would first and foremost like to thank all my committee members for their feedback throughout the process of writing this thesis. I would especially like to thank Professors Jeff Beck and Miguel Nicolelis. Jeff, it has been a privilege to learn from and work with you, and I greatly appreciate your time and dedication to mentoring me. Your guidance was instrumental to my progress in completing this work. Miguel, thank you so much for your generous support over the past few years, without which none of this would have been possible.

I would like to thank all my fellow lab members (past and present), friends, and family for their support and for reminding me that there is a life outside of the lab/library. David Carlson, I have learned so much from you. Thank you for your encouragement, patience, and numerous adopt-a-cat links, which have never failed to motivate me. Maybe you will actually convince me to get my own cat one day... Kesh, Billy, Ahmed, Jonathan, Justin, and Charmaine - thank you all for making me laugh; for keeping me sane; and for the great company over the years. Your friendship means so much to me. Albert, Hui, Judy, Kevin1 and Kevin2 - you promised you

would come back for my MS defense! Where are you guys? Also, Blue Express after? I heard the special is chicken cordon bleu...

Mom and Dad - thank you for all the love, encouragement, and support you have given me since the beginning. I love you very much.

# 1

# Introduction

## 1.1 Brain-Machine Interface and a Review of Motor Tuning

Brain-machine interfaces (BMIs) are devices that interpret patterns of neural activity to control robotic actuators. For patients with spinal cord injury and for amputees, BMIs can provide a means to regain limb control by bypassing severed neural circuits and directly communicating with a robotic actuator. Invasive BMIs operate by processing firing rates from individual neurons or groups of neurons. These firing rates can be recorded using extracellular electrodes that are part of electrode arrays. Such arrays can record from hundreds to thousands of neurons and are implanted in the primary motor cortex (M1) and other areas of the brain known to contain information relevant to motion.

The first successful demonstration of a BMI was peformed by Chapin and colleagues in the Nicolelis laboratory (Chapin et al. (1999)). In this pioneering work, rats were implanted with electrodes and trained to obtain a reward by controlling a lever to reach a target. Artificial neural networks were used to decode neuronal population activity into lever movement and showed that rats could achieve nearly 100% efficiency in controlling the lever for a reward over a 280-second period (Chapin

et al. (1999)). These results set off a wave of research in the BMI field. Taylor et al. (2002) developed a neuroprosthetic that could be controlled in 3D by a rhesus monkey. Schalk et al. (2004) built a tool called BCI:2000 which streamlined the signal processing pipeline in EEG-based BMIs. Velliste et al. (2008) followed up with a BMI that could process cortical neural activity of a rhesus monkey to not only allow realistic 3D movement but also allow the subject to control the strength of grip of a robotic hand. O'Doherty et al. (2011) built a BMI capable of providing sensory feedback to the user through cortical microstimulation. Ifft et al. (2013) trained rhesus monkeys to operate a bimanual BMI, which allows both arms to be simultaneously controlled. More recent work has focused on brain-to-brain and brain-plus-brain interfaces which allow subjects to cooperatively operate BMIs by sharing or integrating information (Pais-Vieira et al. (2015); Ramakrishnan et al. (2015)).

The BMI decoder translates neural activity into kinematic commands by inverting a so-called tuning curve, which relates neural activity to a behavioral parameter such as arm position or velocity. For invasive BMIs, this neural activity takes the form of firing rates; hence, a tuning curve for a given neuron shows how firing rate is modulated as a function of behavior. Neurons only respond when stimuli are present within their receptive fields, which are classically defined as the region in sensory space which elicits a neural response from a sensory neuron. To illustrate the concept of a receptive field, figure 1.1 depicts the receptive field of a retinal ganglion cell, which is a type of neuron commonly studied in sensory neuroscience. In each of the subplots, the large outer circles indicate the field of view of the retina. Yellow and purple regions indicate regions in the field of view which cause neurons to fire or stop firing depending on whether the cell is an on center cell or off center cell. For the on center cell, the neuron fires when light strikes the center of the field of view. On the other hand, for the off center cell, the ganglion cell tonically fires and stops firing only when light strikes the center. Opposite effects are observed in

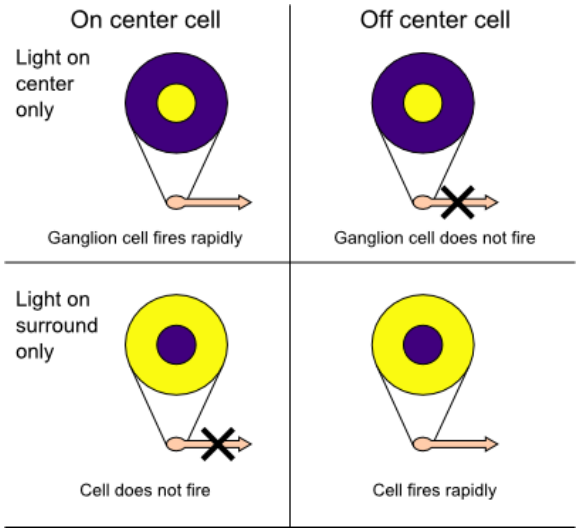



FIGURE 1.1: Receptive field of a retinal ganglion cell.

each cell when the light strikes the region surrounding the center.

The tuning function relates the firing rate of the neuron to the "intensity" of the stimulus in the receptive field. Figure 1.2 gives an example of a tuning function of a neuron in the primary visual cortex, which was first discovered by Hubel and Wiesel (1959). The neuron has a receptive field which is sensitive to the orientation of bars observed in the environment, some of which are shown under "stimulus" on the left subplot. The middle subplot ("response") shows a raster plot of spikes generated by the neuron for different bar orientations. These firing rates are averaged over many trials and the count is plotted as a function of bar angle in the right subplot, which depicts the "tuning curve."

In the context of motor neurons, the receptive field is the region in "behavioral space" which elicits a neural response. The tuning function then gives the firing rate of the neuron as a function of stimulus within this "behavioral receptive field." Figure 1.3 illustrates this concept for a hypothetical neuron whose receptive field includes only outward movements of the arms. The left half of the plot illustrates movement of the left arm, and the right half movement of the right arm. When the arms are

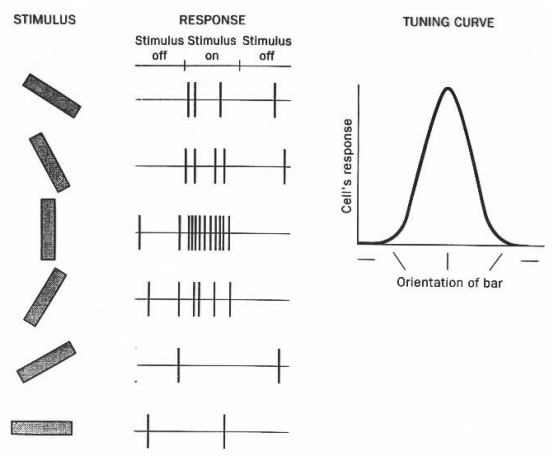



FIGURE 1.2: Tuning function of a neuron in the primary visual cortex.

moved a great deal away from the center (top plot) the firing rate is maximal. When they are moved slighltly away, the firing rate is less. When the arms are moved in the perpendicular or antiparallel directions, the firing rate is small or zero.

State-of-the-art BMIs employ various parametric forms for tuning functions. For instance, van Hemmen and Schwartz (2008) employ cosine tuning, which is also

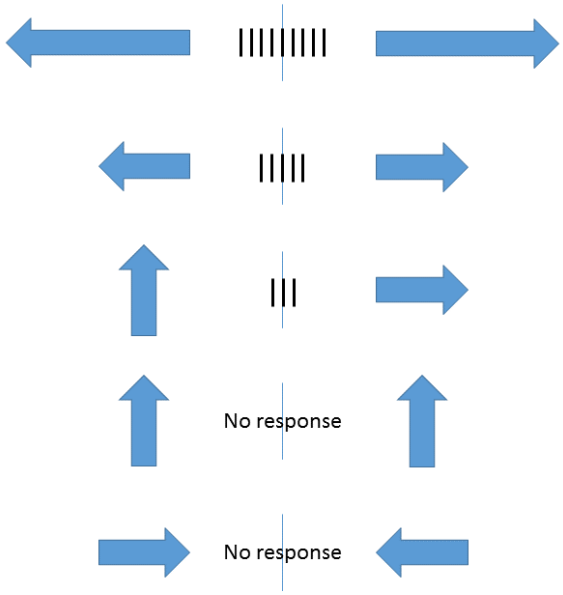



FIGURE 1.3: Example of a neuron whose receptive field encompasses only outward movements of the arms.

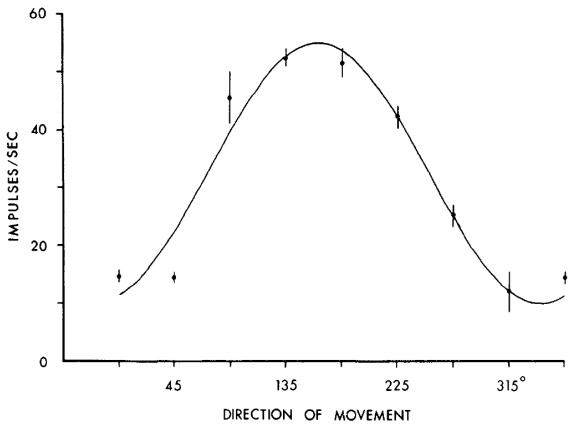



FIGURE 1.4: Cosine tuning function. Firing rate of a motor cortical neuron is maximal at $\theta = 170\circ$ and then behaves as $\cos(\theta)$.

known as the population vector algorithm. This model suggests that each neuron has a single "preferred direction" $\theta$ of movement for which it fires with greatest intensity; as one moves further from this direction, the firing rate decreases as $\cos(\theta)$. Figure 1.4 depicts the cosine tuning function first identified in Georgopoulos et al. (1982).

Li et al. (2009) compare linear and quadratic tuning functions to position and velocity in combination with an unscented kalman filter decoder. The quadratic tuning function also assumes a neuron has a preferred "direction," in the sense that a neuron fires maximally for a linear function of behavioral parameters and neigboring "directions" elicit firing rates which fall off quadratically. They show that decoding performance significantly improves when employing a quadratic tuning function compared to a linear function.

Evidence suggests that motor neurons may have more than one preferred direction and may be tuned to behavioral parameters other than simply direction, position (absolute or relative), velocity, etc. For instance, Sergio et al. (2005) show that while a monkey performs isometric force and arm movement tasks, a neuron's preferred direction can switch as the subject executes a movement. In fact, 59 out of 132 neurons that were recorded had preferred direction changes greater than $90^\circ$ immediately

after force was applied. Moran and Schwartz (1999) map firing rates of M1 neurons measured during a center-out/finger spiral/figure-eight drawing task to direction, speed, and multiple joint angles. They show that the same neuron can be tuned to multiple kinematic variables and that hand trajectory (a combination of direction, speed, and joint angle) can be accurately reconstructed from the cortical activity. In addition, there has been much debate centered around whether motor neurons utilize "force control" or "position control" to execute limb movements. The former relies on the assumption that the neurons encode Newtonian mechanics and specify the underlying forces and torques necessary to execute a movement (Bizzi et al. (1984)) while the latter suggests that movements are the result of brain-controlled transitions among so-called "equilibrium points" (Asatryan and Feldman (1965)). Thus, limbs provide proprioceptive feedback which the brain interprets in order to modulate position. Lastly, while a neuron might not be *directly* tuned to a given stimulus, the inputs it receives from upstream neurons that *are* tuned to the stimulus can interact at the dendrite (where inputs converge) to influence firing rates nonlinearly.

## 1.2 The Linear-Nonlinear-Poisson Model

To determine how to model this complex, nonlinear tuning function, we start by branching out of the BMI literature and into the computational neuroscience literature. The McCulloch-Pitts model forms the foundation of many modern nonlinear regression models (McCulloch and Pitts (1943)). This model integrates synaptic inputs to a neuron using a weight vector and passes the output of the integration through a threshold nonlinearity. Parameters to be inferred include the weights and the threshold. While useful conceptually, this model is overly simplistic: it ignores the fact that synaptic inputs are stochastic (i.e. just because a presynaptic neuron caused a postsynaptic neuron to fire does not mean the postsynaptic neuron will propagate the information to the neuron being measured) and not memoryless.

The McCullogh-Pitts model led to a much more commonly used model known as the linear-nonlinear-Poisson (LNP) model (Schwartz et al. (2006)). This model describes the relationship between stimulus and single neuron firing rate and consists of three stages. First, the stimulus vector is projected onto (or convolved with) a linear filter, which is the receptive field of the neuron. Second, a static nonlinearity (such as logistic sigmoid) is applied to the output of the convolution. The result is then treated as the mean firing rate in an inhomogenous Poisson process. In summary, this is given by:

$$p(y_t|\mathbf{x}_t) = \frac{f(\mathbf{k}^\top \mathbf{x}_t)^{y_t}}{y_t!} \exp\left[-f(\mathbf{k}^\top \mathbf{x}_t)\right] \tag{1.1}$$

where $\mathbf{k}$ is the receptive field and $f$ is the static nonlinearity. The LNP model accounts for stochasticity in neural inputs, and moreover, several biologically realistic models that account for ion channel activity (including the leaky integrate-and-fire, where the leak refers to the leakage of input current when when the neuron does not reach threshold voltage after receiving input; exponential integrate-and-fire; and generalized exponential models) can be closely approximated by the LNP model.

More recently, LNP models have been extended to linear-nonlinear-LNP (LN-LNP) models, which hierarchically chain many LNP-modeled neurons to model computation in dendritic trees (Vintch et al. (2012)). While the first layer models many neurons as LNP, the second layer combines the output of the first layer and passes this output through a separate nonlinearity to obtaining a mean firing rate. Although simple LNPs are easy to fit, they lack the richness of LN-LNP models. On the other hand, LN-LNP models are much more difficult to fit; Wu et al. (2015) presents way to achieve maximum likelihood (ML) esimates for a specific choice of the two nonlinearities.

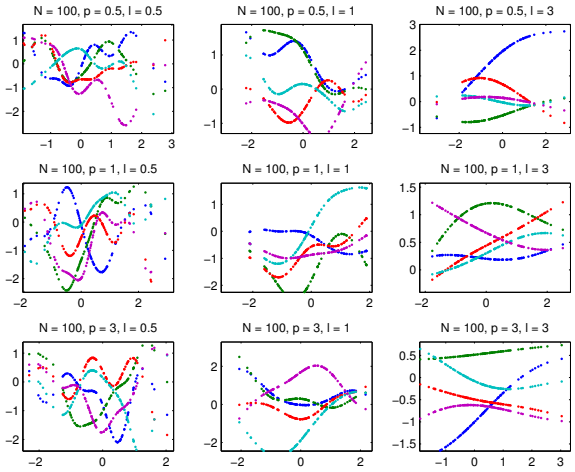



FIGURE 1.5: Samples from a Gaussian process with $N = 100$, different pre-factors $p$, and different length scales $l$. The pre-factor controls the overall variance of the generated function, and the length scale controls the amount of change in the input needed to achieve a prescribed variance in the output.

## 1.3 Statistical Models for Nonlinear Regression

Having briefly explored the recent neural modeling literature, we turn to the stastistics literature to seek flexible yet tractable models for nonlinear regression. Rasmussen (2006) proposes Gaussian processes (GPs) for Bayesian nonlinear regression. GPs model the joint distribution of any subset of observations as Gaussian. Furthermore, the covariance between observations is a function of only the regressors. Hence, given a valid covariance function and a set of observations, one can make predictions by employing conditional covariance properties of the Gaussian distribution. GPs are useful because analytic expressions can be provided for predictive distributions and because they can be highly nonlinear (see figure 1.5).

Fitting a single GP to the entire regressor space could cause local trends to get washed out. One could therefore extend the GP to a "mixture" of GPs, which consists of many components that together span the full regressor space. Each component could be fit with the same kernel function but with different parameters or with different kernel functions altogether. Rasmussen and Ghahramani (2002) and Snel-

son and Ghahramani (2007) provide some theoretical results for such GP mixtures, demonstrating that local approximations improve predictive performance without significant addition to the computational cost. One issue that comes up is how to choose the number of mixture components $K$. Rasmussen and Ghahramani (2002) employ the Dirichlet process (DP), which is an elegant approach to choosing $K$ and is ubiquitous in nonparametric Bayes literature (Ferguson (1973)). Formally, the Dirichlet process is defined as follows (El-Arini (2008)). Assume a continuous base measure $G_0$ over a support $A$ and a positive, real-valued concentration parameter $\alpha$. Then, for any finite set of partitions $A_1 \cup A_2 \cup \ldots \cup A_K$:

$$(G(A_1), \ldots, G(A_K)) \sim \text{Dirichlet}(\alpha G_0(A_1), \ldots, \alpha G_0(A_K)) \tag{1.2}$$

Although useful in theory, this form does not lend much to intuition. Two constructive definitions of the DP are particularly useful. The first is the Chinese restaurant process (CRP). In this culinary analogy, a customer walks into a restaurant and sits down at the first table. For every future customer $i$, the probability of sitting at an existing table $j$ is proportional to the number of customers $n_{-i,j}$ already at that table, and the probability of creating a new table is proportional to the concentration parameter $\alpha$ of the Dirichlet process. Mathematically, this is given by:

$$p(z_i = j) = \frac{n_{-i,j}}{n + \alpha - 1} \tag{1.3a}$$

$$p(z_i \neq z_{i'} \; \forall \; i \neq i') = \frac{\alpha}{n + \alpha - 1} \tag{1.3b}$$

Here, $n$ is the total number of customers at the restaurant. If we think of the customers as data points and the tables as clusters (more specifically, the cluster means and covariances), the CRP analogy illustrates the "clustering effect" or "rich-gets-richer" property of the DP (Aldous (1985)). The DP corresponds to the limit as the number of customers, and hence, the number of tables, goes to infinity. The

stick-breaking analogy is another useful analogy in which a stick of unit length is broken into infinitely many pieces whose lengths are given by draws from a Beta$(1, \alpha)$ distribution. We see again that the distribution is discrete and the greater the value of alpha the more the number of sizable sticks. Referring back to equation (1.2), we see that the partitions correspond to cluster parameters and the $G(A_k)$'s to the mixing coefficients.

The DP is elegant for several reasons. First, since it is essentially a distribution over distributions, it can be directly incorporated into a probabilistic modeling framework. Second, although the two analogies above may intuitively suggest that data points modeled as coming from a DP are not exchangeable, writing out the joint distribution shows that they in fact are. Hence, the order in which the data points is observed does not matter for DP inference. Third, because of the clustering property, if $\alpha$ is chosen to be small relative to the number of data points, the DP will naturally prefer models that are simpler (i.e. containing fewer clusters). Given enough data, the model will discover the number of mixture components in the inference process by maximizing the posterior probability of the data given the number of components and their values. Last but not least, fast approximate inference for the DP is relatively straightforward and can be performed using variational inference, which will be discussed below.

Motivated by an understanding of the DP which gives us a nonparametric way to identify $K$ in the mixture of GPs, we could in theory proceed by fitting a nonparametric mixture of GPs to our data to identify the nonlinearities. GPs present several drawbacks, however. First, the covariance kernel $K(x_1, x_2)$, where $x_1$ and $x_2$ are the data, must be chosen, which is in a sense very similar to the underlying problem we are trying to eliminate (i.e. choosing a sigmoid, quadratic, etc. function for the tuning curve). Second, the conditional covariance of a Gaussian distribution requires matrix inversion of an $N \times N$ covariance matrix, where $N$ is the number

of observed data points. Matrix inversion is an $\mathcal{O}(N^3)$ operation, and is hence very inefficient. Observations must be also be Gaussian distributed, which may not be the case for neural data (Pillow et al. (2003)). Therefore, we seek methods that are fast and allow us to do nonparametric inference for nonlinearities.

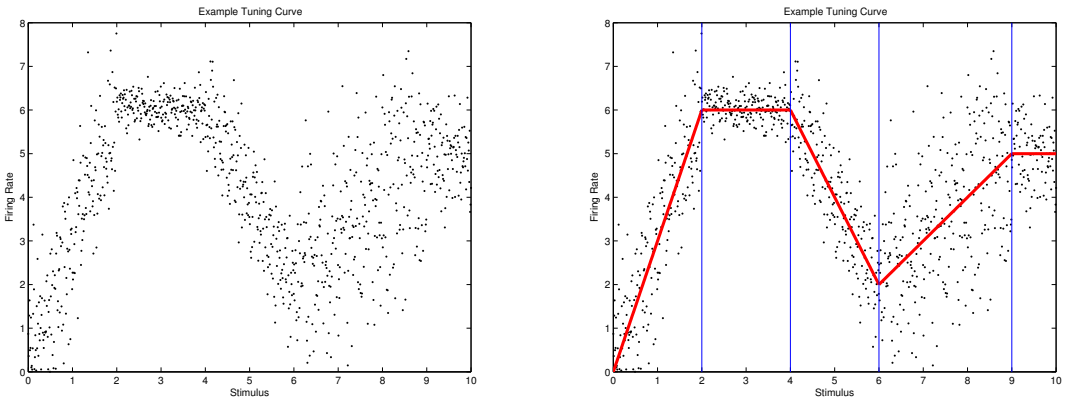
One substitute for the GP that comes to mind is the generalized linear model (GLM) (Nelder and Baker (1972)). Briefly, a GLM is linear model whose error covariance is not restricted to the Gaussian distribution. Furthermore, the response variable is related to a linear function of the regressors by an invertible link function. For example, for Poisson regression, the GLM is given by:

$$\mathbb{E}[y_t|\mathbf{x}_t, \boldsymbol{\theta}] = \exp\left(\boldsymbol{\theta}'\mathbf{x}_t\right) \tag{1.4}$$

where $\exp(\cdot)$ is the link function. Hence, the PMF is given by:

$$p(y_t|\mathbf{x}_t, \boldsymbol{\theta}) = \frac{\mathbb{E}[y_t|\mathbf{x}_t, \boldsymbol{\theta}]^{y_t} \times \exp\left(-\mathbb{E}[y_t|\mathbf{x}_t, \boldsymbol{\theta}]\right)}{y_t!} \tag{1.5}$$

Just as we did for GPs, we can apply the idea of a mixture model to GLMs. Please see figure 1.6 for an illustration of fitting a neural tuning curve with a GLM mixture. Figure 1.6a depicts an example tuning curve. While this looks like it may be fitted with a parametric function such as cubic or sin, there are regions in the stimulus space where the function does not match either of these functions well. For instance, between stimulus values of 2 and 4 and between 9 and 10, the function flattens out, and around a stimulus value of 6, there is a sharp transition from negative to positive slope. Furthermore, the noise variance is siginificantly larger in the region between 6 and 9 compared to the other regions. (Note that this is also just one example of a tuning function, and that, in reality, tuning functions can be much more complex. Additionally, there can be several different, nonlinearly interactive receptive fields for which the neuron exhibits some response.) To fit this using a mixture of GLMs, we could employ a linear link function for each mixture component, which implies




(a) Simulated neural tuning curve.

(b) Fit with mixture of GLMs.

FIGURE 1.6: Simulated neural tuning curve fitted with mixture of GLMs.

that, locally, there is a linear relationship between stimulus and firing rate. In order to choose the number of components, we could perform nonparametric clustering on the stimulus space. This would allow us to identify regions in the stimulus space for which the linear model of neural firing rate has the same slope and intercept. This clustering and fit with a linear link function is illustrated in figure 1.6b.

Hannah et al. (2011) propose a Dirichlet Process mixture of Generalized Linear Models (DP-GLM) to address the problem of nonparametric Bayesian regression. The DP-GLM fits an infinite mixture of GLMs, where each component is specialized for a local subset of covariates. The DP-GLM is a rich, flexible model in that each component can have a completely different GLM associated with it, and the total number of components is learned from the data. More specifically, the noise distribution of each GLM component need not be Gaussian. State-of-the-art inference algorithms for DP-GLMs rely on sampling-based methods, which are known to be slow and are only accurate as the number of samples taken approaches infinity, which is never true in practice. Furthermore, the model does not perform any dimensionality reduction, so high-dimensional regressors would require fitting a large number of parameters.

In the following chapter, we describe our model, which is an extension of the DP-GLM. In addition to the features of the DP-GLM, the proposed model allows for identification of nonlinearly interacting receptive fields. In the process, we also perform a dimensionality reduction so that rather than working in the stimulus space directly, the model is fit in the receptive field space, which is typically of much lower dimensionality than the stimulus space. Finally, our model is fit with VB inference, which is a fast alternative to sampling-based inference that has numerous other advantages, which will be discussed.

# 2

# Model

## 2.1 Model Formulation

We provide here details of the statistical model used for fitting the nonlinear tuning function. Figure 2.1 illustrates the model in graphical form. $\eta_t$ represents the firing

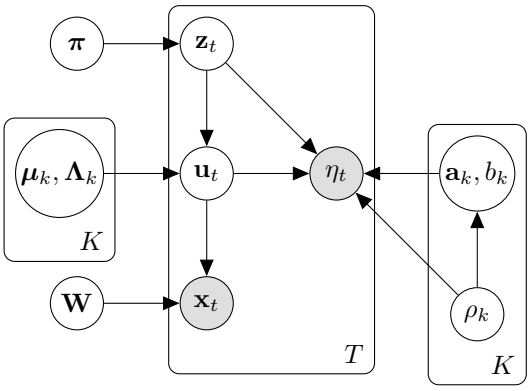



FIGURE 2.1: Graphical model of nonlinear regression.

rate of a single neuron. (While one could model many neurons jointly, we consider one neuron at a time for simplicity.) $\mathbf{x}_t$ represents some observed stimulus or behavioral parameter; for example, this could be arm position or velocity. Rather than perfoming the non-linear regression directly on these behavioral variables, we do the regression on a lower dimensional latent variable $\mathbf{u}_t$. This is useful and important

for two reasons. First and foremost, a neuron's firing rate may not be tuned to velocity, position, or any other behavioral measurements in particular. Hence, we can apply $\mathbf{W}$ to $\mathbf{x}_t$ to find a low-dimensional projection of these measurements to which the neuron is best tuned. Second, without the projection, the number of regression parameters that we must learn scales linearly with the number of behavioral measurements. If this is very large compared to the size of the dataset, one can suffer from problems in estimating model parameters due to overfitting. Thus, we model an explicit dimensionality reduction on a full set of regressors that is cluster-independent, i.e. $\mathbf{u}_t = \mathbf{W}\mathbf{x}_t$. This also allows us to discover non-linear interactions between linear receptive fields.

In order to discover the nonlinearity, we use a piecewise linear function modeled by a mixture of Gaussians. Each Gaussian has a mean, which is a function of $\mathbf{a}_k$, $b_k$, and $\mathbf{W}$. Here $\mathbf{a}_k$ controls the slope of each line while $b_k$ controls the intercept; when the model is fit, the transformation $\mathbf{a}_k\mathbf{u}_t + b_k$ gives the mean of the regressors and provides a smoothed estimate of the tuning curve when averaging is performed with posterior cluster probabilities. $\mathbf{W}$ simply acts on $\mathbf{x}_t$ to transform it to the lower-dimensional space on which we are modeling the tuning function. The parameter $\rho_k$ specifies the precision of the observed firing rate and is specific to each Gaussian cluster; thus, $\rho_k$ allows the tuning curve to be heteroscedastic, with different variance along the $\mathbf{u}_t$ axis.

The number of clusters (i.e. piecewise linear components) is learned using a Dirichlet distribution approximation to the Dirichlet process. The Dirichlet is a distribution over distributions on the $K$-dimensional simplex. It can be parameterized by a single concentration parameter $\alpha_0$. As the number of clusters $K$ approaches $\infty$, we recover the Dirichlet process. (Although the stick-breaking prior can be used explicitly in practice by truncating the number of components, the results are often similar to those obtained using a Dirichlet prior with the same cap on number of

components. We use the Dirichlet prior here for convenience.) Because the Dirichlet process has a so-called "rich get richer" property, the model favors placing data points into as few clusters as possible, so there is a built in penalty for overfitting.

The observations and parameters are modeled as follows:

$$\boldsymbol{\pi} \sim \text{Dirichlet}(\alpha_0, \ldots, \alpha_0) \tag{2.1}$$

$$\mathbf{z}_t|\boldsymbol{\pi} \sim \text{Multniomial}(\boldsymbol{\pi}) \tag{2.2}$$

$$\boldsymbol{\Lambda}_k \sim \text{Wishart}(\nu_0, \mathbf{P}_0) \tag{2.3}$$

$$\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k \sim \mathcal{N}(\mathbf{m}_0, (\kappa_0\boldsymbol{\Lambda}_k)^{-1}) \tag{2.4}$$

$$\mathbf{W} \sim \text{matrix-}\mathcal{N}(\mathbf{M}_0, \mathbf{U}_0, \mathbf{V}_0) \tag{2.5}$$

$$\rho_k \sim \text{Gamma}(\alpha_0, \beta_0) \tag{2.6}$$

$$[\mathbf{a}_k, b_k]^\top|\rho_k \sim \mathcal{N}(\mathbf{g}_0, (\rho_k\mathbf{B}_0)^{-1}) \tag{2.7}$$

$$\mathbf{u}_t|\{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\}_K, z_{tk} \sim \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{tk}} \tag{2.8}$$

$$\eta_t|\{\mathbf{a}_k, b_k, \rho_k\}_K, \mathbf{u}_t, z_{tk} \sim \prod_{k=1}^{K} \mathcal{N}(\mathbf{a}_k^\top\mathbf{u}_t + b_k, \rho_k^{-1})^{z_{tk}} \tag{2.9}$$

# 3

# Inference

## 3.1 Expectation Maximization

In order to take advantage of the richness of the DP-GLM model framework without sacrificing on speed, we employ variational Bayesian (VB) inference (Beal (2003)). To motivate VB, we start with its simpler predecessor, the expectation maximization (EM) algorithm (Dempster et al. (1977)). For models involving latent variables, the EM algorithm allows one to find ML (or maximum a posteriori, i.e. MAP) estimates of the parameters by proceeding alternately between two steps: (i) expectation and (ii) maximization. EM can be illustrated very simply for ML estimatation in a Gaussian mixture model (GMM). For a GMM, the probabilistic model is given by:

$$p(\mathbf{X}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\pi}) = \prod_{t=1}^{T} \sum_{k=1}^{K} [\pi_k \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \tag{3.1}$$

where $\mathbf{X}$ is a $T \times d$ matrix of $\mathbf{x}_t$'s. Parameters are defined as variables which are fixed in number with respect to the size of the dataset; thus, the parameters to estimate are the cluster means $\boldsymbol{\mu}_k$, cluster covariances $\boldsymbol{\Sigma}_k$, and mixing proportions $\pi_k$. If we attempt to maximize the log likelihood (which is equivalent to maximizing the

likelihood since ln is a monotonic function) directly, i.e. maximize

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{t=1}^{T} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{3.2}$$

we run into two issues. First, the ln acts on the sum over $k$, which means that expressions for derivatives with respect to $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and $\boldsymbol{\pi}$ can be complicated (or, in some cases, intractable). These are needed in order to find values that maximize the likelihood. Second, the MLE could result in a cluster mean collapsing onto a single data point. In this case, the variance of the cluster would shrink to zero, resulting in extreme overfitting; i.e. when clustering new data points, they would all be assigned to the other components unless they had a value exactly equal to the overfitted mean.

The EM algorithm remedies the first issue by augmenting the model with a set of latent variables $\mathbf{z}_t$. Latent variables increase in number with the number of observations. We first define the complete likelihood function as the distribution over both observations and latent variables, i.e.:

$$p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{t=1}^{T} \prod_{k=1}^{K} \pi_k^{z_{tk}} \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{tk}} \tag{3.3}$$

Here, $\mathbf{z}_t$ follows a categorical distribution, and $z_{tk}$ is an indicator which has a value of 1 when $z_{tk} = 1$ and 0 otherwise. Taking the log, we notice immediately that the ln no longer acts on a sum, but rather, it directly acts on the Gaussian density:

$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{t=1}^{T} \sum_{k=1}^{K} z_{tk} [\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \tag{3.4}$$

In this form, the expressions for the MLEs are greatly simplified.

Utilizing the complete likelihood function assumes that we know the cluster labels $\mathbf{z}_t$ when in fact, they are unknown. As an intermediate step, we can average, or take the expectation of, the complete data likelihood function over the posterior probability of $\mathbf{z}_t$ using the current estimates of the model parameters. This intermediate

step is known as the expectation, or E, step. We write:

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})] = \sum_{t=1}^{T}\sum_{k=1}^{K} \mathbb{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}}[z_{tk}][\ln \pi_k + \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \tag{3.5}$$

and we find $\mathbb{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}}[z_{tk}]$ to be:

$$\mathbb{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}}[z_{tk}] = \frac{\pi_k \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{3.6}$$

which is simply the posterior cluster assignment probability. We can then maximize (3.5), which is known as the M step. Thus, we can alternate between computing expectations in (3.6) and maximizing (3.5) until the algorithm reaches some convergence criterion (e.g. change in log likelihood is less than some $\epsilon$). Together, these steps constitute the EM algorithm. We refer the reader to Bishop (2006) for a full derivation of EM. Note the similarity between the EM algorithm for Gaussian mixtures and the $K$-means algorithm (MacQueen et al. (1967)). While in $K$-means we use "hard" cluster assignments for data points, in EM for GMMs, we model the data as conditionally Gaussian with some mean and covariance and, hence, are able to provide "soft" cluster assignments.

## 3.2 Maximizing the lower bound

We made two major jumps in the explanation of the EM algorithm. First, why did we compute the expectation with respect to the posterior distribution of $\mathbf{Z}$? And second, how do we know the EM algorithm will converge? To answer these questions, we first define a functional as a mapping which takes in a function and outputs a scalar value (Bishop (2006)). Next, we introduce a functional on latent variables $q(\mathbf{Z})$, and consider the following decomposition of the likelihood function on our set of parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}\}$ (Bishop (2006)):

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \mathrm{KL}(q||p) \tag{3.7}$$

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \right\} \tag{3.8}$$

$$\mathrm{KL}(q||p) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})} \right\} \tag{3.9}$$

where KL is the Kullback-Leibler divergence between $q(\mathbf{Z})$ and $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$ (Kullback and Leibler (1951)). Since KL divergence must be $\geqslant 0$, this set of equations shows that the log likelihood can be decomposed into the sum of a lower bound $\mathcal{L}$ (given the current estimate of the parameters and choice of $q$) and the KL divergence between the functional of $\mathbf{Z}$ and posterior distribution on $\mathbf{Z}$.

EM is essentially a coordinate ascent algorithm in which each step iteratively maximizes the lower bound on the log likelihood $\mathcal{L}(q, \boldsymbol{\theta})$ with respect to $q$ and $\boldsymbol{\theta}$ and adjusts the KL divergence. Assume that at the start of the current iteration of EM, our parameter $\boldsymbol{\theta}$ equals $\boldsymbol{\theta}_0$. Then, in the E step, if we choose $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0)$ and plug this expression into (3.8), we see that the lower bound reduces exactly to the expression for the E step we found in (3.5) above and justifies choosing $q(\mathbf{Z})$ equal to the posterior distribution of $\mathbf{Z}$. Moreover, the KL divergence disappears and the lower bound is maximized with respect to $q(\mathbf{Z})$.

The M step seeks to maximize the log likelihood with respect to $\boldsymbol{\theta}$. We know that since the KL divergence term in (3.7) is 0 after the E step, maximizing the log likelihood is equivalent to maximizing the lower bound. This in turn is equivalent to maximizing (3.5) since the expressions are equal. Unless we are already at a local maximum, this will necessarily cause the lower bound to increase. KL divergence, by definition, must be $\geqslant 0$, and since changing $\boldsymbol{\theta}$ will change the posterior on $\mathbf{Z}$, the KL divergence will be nonzero. Hence, the log likelihood function will increase, and the increase will be greater than the increase in the lower bound since it also includes the positive KL divergence term. This process of recomputing $q(\mathbf{Z})$ and maximizing the lower bound with respect to $\boldsymbol{\theta}$ continues until we reach a local maximum. After this

point, the KL divergence will always be zero, and the algorithm will have converged. Hence, convergence to a local maximum is guaranteed.

## 3.3 Variational Inference

Now that we have proper background and motivation from the EM algorithm, we return our attention to VB. VB has its origins in mean field theory with applications to statistical physics and was more recently employed for fitting hierarchical Bayesian models (Weiss (1907); Parisi (1988); Beal (2003)). The name VB comes from the calculus of variations, which deals with functionals and functional derivatives. Functional derivatives describe how the value of a functional changes as the input function is changed infinitesimally (Bishop (2006)). In VB, we seek functionals that best approximate the posterior distributions of the parameters and latent variables in the model. These approximate posteriors allow us to avoid overfitting and to perform model comparison by calculating the marginal likelihood of the data (or a lower bound thereof) and computing Bayes factors. (Technically, overfitting can be avoided in EM by placing priors on parameters and searching for MAP estimates rather than MLEs. This still only gives point estimates, however, so we cannot perform model comparison with EM.) Like EM, VB is a coordinate ascent algorithm which allows us to avoid having to choose a learning rate parameter which is often required for gradient descent-based algorithms. In practice, convergence requries only a handful of VB iterations as opposed to thousands of samples that would need to be drawn when doing inference via sampling. This makes VB big-data friendly. Note that, like EM, VB does not guarantee convergence to the global mode of the joint posterior - only to a local mode. Hence, initialization of parameters is important for converging to a "desirable" local mode, i.e. one which by definition locally maximizes the posterior probability of the data but also yields predictions that are in the vicinity of the true values.

We start by referring back to equations (3.7)-(3.9). (Note that there are many similarities between VB and EM, so it is useful to keep in mind the derivations of the E and M steps in EM.) In VB inference, we group all latent variables and parameters into $\mathbf{Z}$. Like in EM, we can decompose the log marginal likelihood as follows (Bishop (2006)):

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q||p) \tag{3.10}$$

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \tag{3.11}$$

$$\mathrm{KL}(q||p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \tag{3.12}$$

As in the E step of the EM algorithm, we can maxmize the lower bound by zeroing out the KL divergence, which occurs when $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X})$. In most situations, $p(\mathbf{Z}|\mathbf{X})$ is intractable, so we seek approximate posteriors which yield tractable normalizing constants but also provide a sufficiently good approximation to the true posterior. We assume we can factorize the variational posterior into $M$ groups, such that:

$$q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(\mathbf{Z}_i) \tag{3.13}$$

We seek to maximize the lower bound with respect to each of the $q_i$'s. First, we substitute (3.13) into (3.11) and consider only one specific $q_j$:

$$\begin{aligned}
\mathcal{L}(q) &= \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \\
&= \int \prod_i q_i \left\{ \ln p(\mathbf{X}, \mathbf{Z}) - \sum_i \ln q_i \right\} d\mathbf{Z} \\
&= \int q_j \left\{ \int \ln p(\mathbf{X}, \mathbf{Z}) \prod_{i \neq j} q_i d\mathbf{Z}_i \right\} d\mathbf{Z}_j - \sum_j \int q_j \ln q_j d\mathbf{Z}_j \int \prod_{i \neq j} q_i d\mathbf{Z}_i \\
&= \int q_j \ln \widetilde{p}(\mathbf{X}, \mathbf{Z}_j) d\mathbf{Z}_j - \int q_j \ln q_j d\mathbf{Z}_j + \text{const.}
\end{aligned} \tag{3.14}$$

where

$$\ln \widetilde{p}(\mathbf{X}, \mathbf{Z}_j) = \int \ln p(\mathbf{X}, \mathbf{Z}) \prod_{i \neq j} q_i d\mathbf{Z}_i + \text{const.}$$

$$= \mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Z})] + \text{const.}$$

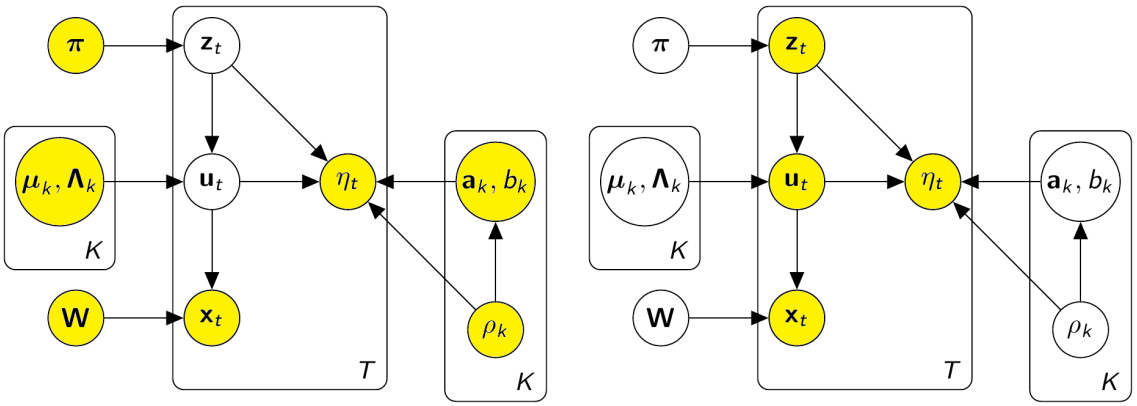
Also note that the terms not involving $q_j$ (namely, $q_i \ln q_i \, \forall \, i \neq j$) have been absorbed into the constant. Equation (3.14) is sometimes referred to as the variational free energy $\mathcal{F}$.

Next, we recognize that (3.14) is the negative KL divergence between the approximate posterior $q(\mathbf{Z}_j)$ and $\widetilde{p}(\mathbf{X}, \mathbf{Z}_j)$, which is simply the true joint posterior averaged over variational distributions of all latent variables and parameters except $\mathbf{Z}_j$. Minimizing this KL divergence is equivalent to maximizing the the lower bound $\mathcal{L}(q)$ with respect to $q_j$, and by inspection of (3.14), we find that:

$$\ln q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Z})] + \text{const.} \tag{3.15}$$

Equation (3.15) is the defining equation for variational inference.

By separating $\mathbf{Z}$ back out into latent variables and parameters, one can think of VB inference as being composed of an E step and an M step. For a graphical illustration of VB inference for our model, please see figure 3.1. In the VB-E step, we observe the data and take the expectation of the joint posterior with respect to all parameters, which gives us a term proportional (in the log-domain) to the distribution of the latents. The parameters and data are highlighted in yellow in figure 3.1a. In the VB-M step, we observe the data and take the expectation of the joint posterior with respect to all latents, which gives us a term proportional (in the log-domain) to the distribution of the parameters. The latent variables and data are highlighted in yellow in figure 3.1b. As in EM, each step increases the lower bound. Convergence here is guaranteed since the lower bound is convex with respect to each of the parameters (Boyd and Vandenberghe (2004)).




(a) VB-E step. (b) VB-M step.

FIGURE 3.1: Steps of VB inference involve iteratively averaging over distributions of parameters and latent variables.

In summary, VB is an approximate inference scheme in the sense that one iteratively optimizes the lower bound on the marginal likelihood of the observations by minimizing the Kullback-Leibler divergence between the approximate and true posterior distributions. There are two main steps in VB: the expectation step (VB-E) and the maximization step (VB-M). Since the variational distribution over the latents is found in the VB-E step, taking the expectation over the latents is straightforward when the distribution of the latents has some parametric form. One of the main advantages VB has over sampling-based methods is speed. Approximate posterior distributions are calculated for each of the model parameters through VB. Derivations for the VB-E and VB-M steps are provided in the following sections.

## 3.4 VB M-step

In the M-step, we take the expectation of the log joint distribution over parameters, latent variables, and data with respect to all latent variables and parameters except the one we are considering. We first write the variational posterior as a product of

marginal distributions over which it factorizes:

$$q^*(\{\mathbf{a}_k, b_k, \rho_k\}_{1:K}, \mathbf{W}, \{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\}_{1:K}, \boldsymbol{\pi}) =$$

$$q^*(\boldsymbol{\pi})q^*(\mathbf{W})\prod_{k=1}^{K}\left[q^*(\mathbf{a}_k, b_k|\rho_k)q^*(\rho_k)q^*(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k)q^*(\boldsymbol{\Lambda}_k)\right] \tag{3.16}$$

Next, we identify hyperparameters of the variational posteriors using the defining equation:

$$\ln q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}\left[\ln p(\mathbf{X}, \mathbf{Z})\right] + \text{const.} \tag{3.17}$$

where $\mathbf{Z}$ is the set of all latent variables and parameters and $\mathbf{X}$ is the set of data.

We begin with the distribution $q^*(\boldsymbol{\pi}_k)$:

$$\begin{aligned}
\ln q^*(\boldsymbol{\pi}) &= \mathbb{E}_{-\boldsymbol{\pi}}\left[\ln p(\boldsymbol{\eta}, \{\mathbf{a}_k, b_k, \rho_k\}_{1:K}, \mathbf{X}, \mathbf{W}, \{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\}_{1:K}, \boldsymbol{\pi}, \mathbf{Z})\right] + \text{const.} \\
&= \mathbb{E}_{-\boldsymbol{\pi}}\left[\sum_{k=1}^{K}\ln p(\pi_k) + \sum_{t=1}^{T}\sum_{k=1}^{K}\ln p(z_{tk}|\boldsymbol{\pi})\right] \\
&= \mathbb{E}_{-\boldsymbol{\pi}}\left[\sum_{k=1}^{K}(\lambda_0 - 1)\ln\pi_k + \sum_{k=1}^{K}\sum_{t=1}^{T} z_{tk}\ln\pi_k\right] + \text{const.} \\
&= \sum_{k=1}^{K}\left(\sum_{t=1}^{T}\mathbb{E}\left[z_{tk}\right] + \lambda_0 - 1\right)\ln\pi_k + \text{const.}
\end{aligned}$$

We recognize this as a Dirichlet distribution. Thus, we take $q^*(\boldsymbol{\pi})$ to be Dirichlet$(\lambda_1, \ldots, \lambda_K)$. By inspection, we determine the hyperparameters of $\ln q^*(\boldsymbol{\pi})$ to be:

$$\lambda_k = \lambda_0 + \sum_{t=1}^{T}\mathbb{E}[z_{tk}] \tag{3.18}$$

Next, we consider the joint distribution $q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$:

$$\begin{aligned}
\ln q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) &= \mathbb{E}_{-\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k}\left[\ln p(\boldsymbol{\eta}, \{\mathbf{a}_k, b_k, \rho_k\}_{1:K}, \mathbf{X}, \mathbf{W}, \{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\}_{1:K}, \boldsymbol{\pi}, \mathbf{Z})\right] \\
&\quad + \text{const.}
\end{aligned}$$

$$= \mathbb{E}_{-\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}\left[\ln p(\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k) + \sum_{t=1}^{T}\ln p(\mathbf{W}\mathbf{x}_t|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k,z_{tk})\right] + \text{const.}$$

$$= \mathbb{E}_{-\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}\Bigg[\ln p(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k) + \ln p(\boldsymbol{\Lambda}_k) + \sum_{t=1}^{T}\ln p(\mathbf{W}\mathbf{x}_t|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k,z_{tk})\Bigg] + \text{const.}$$

$$= \mathbb{E}_{-\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}\Bigg\{\frac{1}{2}\ln|\boldsymbol{\Lambda}_k| - \frac{\kappa_0}{2}(\boldsymbol{\mu}_k-\mathbf{m}_0)^\top\boldsymbol{\Lambda}_k(\boldsymbol{\mu}_k-\mathbf{m}_0) + \frac{\nu_0-d-1}{2}\ln|\boldsymbol{\Lambda}_k| - \frac{1}{2}\operatorname{Tr}\left(\mathbf{P}_0^{-1}\boldsymbol{\Lambda}_k\right) + \frac{1}{2}\sum_{t=1}^{T}\left[\ln|\boldsymbol{\Lambda}_k| - \frac{1}{2}\left(\mathbf{W}\mathbf{x}_t-\boldsymbol{\mu}_k\right)^\top\boldsymbol{\Lambda}_k\left(\mathbf{W}\mathbf{x}_t-\boldsymbol{\mu}_k\right)\right]z_{tk}\Bigg\} + \text{const.}$$

We first solve for the hyperparameters of $\ln q^*(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k)$. Retaining only the terms that involve $\boldsymbol{\mu}_k$ and temporarily dropping $\mathbb{E}_{-\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}$ for notational convenience:

$$\ln q^*(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k) = \frac{\kappa_0}{2}\left(\boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k - 2\boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}_k\mathbf{m}_0 + \mathbf{m}_0^\top\boldsymbol{\Lambda}_k\mathbf{m}_0\right) + \frac{1}{2}\sum_{t=1}^{T}\left[\mathbf{x}_t^\top\mathbf{W}_t^\top\boldsymbol{\Lambda}_k\mathbf{W}\mathbf{x}_t - 2\boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}_k\mathbf{W}\mathbf{x}_t + \boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k\right]z_{tk} + \text{const.}$$

$$= -\frac{1}{2}\Bigg[\boldsymbol{\mu}_k^\top\left(\kappa_0 + \sum_{t=1}^{T}z_{tk}\right)\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k - 2\boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}_k\left(\kappa_0\mathbf{m}_0 + \mathbf{W}\sum_{t=1}^{T}\left(\mathbf{x}_t z_{tk}\right)\right)\Bigg] + \text{const.}$$

We realize that the distribution is quadratic in $\boldsymbol{\mu}_k$, and hence, it is Gaussian. We can set $\ln q^*(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k)$ to be Gaussian and inspect the density to identify the hyperparameters:

$$\ln q^*(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k) = \frac{1}{2}\ln|\boldsymbol{\Lambda}_k| - \frac{\kappa_k}{2}\left[(\boldsymbol{\mu}_k-\mathbf{m}_k)^\top\boldsymbol{\Lambda}_k(\boldsymbol{\mu}_k-\mathbf{m}_k)\right]$$

$$= \frac{1}{2}\ln|\mathbf{\Lambda}_k| - \frac{1}{2}\left[\boldsymbol{\mu}_k^\top \mathbf{\Lambda}_k \boldsymbol{\mu}_k \kappa_k - 2\boldsymbol{\mu}_k^\top \mathbf{\Lambda}_k \mathbf{m}_k \kappa_k + \mathbf{m}_k^\top \mathbf{\Lambda}_k \mathbf{m}_k \kappa_k\right]$$

Hence, by inspection, and after reinstating $\mathbb{E}_{-\boldsymbol{\mu}_k, \mathbf{\Lambda}_k}$:

$$\kappa_k = \kappa_0 + \sum_{t=1}^{T} \mathbb{E}[z_{tk}] \tag{3.19}$$

$$\begin{aligned}\mathbf{m}_k &= \frac{1}{\kappa_k}\left[\kappa_0 \mathbf{m}_0 + \mathbb{E}[\mathbf{W}] \sum_{t=1}^{T} \mathbf{x}_t \mathbb{E}[z_{tk}]\right] \\ &= \frac{1}{\kappa_k}\left[\kappa_0 \mathbf{m}_0 + \mathbf{M} \sum_{t=1}^{T} \mathbf{x}_t \mathbb{E}[z_{tk}]\right] \end{aligned} \tag{3.20}$$

Next, we recognize that

$$\begin{aligned}\ln q^*(\mathbf{\Lambda}_k) &= \ln q^*(\boldsymbol{\mu}_k, \mathbf{\Lambda}_k) - \ln q^*(\boldsymbol{\mu}_k|\mathbf{\Lambda}_k) \\ &= \mathbb{E}_{-\boldsymbol{\mu}_k, \mathbf{\Lambda}_k}\left[\ln p(\boldsymbol{\eta}, \{\mathbf{a}_k, \mathbf{b}_k, \rho_k\}_{1:K}, \mathbf{X}, \mathbf{W}, \{\boldsymbol{\mu}_k, \mathbf{\Lambda}_k\}_{1:K}, \boldsymbol{\pi}, \mathbf{Z})\right] - \\ &\quad \ln q^*(\boldsymbol{\mu}_k|\mathbf{\Lambda}_k)\end{aligned}$$

We drop expectations temporarily for notational convenience. Retaining terms that involve $\mathbf{\Lambda}_k$ and noting that terms involving $\boldsymbol{\mu}_k$ should cancel:

$$\begin{aligned}\ln q^*(\mathbf{\Lambda}_k) = {} & \frac{1}{2}\ln|\mathbf{\Lambda}_k| - \frac{1}{2}\left[\boldsymbol{\mu}_k^\top \mathbf{\Lambda}_k \boldsymbol{\mu}_k \kappa_0 - 2\boldsymbol{\mu}_k \mathbf{\Lambda}_k \mathbf{m}_0 \kappa_0 + \mathbf{m}_0^\top \mathbf{\Lambda}_k \mathbf{m}_0 \kappa_0\right] + \\ & \frac{\nu_0 - d - 1}{2}\ln|\mathbf{\Lambda}_k| - \frac{1}{2}\operatorname{Tr}\left[\mathbf{P}_0^{-1}\mathbf{\Lambda}_k\right] + \\ & \sum_{t=1}^{T}\left[\frac{1}{2}\ln|\mathbf{\Lambda}_k| - \frac{1}{2}(\mathbf{x}_t^\top \mathbf{W}^\top \mathbf{\Lambda}_k \mathbf{W}\mathbf{x}_t - \right. \\ & \qquad \left. 2\mathbf{x}_t^\top \mathbf{W}^\top \mathbf{\Lambda}_k \boldsymbol{\mu}_k + \boldsymbol{\mu}_k^\top \mathbf{P}\boldsymbol{\mu}_k) z_{tk}\right] - \\ & \left[\frac{1}{2}\ln|\mathbf{\Lambda}_k| - \frac{1}{2}\left(\boldsymbol{\mu}_k^\top \mathbf{\Lambda}_k \boldsymbol{\mu}_k \kappa_k - 2\boldsymbol{\mu}_k^\top \mathbf{\Lambda}_k \mathbf{m}_k \kappa_k + \mathbf{m}_k^\top \mathbf{\Lambda}_k \mathbf{m}_k \kappa_k\right)\right] \\ & + \text{const.}\end{aligned}$$

Substituting for $\kappa_k$ and $\mathbf{m}_k$ into terms from $\ln q^*(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k)$ that are linear and quadratic in $\boldsymbol{\mu}_k$ clearly shows that all terms involving $\boldsymbol{\mu}_k$ cancel, leaving:

$$= -\frac{1}{2}\mathbf{m}_0^\top \boldsymbol{\Lambda}_k \mathbf{m}_0 \kappa_0 + \frac{\nu_0 - d - 1}{2} \ln|\boldsymbol{\Lambda}_k| - \frac{1}{2}\operatorname{Tr}\left[\mathbf{P}_0^{-1}\boldsymbol{\Lambda}_k\right] +$$

$$\frac{1}{2}\ln|\boldsymbol{\Lambda}_k| \sum_{t=1}^{T} z_{tk} - \frac{1}{2}\sum_{t=1}^{T}\left[ z_{tk}\mathbf{x}_t^\top \mathbf{W}^\top \boldsymbol{\Lambda}_k \mathbf{W}\mathbf{x}_t + \frac{1}{2}\mathbf{m}_k^\top \boldsymbol{\Lambda}_k \mathbf{m}_k \kappa_k \right]$$

Applying the trace property $\operatorname{tr}[\mathbf{ABC}] = \operatorname{tr}[\mathbf{BCA}] = \operatorname{tr}[\mathbf{CAB}]$ and collecting terms:

$$= \ln|\boldsymbol{\Lambda}_k| \left[ \frac{\nu_0 - d - 1 + \sum_{t=1}^{T} z_{tk}}{2} \right] -$$

$$\frac{1}{2}\operatorname{Tr}\left[ \left( \mathbf{P}_0^{-1} + \mathbf{m}_0\mathbf{m}_0^\top \kappa_0 - \mathbf{m}_k\mathbf{m}_k^\top \kappa_k + \right.\right.$$

$$\left.\left. \mathbf{W}\left( \sum_{t=1}^{T} \mathbf{x}_t\mathbf{x}_t^\top z_{tk} \right)\mathbf{W}^\top \right)\boldsymbol{\Lambda}_k \right]$$

We realize that the distribution is a Wishart. Hence, we can set $\ln q^*(\boldsymbol{\Lambda}_k)$ to be Wishart$(\mathbf{P}_k, \nu_k)$ and inspect the density to identify the hyperparameters. Reinstating expectations, we find:

$$\nu_k = \nu_0 + \sum_{t=1}^{T} \mathbb{E}[z_{tk}] \tag{3.21}$$

$$\mathbf{P}_k = \left\{ \mathbf{P}_0^{-1} + \mathbf{m}_0\mathbf{m}_0^\top \kappa_0 - \mathbf{m}_k\mathbf{m}_k^\top \kappa_k + \mathbb{E}\left[ \mathbf{W}\left( \sum_{t=1}^{T} \mathbf{x}_t\mathbf{x}_t^\top \mathbb{E}[z_{tk}] \right)\mathbf{W}^\top \right] \right\}^{-1}$$

$$= \left\{ \mathbf{P}_0^{-1} + \mathbf{m}_0\mathbf{m}_0^\top \kappa_0 - \mathbf{m}_k\mathbf{m}_k^\top \kappa_k + \mathbf{U}\operatorname{Tr}\left[ \left( \sum_{t=1}^{T} \mathbf{x}_t\mathbf{x}_t^\top \mathbb{E}[z_{tk}] \right)\mathbf{V} \right] + \right.$$

$$\left. \mathbf{M}\left( \sum_{t=1}^{T} \mathbf{x}_t\mathbf{x}_t^\top \mathbb{E}[z_{tk}] \right)\mathbf{M}^\top \right\}^{-1} \tag{3.22}$$

Next, we consider the joint distribution $q^*(\mathbf{a}_k, b_k, \rho_k)$. We first define the following for convenience:

$$\boldsymbol{\gamma}_k = \begin{bmatrix} \mathbf{a}_k \\ b_k \end{bmatrix} \qquad \mathbf{g}_0 = \begin{bmatrix} \mathbf{a}_0 \\ b_0 \end{bmatrix} \qquad \widetilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

$$\widetilde{\mathbf{W}} = \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \qquad \widetilde{\mathbf{M}} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \qquad \widetilde{\mathbf{U}} = \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}$$

$$\widetilde{\mathbf{V}} = \begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}$$

We proceed with the joint distribution $q^*(\boldsymbol{\gamma}_k, \rho_k)$:

$$\ln q^*(\boldsymbol{\gamma}_k, \rho_k) =$$

$$= \mathbb{E}_{-\boldsymbol{\gamma}_k, \rho_k} \left[ \ln p(\boldsymbol{\eta}, \{\boldsymbol{\gamma}_k, \rho_k\}_{1:K}, \mathbf{X}, \mathbf{W}, \{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\}_{1:K}, \boldsymbol{\pi}, \mathbf{Z}) \right] + \text{const.}$$

$$= \mathbb{E}_{-\boldsymbol{\gamma}_k, \rho_k} \left[ \sum_{t=1}^{T} \ln p(\eta_t | \boldsymbol{\gamma}_k, \rho_k, \mathbf{W}, z_{tk}) + \ln p(\boldsymbol{\gamma}_k, \rho_k) \right]$$

$$= \mathbb{E}_{-\boldsymbol{\gamma}_k, \rho_k} \left[ \sum_{t=1}^{T} \ln p(\eta_t | \boldsymbol{\gamma}_k, \rho_k, \mathbf{W}, z_{tk}) + \ln p(\boldsymbol{\gamma}_k | \rho_k) + \ln p(\rho_k) \right]$$

$$= \mathbb{E}_{-\boldsymbol{\gamma}_k, \rho_k} \left\{ \sum_{t=1}^{T} \left[ \frac{1}{2} \ln \rho_k - \frac{\rho_k}{2} \left( \eta_t^2 - 2\boldsymbol{\gamma}_k^\top \widetilde{\mathbf{W}} \widetilde{\mathbf{x}}_t \eta_t + \boldsymbol{\gamma}_k^\top \widetilde{\mathbf{W}} \widetilde{\mathbf{x}}_t \widetilde{\mathbf{x}}_t^\top \widetilde{\mathbf{W}}^\top \boldsymbol{\gamma}_k \right) \right] z_{tk} + \right.$$

$$\frac{d}{2} \ln \rho_k - \frac{\rho_k}{2} \left( \boldsymbol{\gamma}_k^\top \mathbf{B}_0 \boldsymbol{\gamma}_k - 2\boldsymbol{\gamma}_k^\top \mathbf{B}_0 \mathbf{g}_0 + \mathbf{g}_0^\top \mathbf{B}_0 \mathbf{g}_0 \right) +$$

$$\left. (\alpha_0 - 1) \ln \rho_k - \beta_0 \rho_k \right\} + \text{const.}$$

We first solve for the hyperparameters of $\ln q^*(\boldsymbol{\gamma}_k | \rho_k)$. Retaining only terms that involve $\boldsymbol{\gamma}_k$ and temporarily dropping $\mathbb{E}_{-\boldsymbol{\gamma}_k, \rho_k}$ for notational convenience:

$$\ln q^*(\boldsymbol{\gamma}_k | \rho_k) = \sum_{t=1}^{T} \left[ -\frac{1}{2} \left( \boldsymbol{\gamma}_k^\top \widetilde{\mathbf{W}} \widetilde{\mathbf{x}}_t \widetilde{\mathbf{x}}_t^\top \widetilde{\mathbf{W}}^\top \boldsymbol{\gamma}_k \rho_k - 2\boldsymbol{\gamma}_k^\top \widetilde{\mathbf{W}} \widetilde{\mathbf{x}}_t \eta_t \rho_k \right) z_{tk} \right] -$$

$$\frac{1}{2} \left[ \boldsymbol{\gamma}_k^\top \mathbf{B}_0 \boldsymbol{\gamma}_k \rho_k - 2\boldsymbol{\gamma}_k^\top \mathbf{B}_0 \mathbf{g}_0 \rho_k \right] + \text{const.}$$

We realize that the distribution is quadratic in $\boldsymbol{\gamma}_k$, and hence, it is Gaussian. We

can set $\ln q^*(\boldsymbol{\gamma}_k|\rho_k)$ to be Gaussian and inspect the density to identify the hyperparameters:

$$\begin{aligned}\ln q^*(\boldsymbol{\gamma}_k|\rho_k) &= \frac{1}{2}\ln|\rho_k\mathbf{B}_0| - \frac{\rho_k}{2}\left[(\boldsymbol{\gamma}_k-\mathbf{g}_k)^\top\mathbf{B}_k(\boldsymbol{\gamma}_k-\mathbf{g}_k)\right]\\ &= \frac{d+1}{2}\ln|\rho_k| + \frac{1}{2}\ln|\mathbf{B}_0| - \\ &\quad \frac{1}{2}\left[\boldsymbol{\gamma}_k^\top\mathbf{B}_k\boldsymbol{\gamma}_k\rho_k - 2\boldsymbol{\gamma}_k^\top\mathbf{B}_k\mathbf{g}_k\rho_k + \mathbf{g}_k^\top\mathbf{B}_k\mathbf{g}_k^\top\rho_k\right]\end{aligned}$$

Hence, by inspection, and after reinstating $\mathbb{E}_{-\boldsymbol{\gamma}_k,\rho_k}$:

$$\begin{aligned}\mathbf{B}_k &= \mathbb{E}\left[\widetilde{\mathbf{W}}\left(\sum_{t=1}^T\widetilde{\mathbf{x}}_t\widetilde{\mathbf{x}}_t^\top z_{tk}\right)\widetilde{\mathbf{W}}^\top\right] + \mathbf{B}_0\\ &= \widetilde{\mathbf{U}}\,\mathrm{Tr}\left[\left(\sum_{t=1}^T\widetilde{\mathbf{x}}_t\widetilde{\mathbf{x}}_t^\top\mathbb{E}[z_{tk}]\right)\widetilde{\mathbf{V}}\right] + \widetilde{\mathbf{M}}\left(\sum_{t=1}^T\widetilde{\mathbf{x}}_t\widetilde{\mathbf{x}}_t^\top\mathbb{E}[z_{tk}]\right)\widetilde{\mathbf{M}}^\top + \mathbf{B}_0 \qquad (3.23)\\ \mathbf{g}_k &= \mathbf{B}_k^{-1}\mathbb{E}\left[\widetilde{\mathbf{W}}\left(\sum_{t=1}^T\widetilde{\mathbf{x}}_t\eta_t z_{tk}\right) + \mathbf{B}_0\mathbf{g}_0\right]\\ &= \mathbf{B}_k^{-1}\left[\widetilde{\mathbf{M}}\left(\sum_{t=1}^T\widetilde{\mathbf{x}}_t\eta_t\mathbb{E}[z_{tk}]\right) + \mathbf{B}_0\mathbf{g}_0\right] \qquad (3.24)\end{aligned}$$

Next, we recognize that

$$\begin{aligned}\ln q^*(\rho_k) &= \ln q^*(\boldsymbol{\gamma}_k,\rho_k) - \ln q^*(\boldsymbol{\gamma}_k|\rho_k)\\ &= \mathbb{E}_{-\boldsymbol{\gamma}_k,\rho_k}\left[\ln p(\boldsymbol{\eta},\{\boldsymbol{\gamma}_k,\rho_k\}_{1:K},\mathbf{X},\mathbf{W},\{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k\}_{1:K},\boldsymbol{\pi},\mathbf{Z})\right] - \\ &\quad \ln q^*(\boldsymbol{\gamma}_k|\rho_k)\end{aligned}$$

We drop expectations temporarily for notational convenience. Retaining terms that involve $\rho_k$ and noting that terms involving $\boldsymbol{\gamma}_k$ should cancel:

$$\begin{aligned}\ln q^*(\rho_k) = \frac{1}{2}\Bigg[&\ln\rho_k\sum_{t=1}^T z_{tk} - \\ &\rho_k\Bigg[\sum_{t=1}^T\eta_t^2 z_{tk} - 2\boldsymbol{\gamma}_k^\top\widetilde{\mathbf{W}}\sum_{t=1}^T\widetilde{\mathbf{x}}_t\eta_t z_{tk} +\end{aligned}$$

$$\boldsymbol{\gamma}_k^\top \widetilde{\mathbf{W}} \left( \sum_{t=1}^{T} \widetilde{\mathbf{x}}_t \widetilde{\mathbf{x}}_t^\top z_{tk} \right) \widetilde{\mathbf{W}}^\top \boldsymbol{\gamma}_k \Bigg] \Bigg] +$$

$$\frac{1}{2} \left[ (d+1) \ln \rho_k - \rho_k \left( \boldsymbol{\gamma}_k^\top \mathbf{B}_0 \boldsymbol{\gamma}_k - 2\boldsymbol{\gamma}_k^\top \mathbf{B}_0 \mathbf{g}_0 + \mathbf{g}_0^\top \mathbf{B}_0 \mathbf{g}_0 \right) \right] +$$

$$(\alpha_0 - 1) \ln \rho_k - \beta_0 \rho_k -$$

$$\left[ \frac{d}{2} \ln |\rho_k| - \frac{\rho_k}{2} \left( \boldsymbol{\gamma}_k^\top \mathbf{B}_k \boldsymbol{\gamma}_k - 2\boldsymbol{\gamma}_k^\top \mathbf{B}_k \mathbf{g}_k + \mathbf{g}_k^\top \mathbf{B}_k \mathbf{g}_k \right) \right] + \text{const.}$$

Substituting for $\mathbf{B}_k$ into terms from $\ln q^*(\boldsymbol{\gamma}_k | \rho_k)$ that are linear and quadratic in $\boldsymbol{\gamma}_k$ clearly shows that all terms involving $\boldsymbol{\gamma}_k$ cancel, leaving:

$$= \ln \rho_k \sum_{t=1}^{T} z_{tk} - \frac{\rho_k}{2} \sum_{t=1}^{T} \eta_t^2 z_{tk} - \frac{\rho_k}{2} \mathbf{g}_0^\top \mathbf{B}_0 \mathbf{g}_0 + \frac{\rho_k}{2} \mathbf{g}_k^\top \mathbf{B}_k \mathbf{g}_k +$$

$$(\alpha_0 - 1) \ln \rho_k - \beta_0 \rho_k$$

Collecting terms:

$$= \ln \rho_k \left[ \frac{1}{2} \sum_{t=1}^{T} z_{tk} + \alpha_0 - 1 \right] -$$

$$\rho_k \left[ \frac{1}{2} \left[ \sum_{t=1}^{T} \eta_t^2 z_{tk} + \mathbf{g}_0^\top \mathbf{B}_0 \mathbf{g}_0 - \mathbf{g}_k^\top \mathbf{B}_k \mathbf{g}_k \right] + \beta_0 \right]$$

We realize this is a Gamma distribution. Hence, we can set $\ln q^*(\rho_k)$ to be Gamma$(\alpha_k, \beta_k)$ and inspect the density to identify the hyperparameters. Reinstating expectations, we find:

$$\alpha_k = \alpha_0 + \frac{1}{2} \sum_{t=1}^{T} \mathbb{E}[z_{tk}] \tag{3.25}$$

$$\beta_k = \beta_0 + \frac{1}{2} \left[ \sum_{t=1}^{T} \eta_t^2 \mathbb{E}[z_{tk}] + \mathbf{g}_0^\top \mathbf{B}_0 \mathbf{g}_0 - \mathbf{g}_k^\top \mathbf{B}_k \mathbf{g}_k \right] \tag{3.26}$$

Next, we solve for the variational posterior on $\mathbf{W}$, which we assume follows a matrix normal distribution. To compute the hyperparameters of the posterior, we can maximize the free energy (or variational lower bound) over $\mathbf{M}$, $\mathbf{U}$, and $\mathbf{V}$. The

free energy is defined as the expectation of the log joint probability over parameters and data minus the entropy of the log posterior distribution of $\mathbf{W}$:

$$\begin{aligned}
\mathcal{F} = \mathbb{E}\Bigg[ & \sum_{t=1}^{T}\sum_{k=1}^{K} \ln p(\eta_t|\boldsymbol{\gamma}_k, \rho_k, \mathbf{W}, \mathbf{x}_t, z_{tk}) + \sum_{t=1}^{T}\sum_{k=1}^{K} \ln p(\mathbf{W}\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k, z_{tk}) + \\
& \ln p(\mathbf{W}) - \ln q^*(\mathbf{W}) \Bigg] + \text{const.} \\
= \mathbb{E}\Bigg\{ & \sum_{t=1}^{T}\sum_{k=1}^{K} \left[ -\frac{\rho_k}{2}(\eta_t - \boldsymbol{\gamma}_k^\top \widetilde{\mathbf{W}}^\top \tilde{\mathbf{x}}_t^\top)^2 z_{tk} \right] + \\
& \sum_{t=1}^{T}\sum_{k=1}^{K} \left[ -\frac{1}{2} \left[ (\mathbf{W}\mathbf{x}_t - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{W}\mathbf{x}_t - \boldsymbol{\mu}_k) \right] z_{tk} \right] + \\
& - \frac{\xi_0}{2} \operatorname{Tr}\left[ \mathbf{V}_0^{-1}(\mathbf{W} - \mathbf{M}_0)^\top \mathbf{U}_0^{-1}(\mathbf{W} - \mathbf{M}_0) \right] - \\
& \left[ -\frac{d}{2}\ln|\mathbf{V}| - \frac{p}{2}\ln|\mathbf{U}| - \frac{1}{2}\operatorname{Tr}\left[ \mathbf{V}^{-1}(\mathbf{W} - \mathbf{M})^\top \mathbf{U}^{-1}(\mathbf{W} - \mathbf{M}) \right] \right] \Bigg\} \\
& + \text{const.}
\end{aligned} \tag{3.27}$$

We can take this expectation term by term. Note that in all the following expectations, we have dropped some terms that are constant with respect to $\mathbf{W}$, $\mathbf{U}$, and $\mathbf{V}$. We begin with the $\eta_t$ observation term:

$$\mathbb{E}\left[ -\frac{\rho_k}{2}(\eta_t - \boldsymbol{\gamma}_k^\top \widetilde{\mathbf{W}}^\top \tilde{\mathbf{x}}_t^\top)^2 z_{tk} \right] =$$

$$\mathbb{E}\left\{ -\frac{\rho_k}{2}\left( -2\mathbf{a}_k^\top \mathbf{W}\mathbf{x}_t\eta_t + \mathbf{a}_k^\top \mathbf{W}\mathbf{x}_t\mathbf{x}_t^\top \mathbf{W}^\top \mathbf{a}_k + 2\mathbf{a}_k^\top \mathbf{W}\mathbf{x}_t b_k \right) z_{tk} \right\}$$

We can break this term down into subterms. We begin with the first subterm:

$$\begin{aligned}
\mathbb{E}\left[ -\frac{\rho_k}{2}\left( -2\mathbf{a}_k^\top \mathbf{W}\mathbf{x}_t\eta_t \right) \right] &= \mathbb{E}_{\mathbf{W}}[\mathbb{E}_{\rho_k}[\mathbb{E}_{\mathbf{a}_k|\rho_k}[\left( -\frac{\rho_k}{2} \right)\left( -2\mathbf{a}_k^\top \mathbf{W}\mathbf{x}_t\eta_t \right)]]] \\
&= \mathbb{E}_{\mathbf{W}}[\mathbb{E}_{\rho_k}[\rho_k(\mathbf{g}_{1:d}^{(k)\top}\mathbf{W}\mathbf{x}_t\eta_t)]] \\
&= \mathbb{E}_{\mathbf{W}}\left[ \frac{\alpha_k}{\beta_k}(\mathbf{g}_{1:d}^{(k)\top}\mathbf{W}\mathbf{x}_t\eta_t) \right]
\end{aligned}$$

$$= \frac{\alpha_k}{\beta_k}(\mathbf{g}_{1:d}^{(k)\top}\mathbf{M}\mathbf{x}_t\eta_t)$$

Next, the second subterm. For this subterm, we can apply the trace property:

$$\begin{aligned}
\mathbb{E}\left[-\frac{\rho_k}{2}\mathbf{a}_k^\top\mathbf{W}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{W}^\top\mathbf{a}_k\right] &= \mathbb{E}_{\mathbf{W}}\left[\mathbb{E}_{\rho_k}\left[\mathbb{E}_{\mathbf{a}_k|\rho_k}\left[\mathrm{Tr}\left(-\frac{\rho_k}{2}\mathbf{a}_k\mathbf{a}_k^\top\mathbf{W}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{W}^\top\right)\right]\right]\right] \\
&= \mathbb{E}_{\mathbf{W}}\Big[\mathbb{E}_{\rho_k}\Big[\,\mathrm{Tr}\,\big(\,-\frac{\rho_k}{2}\Big[\Big[\mathbf{B}_k^{-1}\Big]_{1:d,1:d}\rho_k^{-1}+ \\
&\qquad \mathbf{g}_{1:d}^{(k)}\mathbf{g}_{1:d}^{(k)\top}\Big]\mathbf{W}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{W}^\top\Big)\Big]\Big] \\
&= \mathbb{E}_{\mathbf{W}}\Big[\,\mathrm{Tr}\,\Big(\,-\frac{1}{2}\Big[\mathbf{B}_k^{-1}\Big]_{1:d,1:d}\mathbf{W}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{W}^\top- \\
&\qquad \frac{1}{2}\frac{\alpha_k}{\beta_k}\mathbf{g}_{1:d}^{(k)}\mathbf{g}_{1:d}^{(k)\top}\mathbf{W}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{W}^\top\Big)\Big] \\
&= -\frac{1}{2}\,\mathrm{Tr}\left(\mathbf{C}_k\left[\mathbf{U}\,\mathrm{Tr}\left(\mathbf{x}_t\mathbf{x}_t^\top\mathbf{V}\right)+\mathbf{M}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{M}^\top\right]\right)
\end{aligned}$$

where $\mathbf{C}_k := [\mathbf{B}_k^{-1}]_{1:d,1:d} + \frac{\alpha_k}{\beta_k}\mathbf{g}_{1:d}^{(k)}\mathbf{g}_{1:d}^{(k)\top}$. Finally, for the last subterm:

$$\begin{aligned}
\mathbb{E}\left[-\frac{\rho_k}{2}(2\mathbf{a}_k^\top\mathbf{W}\mathbf{x}_t b_k)\right] &= \mathbb{E}_{\mathbf{W}}\left[\mathbb{E}_{\rho_k}\left[\mathbb{E}_{\mathbf{a}_k,b_k|\rho_k}\left[-\frac{\rho_k}{2}(2\mathbf{a}_k^\top b_k\mathbf{W}\mathbf{x}_t)\right]\right]\right] \\
&= \mathbb{E}_{\mathbf{W}}\left[\mathbb{E}_{\rho_k}\left[-\rho_k\left[\left[\mathbf{B}_k^{-1}\right]_{d+1,1:d}\rho_k^{-1}+g_{d+1}^{(k)}\mathbf{g}_{1:d}^{(k)\top}\right]\mathbf{W}\mathbf{x}_t\right]\right] \\
&= \mathbb{E}_{\mathbf{W}}\left[-\left[\left[\mathbf{B}_k^{-1}\right]_{d+1,1:d}+\frac{\alpha_k}{\beta_k}g_{d+1}^{(k)}\mathbf{g}_{1:d}^{(k)\top}\right]\mathbf{W}\mathbf{x}_t\right] \\
&= -\left(\left[\mathbf{B}_k^{-1}\right]_{d+1,1:d}+\frac{\alpha_k}{\beta_k}g_{d+1}^{(k)}\mathbf{g}_{1:d}^{(k)\top}\right)\mathbf{M}\mathbf{x}_t
\end{aligned}$$

Adding these subterms and taking the expectation with resepect to $z_{tk}$:

$$\begin{aligned}
\mathbb{E}\left[-\frac{\rho_k}{2}(\eta_t-\boldsymbol{\gamma}_k^\top\widetilde{\mathbf{W}}^\top\widetilde{\mathbf{x}}_t^\top)^2 z_{tk}\right] = \Big\{&\frac{\alpha_k}{\beta_k}(\mathbf{g}_{1:d}^{(k)\top}\mathbf{M}\mathbf{x}_t\eta_t)- \\
&\frac{1}{2}\,\mathrm{Tr}\left(\mathbf{C}_k\left[\mathbf{U}\,\mathrm{Tr}\left(\mathbf{x}_t\mathbf{x}_t^\top\mathbf{V}\right)+\mathbf{M}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{M}^\top\right]\right)- \\
&\left(\left[\mathbf{B}_k^{-1}\right]_{d+1,1:d}+\frac{\alpha_k}{\beta_k}g_{d+1}^{(k)}\mathbf{g}_{1:d}^{(k)\top}\right)\mathbf{M}\mathbf{x}_t\Big\}\mathbb{E}[z_{tk}] \quad (3.28)
\end{aligned}$$

Next, we deal with the $\mathbf{W}\mathbf{x}_t$ term:

$$\mathbb{E}\Bigg[ -\frac{1}{2}\Bigg[ \left(\mathbf{W}\mathbf{x}_t - \boldsymbol{\mu}_k\right)^\top \boldsymbol{\Lambda}_k \left(\mathbf{W}\mathbf{x}_t - \boldsymbol{\mu}_k\right) \Bigg] z_{tk} \Bigg]$$

$$= -\mathbb{E}\Bigg[ \frac{1}{2}(\mathbf{x}_t^\top \mathbf{W}^\top \boldsymbol{\Lambda}_k \mathbf{W}\mathbf{x}_t - 2\mathbf{x}_t^\top \mathbf{W}^\top \boldsymbol{\Lambda}_k \boldsymbol{\mu}_k + \boldsymbol{\mu}_k^\top \boldsymbol{\Lambda}_k \boldsymbol{\mu}_k) z_{tk} \Bigg]$$

$$= -\mathbb{E}_{z_{tk}}\Bigg[ \mathbb{E}_{\mathbf{W}}\Bigg[ \mathbb{E}_{\boldsymbol{\Lambda}_k}\Bigg[ \mathbb{E}_{\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k}\Bigg[ \frac{1}{2}\mathbf{x}_t^\top \mathbf{W}^\top \boldsymbol{\Lambda}_k \mathbf{W}\mathbf{x}_t - \mathbf{x}_t^\top \mathbf{W}^\top \boldsymbol{\Lambda}_k \boldsymbol{\mu}_k + \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Lambda}_k \boldsymbol{\mu}_k \Bigg]\Bigg]\Bigg] z_{tk} \Bigg]$$

$$= -\mathbb{E}_{z_{tk}}\Bigg[ \mathbb{E}_{\mathbf{W}}\Bigg[ \mathbb{E}_{\boldsymbol{\Lambda}_k}\Bigg[ \frac{1}{2}\mathbf{x}_t^\top \mathbf{W}^\top \boldsymbol{\Lambda}_k \mathbf{W}\mathbf{x}_t - \mathbf{x}_t^\top \mathbf{W}^\top \boldsymbol{\Lambda}_k \mathbf{m}_k + \frac{1}{2}\left[\mathrm{Tr}\left(\boldsymbol{\Lambda}_k \boldsymbol{\Lambda}_k^{-1} \kappa_k^{-1}\right) + \mathbf{m}_k^\top \boldsymbol{\Lambda}_k \mathbf{m}_k\right] \Bigg]\Bigg] z_{tk} \Bigg]$$

$$= -\mathbb{E}_{z_{tk}}\Bigg[ \mathbb{E}_{\mathbf{W}}\Bigg[ \frac{1}{2}\mathbf{x}_t^\top \mathbf{W}^\top \mathbf{P}_k \nu_k \mathbf{W}\mathbf{x}_t - \mathbf{x}_t^\top \mathbf{W}^\top \mathbf{P}_k \nu_k \mathbf{m}_k + \frac{1}{2}\left[\mathrm{Tr}\left(\mathbf{I}_d \kappa_k^{-1}\right) + \mathbf{m}_k^\top \mathbf{P}_k \nu_k \mathbf{m}_k\right] \Bigg] z_{tk} \Bigg]$$

$$= -\Bigg[ \frac{1}{2}\nu_k \mathbf{x}_t^\top \left[\mathbf{V}\,\mathrm{Tr}\left(\mathbf{U}\mathbf{P}_k\right) + \mathbf{M}^\top \mathbf{P}_k \mathbf{M}\right] \mathbf{x}_t - \nu_k \mathbf{x}_t^\top \mathbf{M}^\top \mathbf{P}_k \mathbf{m}_k + \frac{1}{2}\left[d\kappa_k^{-1} + \nu_k \mathbf{m}_k^\top \mathbf{P}_k \mathbf{m}_k\right] \Bigg] \mathbb{E}\left[z_{tk}\right] \tag{3.29}$$

Next, we tackle the prior on $\mathbf{W}$:

$$\mathbb{E}\left[ -\frac{\xi_0}{2}\,\mathrm{Tr}\left[\mathbf{V}_0^{-1}(\mathbf{W} - \mathbf{M}_0)^\top \mathbf{U}_0^{-1}(\mathbf{W} - \mathbf{M}_0)\right] \right]$$

$$= \mathbb{E}_{\mathbf{W}}\Bigg[ -\frac{\xi_0}{2}\,\mathrm{Tr}[\mathbf{V}_0^{-1}\mathbf{W}^\top \mathbf{U}_0^{-1}\mathbf{W} - \mathbf{V}_0^{-1}\mathbf{M}_0^\top \mathbf{U}_0^{-1}\mathbf{W} - \mathbf{V}_0^{-1}\mathbf{W}^\top \mathbf{U}_0^{-1}\mathbf{M}_0] \Bigg] + \text{const.}$$

$$= -\frac{\xi_0}{2}\,\mathrm{Tr}\Bigg[ \mathbf{V}_0^{-1}\left[\mathbf{V}\,\mathrm{Tr}\left(\mathbf{U}\mathbf{U}_0^{-1}\right) + \mathbf{M}^\top \mathbf{U}_0^{-1}\mathbf{M}\right] - \mathbf{V}_0^{-1}\mathbf{M}_0^\top \mathbf{U}_0^{-1}\mathbf{M} -$$

$$\mathbf{V}_0^{-1}\mathbf{M}^\top\mathbf{U}_0^{-1}\mathbf{M}_0\Big] + \text{const.} \tag{3.30}$$

Finally, we do the entropy term. Here, we have omitted $-\frac{d}{2}\ln|\mathbf{V}| - \frac{p}{2}\ln|\mathbf{U}|$ for convenience:

$$\mathbb{E}_{\mathbf{W}}\left[-\frac{1}{2}\operatorname{Tr}\left[\mathbf{V}^{-1}(\mathbf{W}-\mathbf{M})^\top\mathbf{U}^{-1}(\mathbf{W}-\mathbf{M})\right]\right]$$

$$= \mathbb{E}_{\mathbf{W}}\Big[ -\frac{1}{2}\operatorname{Tr}[\mathbf{V}^{-1}\mathbf{W}^\top\mathbf{U}^{-1}\mathbf{W} -$$

$$\mathbf{V}^{-1}\mathbf{M}^\top\mathbf{U}^{-1}\mathbf{W} - \mathbf{V}^{-1}\mathbf{W}^\top\mathbf{U}^{-1}\mathbf{M} + \mathbf{V}^{-1}\mathbf{M}^\top\mathbf{U}^{-1}\mathbf{M})]\Big]$$

$$= -\frac{1}{2}\operatorname{Tr}[\mathbf{V}^{-1}\left[\mathbf{V}\operatorname{Tr}\left(\mathbf{U}\mathbf{U}^{-1}\right) + \mathbf{M}^\top\mathbf{U}^{-1}\mathbf{M}\right] -$$

$$\mathbf{V}^{-1}\mathbf{M}^\top\mathbf{U}^{-1}\mathbf{M} - \mathbf{V}^{-1}\mathbf{M}^\top\mathbf{U}^{-1}\mathbf{M} + \mathbf{V}^{-1}\mathbf{M}^\top\mathbf{U}^{-1}\mathbf{M})]$$

$$= -\frac{dp}{2} \tag{3.31}$$

Substituting the above components into (3.27):

$$\mathcal{F} = \sum_{t=1}^{T}\sum_{k=1}^{K}\Bigg\{\frac{\alpha_k}{\beta_k}(\mathbf{g}_{1:d}^{(k)\top}\mathbf{M}\mathbf{x}_t\eta_t) -$$

$$\frac{1}{2}\operatorname{Tr}\left(\mathbf{C}_k\left[\mathbf{U}\operatorname{Tr}\left(\mathbf{x}_t\mathbf{x}_t^\top\mathbf{V}\right) + \mathbf{M}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{M}^\top\right]\right) -$$

$$\left(\left[\mathbf{B}_k^{-1}\right]_{d+1,1:d} + \frac{\alpha_k}{\beta_k}g_{d+1}^{(k)}\mathbf{g}_{1:d}^{(k)\top}\right)\mathbf{M}\mathbf{x}_t\Bigg\}\mathbb{E}[z_{tk}] +$$

$$\sum_{t=1}^{T}\sum_{k=1}^{K} -\Bigg\{\frac{1}{2}\nu_k\mathbf{x}_t^\top\left[\mathbf{V}\operatorname{Tr}\left(\mathbf{U}\mathbf{P}_k\right) + \mathbf{M}^\top\mathbf{P}_k\mathbf{M}\right]\mathbf{x}_t -$$

$$\nu_k\mathbf{x}_t^\top\mathbf{M}^\top\mathbf{P}_k\mathbf{m}_k + \frac{1}{2}\left[d\kappa_k^{-1} + \nu_k\mathbf{m}_k^\top\mathbf{P}_k\mathbf{m}_k\right]\Bigg\}\mathbb{E}\left[z_{tk}\right] -$$

$$\frac{\xi_0}{2}\operatorname{Tr}\Big[\mathbf{V}_0^{-1}\left[\mathbf{V}\operatorname{Tr}\left(\mathbf{U}\mathbf{U}_0^{-1}\right) + \mathbf{M}^\top\mathbf{U}_0^{-1}\mathbf{M}\right] - \mathbf{V}_0^{-1}\mathbf{M}_0^\top\mathbf{U}_0^{-1}\mathbf{M} -$$

$$\mathbf{V}_0^{-1}\mathbf{M}^\top\mathbf{U}_0^{-1}\mathbf{M}_0\Big] + \frac{d}{2}\ln|\mathbf{V}| + \frac{p}{2}\ln|\mathbf{U}| + \frac{dp}{2} \tag{3.32}$$

We now take derivatives of the above expression with respect to $\mathbf{M}$, $\mathbf{U}$, and $\mathbf{V}$ and set them equal to zero to solve for the hyperparameters. We begin with $\mathbf{U}$:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{U}} = \frac{\partial}{\partial \mathbf{U}} \Bigg\{ \sum_{t=1}^{T} \sum_{k=1}^{K} \Bigg[ -\frac{1}{2} \operatorname{Tr} \left\{ \mathbf{C}_k \left[ \mathbf{U} \operatorname{Tr} \left( \mathbf{x}_t \mathbf{x}_t^\top \mathbf{V} \right) \right] \mathbb{E} \left[ z_{tk} \right] \right\} - \frac{1}{2} \nu_k \mathbf{x}_t^\top \left[ \mathbf{V} \operatorname{Tr} \left( \mathbf{U} \mathbf{P}_k \right) \right] \mathbf{x}_t \mathbb{E} \left[ z_{tk} \right] \Bigg] - \frac{\xi_0}{2} \operatorname{Tr} \left[ \mathbf{V}_0^{-1} \left[ \mathbf{V} \operatorname{Tr} \left( \mathbf{U} \mathbf{U}_0^{-1} \right) \right] \right] + \frac{p}{2} \ln |\mathbf{U}| \Bigg\}$$

$$= \sum_{t=1}^{T} \sum_{k=1}^{K} \Bigg[ -\frac{1}{2} \mathbf{C}_k \operatorname{Tr} \left( \mathbf{x}_t \mathbf{x}_t^\top \mathbf{V} \right) \mathbb{E}[z_{tk}] - \frac{1}{2} \nu_k \mathbf{x}_t^\top \mathbf{V} \mathbf{x}_t \mathbf{P}_k \mathbb{E} \left[ z_{tk} \right] \Bigg] - \frac{\xi_0}{2} \operatorname{Tr} \left[ \mathbf{V}_0^{-1} \mathbf{V} \right] \mathbf{U}_0^{-1} + \frac{p}{2} \mathbf{U}^{-1} = 0$$

Thus,

$$\mathbf{U} = \frac{1}{p} \Bigg[ \sum_{t=1}^{T} \sum_{k=1}^{K} \left[ \operatorname{Tr} \left( \mathbf{x}_t \mathbf{x}_t^\top \mathbf{V} \right) \mathbb{E}[z_{tk}] \mathbf{C}_k + \nu_k \mathbf{x}_t^\top \mathbf{V} \mathbf{x}_t \mathbb{E} \left[ z_{tk} \right] \mathbf{P}_k \right] + \xi_0 \operatorname{Tr} \left[ \mathbf{V}_0^{-1} \mathbf{V} \right] \mathbf{U}_0^{-1} \Bigg]^{-1} \tag{3.33}$$

Next, we solve for $\mathbf{V}$:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{V}} = \frac{\partial}{\partial \mathbf{V}} \Bigg\{ \sum_{t=1}^{T} \sum_{k=1}^{K} \Bigg[ -\frac{1}{2} \operatorname{Tr} \left[ \mathbf{C}_k \left[ \mathbf{U} \operatorname{Tr} \left( \mathbf{x}_t \mathbf{x}_t^\top \mathbf{V} \right) \right] \right] \mathbb{E} \left[ z_{tk} \right] - \frac{1}{2} \nu_k \mathbf{x}_t^\top \mathbf{V} \operatorname{Tr} \left( \mathbf{U} \mathbf{P}_k \right) \mathbf{x}_t \mathbb{E} \left[ z_{tk} \right] \Bigg] - \frac{\xi_0}{2} \operatorname{Tr} \left[ \mathbf{V}_0^{-1} \left[ \mathbf{V} \operatorname{Tr} \left( \mathbf{U} \mathbf{U}_0^{-1} \right) \right] \right] + \frac{d}{2} \ln |\mathbf{V}| \Bigg\}$$

$$= \sum_{t=1}^{T} \sum_{k=1}^{K} \left[ -\frac{1}{2} \operatorname{Tr} \left[ \mathbf{C}_k \mathbf{U} \right] \mathbf{x}_t \mathbf{x}_t^\top \mathbb{E}[z_{tk}] - \frac{1}{2} \nu_k \mathbf{x}_t \mathbf{x}_t^\top \operatorname{Tr} \left( \mathbf{U} \mathbf{P}_k \right) \mathbb{E}[z_{tk}] \right] - \frac{\xi_0}{2} \mathbf{V}_0^{-1} \operatorname{Tr} \left( \mathbf{U} \mathbf{U}_0^{-1} \right) + \frac{d}{2} \mathbf{V}^{-1} = 0$$

Thus,

$$\mathbf{V} = \frac{1}{d}\Bigg[ \sum_{t=1}^{T}\sum_{k=1}^{K} \left[\operatorname{Tr}\left[\mathbf{C}_k\mathbf{U}\right]\mathbb{E}[z_{tk}]\mathbf{x}_t\mathbf{x}_t^\top + \nu_k \operatorname{Tr}\left(\mathbf{U}\mathbf{P}_k\right)\mathbb{E}[z_{tk}]\mathbf{x}_t\mathbf{x}_t^\top\right] + \xi_0 \operatorname{Tr}\left(\mathbf{U}\mathbf{U}_0^{-1}\right)\mathbf{V}_0^{-1}\Bigg]^{-1}$$

$$= \frac{1}{d}\Bigg[ \sum_{t=1}^{T}\sum_{k=1}^{K} \left[\left(\operatorname{Tr}\left[\mathbf{C}_k\mathbf{U}\right] + \nu_k \operatorname{Tr}\left(\mathbf{U}\mathbf{P}_k\right)\right)\mathbb{E}[z_{tk}]\mathbf{x}_t\mathbf{x}_t^\top\right] + \xi_0 \operatorname{Tr}\left(\mathbf{U}\mathbf{U}_0^{-1}\right)\mathbf{V}_0^{-1}\Bigg]^{-1} \tag{3.34}$$

Finally, we solve for $\mathbf{M}$:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{M}} = \frac{\partial}{\partial \mathbf{M}}\Bigg\{ \sum_{t=1}^{T}\sum_{k=1}^{K} \Bigg\{ \frac{\alpha_k}{\beta_k}\left(\mathbf{g}_{1:d}^{(k)\top}\mathbf{M}\mathbf{x}_t\eta_t\right) - \frac{1}{2}\operatorname{Tr}\left[\mathbf{C}_k\left[\mathbf{M}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{M}^\top\right]\right] - \left(\left[\mathbf{B}_k^{-1}\right]_{d+1,1:d} + \frac{\alpha_k}{\beta_k}g_{d+1}^{(k)}\mathbf{g}_{1:d}^{(k)\top}\right)\mathbf{M}\mathbf{x}_t\Bigg\}\mathbb{E}[z_{tk}] -$$

$$\sum_{t=1}^{T}\sum_{k=1}^{K} \Bigg\{ \frac{1}{2}\nu_k\mathbf{x}_t^\top\mathbf{M}^\top\mathbf{P}_k\mathbf{M}\mathbf{x}_t - \nu_k\mathbf{x}_t^\top\mathbf{M}^\top\mathbf{P}_k\mathbf{m}_k\Bigg\}\mathbb{E}[z_{tk}] -$$

$$\frac{\xi_0}{2}\operatorname{Tr}\left[\mathbf{V}_0^{-1}\mathbf{M}^\top\mathbf{U}_0^{-1}\mathbf{M} - \mathbf{V}_0^{-1}\mathbf{M}_0^\top\mathbf{U}_0^{-1}\mathbf{M} - \mathbf{V}_0^{-1}\mathbf{M}^\top\mathbf{U}_0^{-1}\mathbf{M}_0\right]\Bigg\} \tag{3.35}$$

We define the follwing variables as derivatives of terms in equation (3.35):

$$\mathbf{A}_1 := \frac{\partial}{\partial \mathbf{M}}\left\{\sum_{t=1}^{T}\sum_{k=1}^{K}\frac{\alpha_k}{\beta_k}\mathbf{g}_{1:d}^{(k)\top}\mathbf{M}\mathbf{x}_t\eta_t\mathbb{E}[z_{tk}]\right\} = \sum_{t=1}^{T}\sum_{k=1}^{K}\frac{\alpha_k}{\beta_k}\mathbf{g}_{1:d}^{(k)}\mathbf{x}_t^\top\eta_t\mathbb{E}[z_{tk}]$$

$$\mathbf{A}_2 := \frac{\partial}{\partial \mathbf{M}}\left\{\sum_{t=1}^{T}\sum_{k=1}^{K}\left(\left[\mathbf{B}_k^{-1}\right]_{d+1,1:d} + \frac{\alpha_k}{\beta_k}g_{d+1}^{(k)}\mathbf{g}_{1:d}^{(k)\top}\right)\mathbf{M}\mathbf{x}_t\mathbb{E}[z_{tk}]\right\}$$

$$= \sum_{t=1}^{T}\sum_{k=1}^{K}\left(\left[\mathbf{B}_k^{-1}\right]_{d+1,1:d} + \frac{\alpha_k}{\beta_k}g_{d+1}^{(k)}\mathbf{g}_{1:d}^{(k)\top}\right)^\top\mathbf{x}_t^\top\mathbb{E}[z_{tk}]$$

$$\mathbf{A}_3 := \frac{\partial}{\partial \mathbf{M}} \left\{ \sum_{t=1}^{T} \sum_{k=1}^{K} \nu_k \mathbf{x}_t^\top \mathbf{M}^\top \mathbf{P}_k \mathbf{m}_k \right\} = \sum_{t=1}^{T} \sum_{k=1}^{K} \nu_k \mathbf{P}_k \mathbf{m}_k \mathbf{x}_t^\top \mathbb{E}[z_{tk}]$$

In addition:

$$\begin{aligned} \mathbf{A}_4 := \frac{\xi_0}{2} \frac{\partial}{\partial \mathbf{M}} \operatorname{Tr}\left[\mathbf{V}_0^{-1} \mathbf{M}^\top \mathbf{U}_0^{-1} \mathbf{M}\right] &= \frac{\xi_0}{2} \frac{\partial}{\partial \mathbf{M}} \operatorname{Tr}\left[\mathbf{U}_0^{-1} \mathbf{M} \mathbf{V}_0^{-1} \mathbf{M}^\top\right] \\ &= \xi_0 \mathbf{U}_0^{-1} \mathbf{M} \mathbf{V}_0^{-1} \end{aligned} \tag{3.36}$$

$$\mathbf{A}_5 := \frac{\xi_0}{2} \frac{\partial}{\partial \mathbf{M}} \operatorname{Tr}\left[\mathbf{V}_0^{-1} \mathbf{M}_0^\top \mathbf{U}_0^{-1} \mathbf{M}\right] = \frac{\xi_0}{2} \mathbf{U}_0^{-1} \mathbf{M}_0 \mathbf{V}_0^{-1}$$

$$\mathbf{A}_6 := \frac{\xi_0}{2} \frac{\partial}{\partial \mathbf{M}} \operatorname{Tr}\left[\mathbf{V}_0^{-1} \mathbf{M}^\top \mathbf{U}_0^{-1} \mathbf{M}_0\right] = \frac{\xi_0}{2} \mathbf{U}_0^{-1} \mathbf{M}_0 \mathbf{V}_0^{-1}$$

This leaves two more quadratic terms. From matrix calculus identities:

$$\frac{\partial}{\partial \mathbf{X}} \operatorname{Tr}[\mathbf{A}\mathbf{X}\mathbf{B}\mathbf{X}^\top \mathbf{C}] = \mathbf{A}^\top \mathbf{C}^\top \mathbf{X} \mathbf{B}^\top + \mathbf{C}\mathbf{A}\mathbf{X}\mathbf{B} \tag{3.37}$$

Although the remaining quadratic terms can be written in this form, the summations are coupled over $t$ and $k$ through $\mathbb{E}[z_{tk}]$, and hence, the sum cannot be broken. We proceed as follows.

First, consider the quadratic term $\sum_{t=1}^{T} \sum_{k=1}^{K} \frac{1}{2} \operatorname{Tr}\left[\mathbf{C}_k \left(\mathbf{M}\mathbf{x}_t \mathbf{x}_t^\top \mathbf{M}^\top\right)\right] \mathbb{E}[z_{tk}]$:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{M}} \sum_{t=1}^{T} \sum_{k=1}^{K} \frac{1}{2} \operatorname{Tr}\left[\mathbf{C}_k \left(\mathbf{M}\mathbf{x}_t \mathbf{x}_t^\top \mathbf{M}^\top\right)\right] \mathbb{E}[z_{tk}] &= \sum_{t=1}^{T} \sum_{k=1}^{K} \frac{1}{2} \left(\mathbf{C}_k \mathbf{M} \mathbf{x}_t \mathbf{x}_t^\top\right) \mathbb{E}[z_{tk}] \quad (2) \\ &= \sum_{t=1}^{T} \sum_{k=1}^{K} \left(\mathbf{C}_k \mathbf{M} \mathbf{x}_t \mathbf{x}_t^\top\right) \mathbb{E}[z_{tk}] =: \mathbf{K}^{(1)} \end{aligned}$$

where we have defined $\mathbf{K}^{(1)}$ to be the result of the matrix derivative. Rewriting this in elementwise notation gives:

$$K_{ij}^{(1)} = \sum_{t=1}^{T} \sum_{k=1}^{K} \mathbb{E}[z_{tk}] \left[ \sum_{n=1}^{p} \sum_{l=1}^{d} C_{il}^{(k)} M_{ln} x_n^{(t)} \right] x_j^{(t)} \tag{3.38}$$

where $C_{il}^{(k)}$ denotes the entry at row $i$ and column $l$ of $\mathbf{C}_k$, $M_{ln}$ denotes the entry at row $l$ and column $n$ of $\mathbf{M}$, and $x_n^{(t)}$ denotes the $n^{\text{th}}$ entry of $\mathbf{x}_t$. $\mathbf{K}^{(1)}$ is a $d \times p$ matrix, so $i = 1, \ldots, d$ and $j = 1, \ldots, p$.

Similarly, for the second quadratic term $\frac{1}{2}\sum_{t=1}^{T}\sum_{k=1}^{K}\nu_k \mathbf{x}_t^\top \mathbf{M}^\top \mathbf{P}_k \mathbf{M}\mathbf{x}_t \mathbb{E}[z_{tk}]$:

$$\frac{\partial}{\partial \mathbf{M}} \sum_{t=1}^{T}\sum_{k=1}^{K} \frac{1}{2}\nu_k \mathbf{x}_t^\top \mathbf{M}^\top \mathbf{P}_k \mathbf{M}\mathbf{x}_t \mathbb{E}[z_{tk}]$$

$$= \frac{\partial}{\partial \mathbf{M}} \sum_{t=1}^{T}\sum_{k=1}^{K} \frac{1}{2} \operatorname{Tr}\left[\mathbf{P}_k \mathbf{M}\mathbf{x}_t \mathbf{x}_t^\top \mathbf{M}^\top \nu_k \mathbb{E}[z_{tk}]\right]$$

$$= \frac{1}{2}\mathbf{P}_k \mathbf{M}\mathbf{x}_t \mathbf{x}_t^\top \nu_k \mathbb{E}[z_{tk}](2)$$

$$= \mathbf{P}_k \mathbf{M}\mathbf{x}_t \mathbf{x}_t^\top \nu_k \mathbb{E}[z_{tk}] =: \mathbf{K}^{(2)}$$

Rewriting this in elementwise notation gives:

$$K_{ij}^{(2)} = \sum_{t=1}^{T}\sum_{k=1}^{K} \mathbb{E}[z_{tk}]\nu_k \left[\sum_{n=1}^{p}\sum_{l=1}^{d} P_{il}^{(k)} M_{ln} x_n^{(t)}\right] x_j^{(t)}$$

where $P_{il}^{(k)}$ denotes the entry at row $i$ and column $l$ of $\mathbf{P}_k$.

These derivatives are fourth order tensors; yet, we can still solve for $\mathbf{M}$ by a matrix inversion. First, we notice the term from equation (3.36) whose derivative involves $\mathbf{M}$. We write this term in elementwise notation:

$$K_{ij}^{(3)} = \xi_0 \sum_{n=1}^{p}\sum_{l=1}^{d} (\mathbf{U}_0^{-1})_{il} M_{ln} (\mathbf{V}_0^{-1})_{nj}$$

where $(\mathbf{U}_0^{-1})_{il}$ denotes the entry at row $i$ and column $l$ of $\mathbf{U}_0^{-1}$ (similarly for $\mathbf{V}_0^{-1}$).

The full expression of the derivative can be set equal to zero to solve for $M_{ln}$:

$$A_{ij}^{(1)} - A_{ij}^{(2)} + A_{ij}^{(3)} - K_{ij}^{(3)} + A_{ij}^{(5)} + A_{ij}^{(6)} - K_{ij}^{(1)} - K_{ij}^{(2)} = 0$$

where $A_{ij}^{(q)}$ denotes the element at row $i$ and column $j$ of $\mathbf{A}_q$. Isolating $K_{ij}^{(1)}$, $K_{ij}^{(2)}$, and $K_{ij}^{(3)}$, and substituting to view the equation in terms of $W_{ln}$:

$$A_{ij}^{(1)} - A_{ij}^{(2)} + A_{ij}^{(3)} + A_{ij}^{(5)} + A_{ij}^{(6)} = K_{ij}^{(1)} + K_{ij}^{(2)} + K_{ij}^{(3)}$$

$$= \sum_{t=1}^{T}\sum_{k=1}^{K} \mathbb{E}[z_{tk}] \left[\sum_{n=1}^{p}\sum_{l=1}^{d} C_{il}^{(k)} M_{ln} x_n^{(t)}\right] x_j^{(t)} +$$

$$\sum_{t=1}^{T}\sum_{k=1}^{K}\mathbb{E}[z_{tk}]\nu_k\left[\sum_{n=1}^{p}\sum_{l=1}^{d}P_{il}^{(k)}M_{ln}x_n^{(t)}\right]x_j^{(t)}+$$

$$\xi_0\sum_{n=1}^{p}\sum_{l=1}^{d}(\mathbf{U}_0^{-1})_{il}M_{ln}(\mathbf{V}_0^{-1})_{nj} \tag{3.39}$$

This can be rewritten in matrix notation as

$$\mathrm{vec}(\mathbf{D}) = \mathbf{H}\mathrm{vec}(\mathbf{M})$$

where $\mathbf{D}$ denotes the matrix defined elementwise by the left-hand side of equation (3.39); the $\mathrm{vec}(\cdot)$ operation vectorizes its argument matrix by stacking its columns; and $\mathbf{H}$ is a $dp \times dp$ matrix defined elementwise as follows:

$$H_{(j-1)d+i,(n-1)p+l} = \sum_{t=1}^{T}\sum_{k=1}^{K}\mathbb{E}[z_{tk}]C_{il}^{(k)}x_n^{(t)}x_j^{(t)} + \sum_{t=1}^{T}\sum_{k=1}^{K}\mathbb{E}[z_{tk}]\nu_k P_{il}^{(k)}x_n^{(t)}x_j^{(t)}+$$

$$\xi_0(\mathbf{U}_0^{-1})_{il}(\mathbf{V}_0^{-1})_{nj}$$

Finally, we can solve for the least-squares estimate of $\mathrm{vec}(\mathbf{M})$ by:

$$\widehat{\mathrm{vec}(\mathbf{M})} = (\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top\mathrm{vec}(\mathbf{D}) \tag{3.40}$$

The $d \times p$ matrix from restoring the $dp \times 1$ vector $\widehat{\mathrm{vec}(\mathbf{M})}$ to a matrix yields $\mathbf{M}$, the posterior mean of $\mathbf{W}$.

This completes the VB M-step.

## 3.5 VB E-step

In the E-step, we take the expectation over the same joint distribution with respect to all parameters to compute the hyperparameters of the distribution on latent variables.

$$\ln q^*(\mathbf{Z}) =$$

$$= \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\mu},\mathbf{a}_k,b_k,\rho_k,\mathbf{W}}[\ln p(\boldsymbol{\eta},\{\mathbf{a}_k,b_k,\rho_k\}_{1:K},\mathbf{X},\mathbf{W},\{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k\}_{1:K},\boldsymbol{\pi},\mathbf{Z})] + \text{const.}$$

$$= \sum_{t=1}^{T}\sum_{k=1}^{K} z_{tk}\Bigg\{\mathbb{E}\left[\frac{1}{2}\ln\rho_k - \frac{\rho_k}{2}(\boldsymbol{\gamma}_k^\top \widetilde{\mathbf{W}}\widetilde{\mathbf{x}}_t\widetilde{\mathbf{x}}_t^\top\widetilde{\mathbf{W}}^\top\boldsymbol{\gamma}_k - 2\boldsymbol{\gamma}_k^\top\widetilde{\mathbf{W}}\widetilde{\mathbf{x}}_t\eta_t + \eta_t^2)\right] +$$

$$\mathbb{E}\left[\frac{1}{2}\ln|\boldsymbol{\Lambda}_k| - \frac{1}{2}(\mathbf{x}_t^\top\mathbf{W}^\top\boldsymbol{\Lambda}_k\mathbf{W}\mathbf{x}_t - 2\mathbf{x}_t^\top\mathbf{W}^\top\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k)\right] +$$

$$\mathbb{E}[\ln\pi_k]\Bigg\} + \text{const.} \tag{3.41}$$

Next, we define $\ln c_{tk}$ as follows:

$$\ln c_{tk} = \mathbb{E}\left[\frac{1}{2}\ln\rho_k - \frac{\rho_k}{2}(\boldsymbol{\gamma}_k^\top \widetilde{\mathbf{W}}\widetilde{\mathbf{x}}_t\widetilde{\mathbf{x}}_t^\top\widetilde{\mathbf{W}}^\top\boldsymbol{\gamma}_k - 2\boldsymbol{\gamma}_k^\top\widetilde{\mathbf{W}}\widetilde{\mathbf{x}}_t\eta_t + \eta_t^2)\right] +$$

$$\mathbb{E}\left[\frac{1}{2}\ln|\boldsymbol{\Lambda}_k| - \frac{1}{2}(\mathbf{x}_t^\top\mathbf{W}^\top\boldsymbol{\Lambda}_k\mathbf{W}\mathbf{x}_t - 2\mathbf{x}_t^\top\mathbf{W}^\top\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k)\right] +$$

$$\mathbb{E}[\ln\pi_k]$$

Exponentiating both sides of (3.41), we find:

$$q^*(\mathbf{Z}) \propto \prod_{t=1}^{T}\prod_{k=1}^{K} c_{tk}^{z_{tk}}$$

The distribution needs to be normalized for each value of $t$. Hence, we define a new variable $r_{tk}$ as follows:

$$r_{tk} = \frac{c_{tk}}{\sum_{k=1}^{K} c_{tk}}$$

Thus, $z_{tk}$ is categorically distributed with

$$\mathbb{E}[z_{tk}] = r_{tk} \tag{3.42}$$

We need to evaluate the expectations in (3.41) which we can do term by term:

$$\mathbb{E}[\ln\rho_k] = \psi(\alpha_k) - \ln(\beta_k) \tag{3.43}$$

$$\mathbb{E}\left[\frac{\rho_k}{2}(\eta_t - \boldsymbol{\gamma}_k^\top\mathbf{W}^\top\mathbf{x}_t^\top)^2\right] =$$

$$= \mathbb{E}\left[\frac{\rho_k}{2}\left(\boldsymbol{\gamma}_k^\top \widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{W}}^\top\boldsymbol{\gamma}_k - 2\boldsymbol{\gamma}_k^\top\widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t\eta_t + \eta_t^2\right)\right]$$

$$= \mathbb{E}_{\mathbf{W}}\left[\mathbb{E}_{\rho_k}\left[\mathbb{E}_{\boldsymbol{\gamma}_k|\rho_k}\left[\frac{\rho_k}{2}\boldsymbol{\gamma}_k^\top\widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{W}}^\top\boldsymbol{\gamma}_k - \rho_k\boldsymbol{\gamma}_k^\top\widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t\eta_t + \frac{\rho_k}{2}\eta_t^2\right]\right]\right]$$

$$= \mathbb{E}_{\mathbf{W}}\Bigg[\mathbb{E}_{\rho_k}\Bigg[\frac{\rho_k}{2}\left[\mathrm{Tr}\left(\widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{W}}^\top\mathbf{B}_k^{-1}\rho_k^{-1}\right) + \mathbf{g}_k^\top\left(\widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}^\top\widetilde{\mathbf{W}}^\top\right)\mathbf{g}_k\right] -$$

$$\rho_k\mathbf{g}_k^\top\widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t\eta_t + \frac{\rho_k}{2}\eta_t^2\Bigg]\Bigg]$$

$$= \mathbb{E}_{\mathbf{W}}\left[\frac{1}{2}\left\{\mathrm{Tr}\left(\widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{W}}^\top\mathbf{B}_k^{-1}\right) + \frac{\alpha_k}{\beta_k}\mathbf{g}_k^\top\left(\widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}^\top\widetilde{\mathbf{W}}^\top\right)\mathbf{g}_k\right\}\right] -$$

$$\frac{\alpha_k}{\beta_k}\mathbf{g}_k^\top\widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t\eta_t + \frac{1}{2}\frac{\alpha_k}{\beta_k}\eta_t^2\Bigg]\Bigg]$$

$$= \frac{1}{2}\Bigg\{\mathrm{Tr}\left[\left(\widetilde{\mathbf{U}}\,\mathrm{Tr}\left(\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{V}}\right) + \widetilde{\mathbf{M}}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{M}}^\top\right)\mathbf{B}_k^{-1}\right] +$$

$$\frac{\alpha_k}{\beta_k}\mathbf{g}_k^\top\left(\widetilde{\mathbf{U}}\,\mathrm{Tr}\left(\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{V}}\right) + \widetilde{\mathbf{M}}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{M}}^\top\right)\mathbf{g}_k\Bigg\} -$$

$$\frac{\alpha_k}{\beta_k}\mathbf{g}_k^\top\widetilde{\mathbf{M}}\tilde{\mathbf{x}}_t\eta_t + \frac{1}{2}\frac{\alpha_k}{\beta_k}\eta_t^2$$

$$= \frac{1}{2}\left\{\mathrm{Tr}\left[\left(\widetilde{\mathbf{U}}\,\mathrm{Tr}\left(\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{V}}\right) + \widetilde{\mathbf{M}}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{M}}^\top\right)\left(\mathbf{B}_k^{-1} + \frac{\alpha_k}{\beta_k}\mathbf{g}_k\mathbf{g}_k^\top\right)\right]\right\} -$$

$$\frac{\alpha_k}{\beta_k}\mathbf{g}_k^\top\widetilde{\mathbf{M}}\tilde{\mathbf{x}}_t\eta_t + \frac{1}{2}\frac{\alpha_k}{\beta_k}\eta_t^2$$

$$= \frac{1}{2}\left\{\mathrm{Tr}\left[\left(\widetilde{\mathbf{U}}\,\mathrm{Tr}\left(\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{V}}\right) + \widetilde{\mathbf{M}}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top\widetilde{\mathbf{M}}^\top\right)\widetilde{\mathbf{C}}_k\right]\right\} - \frac{\alpha_k}{\beta_k}\mathbf{g}_k^\top\widetilde{\mathbf{M}}\tilde{\mathbf{x}}_t\eta_t + \frac{1}{2}\frac{\alpha_k}{\beta_k}\eta_t^2 \tag{3.44}$$

where $\widetilde{\mathbf{C}}_k = \left(\mathbf{B}_k^{-1} + \frac{\alpha_k}{\beta_k}\mathbf{g}_k\mathbf{g}_k^\top\right)$.

$$\mathbb{E}[\ln|\boldsymbol{\Lambda}_k|] = \sum_{i=1}^{d}\psi\left(\frac{\nu_k + 1 - i}{2}\right) + d\ln 2 + \ln|\mathbf{P}_k| \tag{3.45}$$

$$\mathbb{E}\left[\frac{1}{2}\left(\mathbf{W}\mathbf{x}_t - \boldsymbol{\mu}_k\right)^\top\boldsymbol{\Lambda}_k(\mathbf{W}\mathbf{x}_t - \boldsymbol{\mu}_k)\right) =$$

$$= \mathbb{E}\left[\frac{1}{2}(\mathbf{x}_t^\top\mathbf{W}^\top\boldsymbol{\Lambda}_k\mathbf{W}\mathbf{x}_t - 2\mathbf{x}_t^\top\mathbf{W}^\top\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k)\right]$$

$$= \mathbb{E}_{\mathbf{W}}\left[\mathbb{E}_{\boldsymbol{\Lambda}_k}\left[\mathbb{E}_{\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k}\left[\frac{1}{2}\mathbf{x}_t^\top\mathbf{W}^\top\boldsymbol{\Lambda}_k\mathbf{W}\mathbf{x}_t - \mathbf{x}_t^\top\mathbf{W}^\top\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k + \frac{1}{2}\boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}_k\boldsymbol{\mu}_k\right]\right]\right]$$

$$= \mathbb{E}_{\mathbf{W}}\left[\mathbb{E}_{\boldsymbol{\Lambda}_k}\left[\frac{1}{2}\mathbf{x}_t^\top\mathbf{W}^\top\boldsymbol{\Lambda}_k\mathbf{W}\mathbf{x}_t - \mathbf{x}_t^\top\mathbf{W}^\top\boldsymbol{\Lambda}_k\mathbf{m}_k + \right.\right.$$

$$\left.\left.\frac{1}{2}\left[\mathrm{Tr}\left(\boldsymbol{\Lambda}_k\boldsymbol{\Lambda}_k^{-1}\kappa_k^{-1}\right) + \mathbf{m}_k^\top\boldsymbol{\Lambda}_k\mathbf{m}_k\right]\right]\right]$$

$$= \mathbb{E}_{\mathbf{W}}\left[\frac{1}{2}\mathbf{x}_t^\top\mathbf{W}^\top\mathbf{P}_k\nu_k\mathbf{W}\mathbf{x}_t - \mathbf{x}_t^\top\mathbf{W}^\top\mathbf{P}_k\nu_k\mathbf{m}_k + \right.$$

$$\left.\frac{1}{2}\left[\mathrm{Tr}\left(\mathbf{I}_d\kappa_k^{-1}\right) + \mathbf{m}_k^\top\mathbf{P}_k\nu_k\mathbf{m}_k\right]\right]$$

$$= \frac{1}{2}\nu_k\mathbf{x}_t^\top\left[\mathbf{V}\,\mathrm{Tr}\left(\mathbf{U}\mathbf{P}_k\right) + \mathbf{M}^\top\mathbf{P}_k\mathbf{M}\right]\mathbf{x}_t -$$

$$\nu_k\mathbf{x}_t^\top\mathbf{M}^\top\mathbf{P}_k\mathbf{m}_k + \frac{1}{2}\left[d\kappa_k^{-1} + \nu_k\mathbf{m}_k^\top\mathbf{P}_k\mathbf{m}_k\right] \tag{3.46}$$

$$\mathbb{E}[\ln\pi_k] = \psi(\lambda_k) - \psi\left(\sum_{k=1}^{K}\lambda_k\right) \tag{3.47}$$

This completes the VB E-step.

# 4

# Results and Discussion

## 4.1 Simulated Data

To demonstrate performance, we show results on both simulated and experimental data. Since the model is fully generative, data could be generated from the model itself, but for simplicity, we use the following three functions:

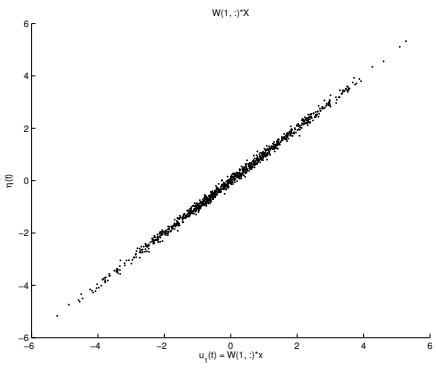
$$f(\mathbf{x}) = \mathbf{W}_{1,1:p}\mathbf{x}_t + \mathcal{N}(0, 0.1) \tag{4.1}$$

$$f(\mathbf{x}) = \tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t) + \mathcal{N}(0, 0.1) \tag{4.2}$$

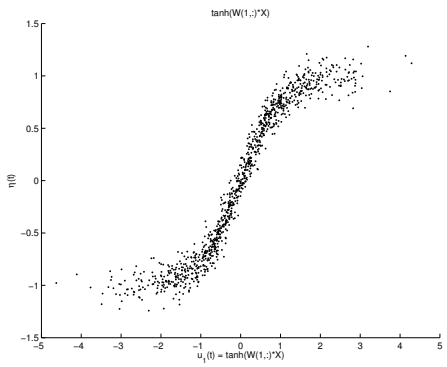
$$f(\mathbf{x}) = \tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)\mathbf{W}_{2,1:p}\mathbf{x}_t + \mathcal{N}(0, 0.1) \tag{4.3}$$

The notation $\mathbf{W}_{1,1:p}$ indicates taking entries in the first row of $\mathbf{W}$, and so on. Data generated from these functions is depicted in figure 4.1. We denote these the linear, hyperbolic tangent, and saddle dataset, respectively. Covariates $\mathbf{x}_t$ were randomly generated using a $p$-variate Gaussian.

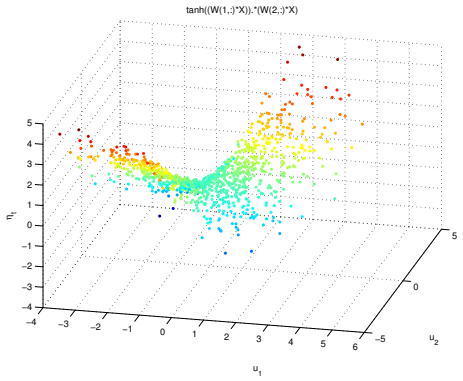
We fit three regressions on each dataset: a GLM with linear link function (i.e. linear regression, abbreivated LR); a GLM with a piecewise linear link function and pointwise (one-dimensional) nonlinearity (abbreviated PNGLM); and a full non-linear regression as specified by our model (FNLR). Each of the first two models




(a) $\mathbf{W}_{1,1:p}\mathbf{x}_t$




(b) $\tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)$




(c) $\tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)\mathbf{W}_{2,1:p}\mathbf{x}_t$

FIGURE 4.1: Simulated datasets.

can be considered submodels of our full model with different choices of $p$ and $K$. Specifically, the LR model corresponds to $K = 1$ and the PNGLM model corresponds to $d = p = 1$. Both the LR and FNLR models were set to have two dimensional receptive fields (this makes visualization straightforward, but models with higher dimensional receptive fields can also be fit), and for the PNGLM and FNLR models, $K = 12$. For each dataset, we tested each of the three models. The models were initialized to have vague priors. Table 4.1 lists the values of the hyperparameters chosen for all models. We ran VB for 300 iterations, after which we deemed that the inference had converged.

Table 4.2 shows the mean-squared errors (MSEs) for each combination of dataset

Table 4.1: Prior hyperparameter values.

| Hyperparameter | Value |
|---|---|
| $\alpha_0$ | $1/K$ |
| $\mathbf{B}_0$ | $\mathbf{I}_{d+1}$ |
| $\beta_0$ | 1 |
| $\mathbf{g}_0$ | $\mathbf{0}_{d+1}$ |
| $\lambda_0$ | 1 |
| $\mathbf{m}_0$ | $\mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$ |
| $\mathbf{M}_0$ | matrix-$\mathcal{N}(\mathbf{0}_{d\times p}, K\mathbf{I}_d, \mathbf{I}_p)$ |
| $\nu_0$ | $d$ |
| $\mathbf{P}_0$ | $\mathbf{I}_d$ |
| $\mathbf{U}_0$ | $\mathbf{I}_d$ |
| $\mathbf{V}_0$ | $\mathbf{I}_p$ |

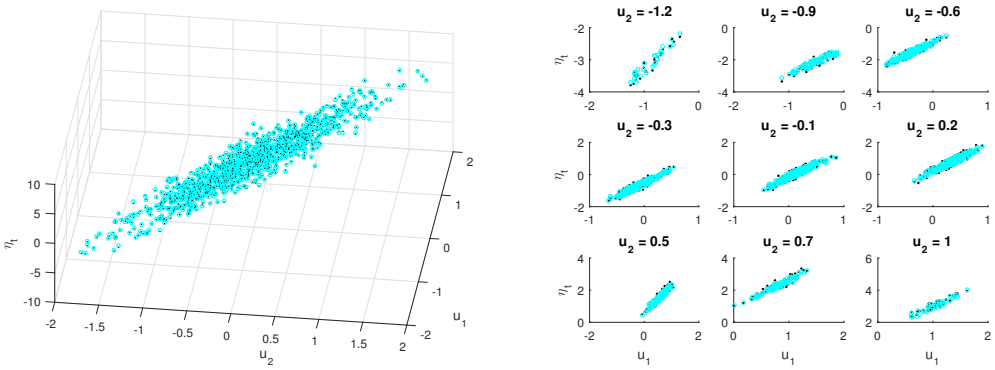
and model as well as the the total variance of the datasets. For the linear dataset, we see that all three models perform about equally well. To see why, we can take a look at figures 4.2-4.4. Each of these figures shows the value predicted by the respective models (in color) overlaid on the original data projected into the space of the discovered receptive field (in black). For each of the plots and subplots, $u_1$ and $u_2$ correspond to the first and second receptive fields discovered for the models. (Note that there is no $u_2$ in figure 4.3 since the PNGLM has a one-dimensional nonlinearity.) The different colors indicate the cluster labels to which the data points were assigned. In figure 4.2, we see that LR perfectly fits a hyperplane to the data. We get a hyperplane rather than a line since the model was set to have $d = 2$; thus, the model seeks the projection of the data in two dimensions which is best fit by a hyperplane. The projection of the data happens to be a hyperplane since the data was generated from a linear transformation ($\mathbf{W}_{1,1:p}\mathbf{x}_t$ is a line in one dimension). Figure 4.2b shows cross sections of the hyperplane for different values of $u_2$. In each of the cross sections, the predicted and true values overlap very closely. This is expected since the dataset is linear; hence, the linear model should be able to fit the data set extremely well.

Table 4.2: MSE on simulated data for Linear Regression (LR), Pointwise Nonlinearity (PNGLM), and Full Nonlinearity (FNLR).

| | $\mathbf{W}_{1,1:p}\mathbf{x}_t$ | $\tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)$ | $\tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)\mathbf{W}_{2,1:p}\mathbf{x}_t$ |
|---|---|---|---|
| $\text{var}(\eta_t)$ | 2.3969 | 0.6432 | 1.1188 |
| LR | 0.0106 | 0.0885 | 1.1077 |
| PNGLM | 0.0109 | **0.0156** | 0.1041 |
| FNLR | **0.0104** | 0.0160 | **0.0336** |

Figure 4.3 shows the same dataset fit by the PNGLM. Since the PNGLM model has only a one-dimensional nonlinearity, the predicted values will always lie along a one-dimensional curve. We see that the PNGLM uses one cluster to perfectly fit the data points to a line. This demonstrates the parsimony property of the Dirchlet prior; the model uses a minimal number of clusters relative to the maximum of $K = 12$ to maximize the posterior probability of the data. Finally, figure 4.4 shows the dataset fit by FNLR. Like the LR model, the FNLR model was fit with $d = 2$, so predictions lie on a plane. All cross sections show that the predicted and true values closely match, and we again see that the model is parsimonious in the number of clusters, employing only one to fit the entire dataset. This also make sense since the data is linear and, hence, only one line (cluster) should be required to fit the dataset.

For the hyperbolic tangent dataset, we see that LR does significantly worse than




(a) Fitted hyperplane. (b) Cross sections.

FIGURE 4.2: LR on $\mathbf{W}_{1,1:p}\mathbf{x}_t$ dataset.

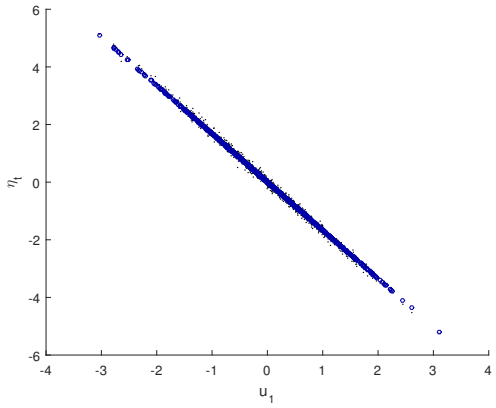
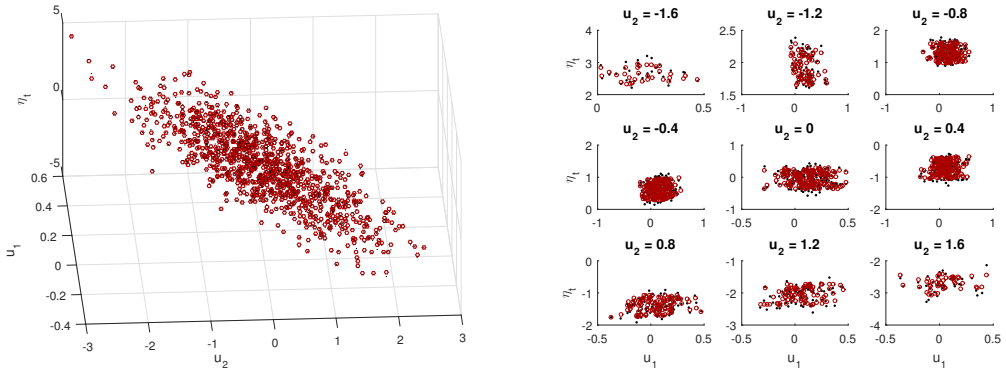



FIGURE 4.3: PNGLM on $\mathbf{W}_{1,1:p}\mathbf{x}_t$ dataset.

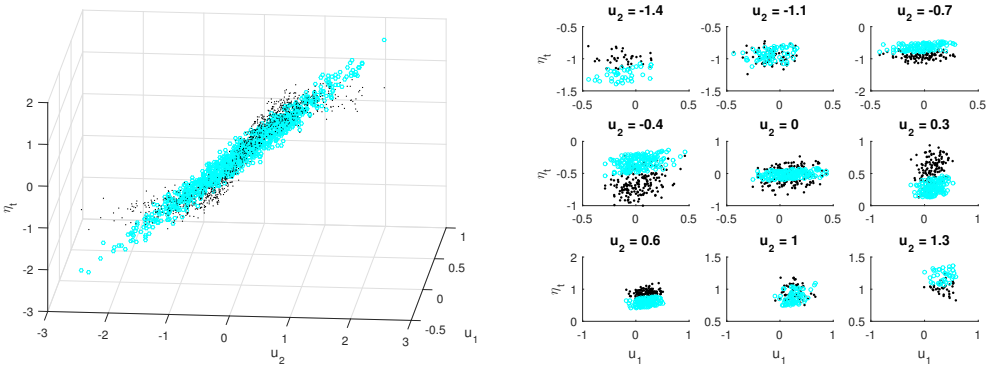
both the PNGLM and FNLR, and that both PNGLM and FNLR do about the same. Figures 4.5-4.7 illustrate these results. In figure 4.5, we see that data is projected into a space that results in predictions by a linear model that best fit the data; however, because the true data was generated by a function with a point (one-dimensional) nonlinearity, LR does worse here than for the linear dataset. The predictions on the plane miss points near the inflection and the tails of the hyperbolic tangent. This is evidenced both in the plot of the fitted hyperplane (figure 4.5a) and in the cross sections (figure 4.5b), where the black and cyan points do not overlap for values of $u_2$ near the inflection points and tails.




(a) Fitted surface.

(b) Cross sections.

FIGURE 4.4: FNLR on $\mathbf{W}_{1,1:p}\mathbf{x}_t$ dataset.




(a) Fitted hyperplane. (b) Cross sections.

FIGURE 4.5: LR on $\tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)$ dataset.

Figure 4.6 shows the same dataset fit by a PNGLM. As expected, the model does an excellent job discovering the nonlinearity. The one-dimensional receptive field is all that is necessary to fit the data since it was generated with a point nonlinearity. The model uses three clusters - one blue cluster in the center and two cyan and dark blue clusters at the tails - to fit the data compared to the maximum of twelve, demonstrating once again the parsimony of the model. While we might expect three colored lines, we instead, see three colored curves. This is another advantage of Bayesian inference: because we have posterior cluster assignment probabilities for each data point, we can take a weighted average over the predictions made by

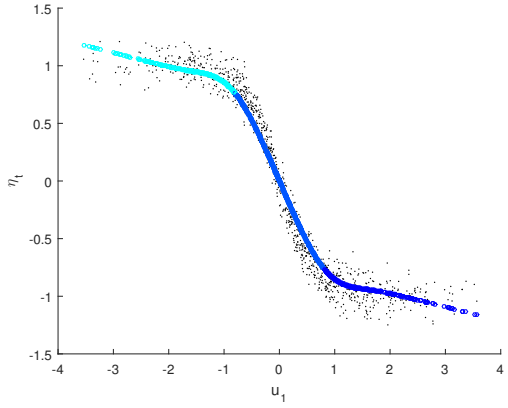
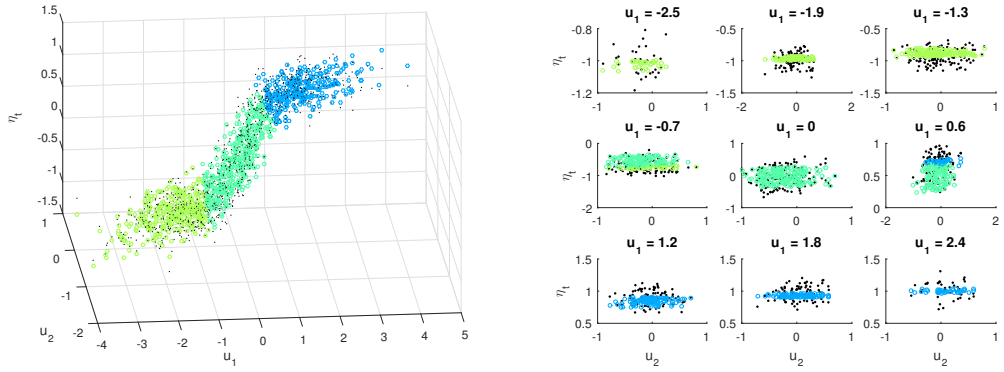



FIGURE 4.6: PNGLM on $\tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)$ dataset.

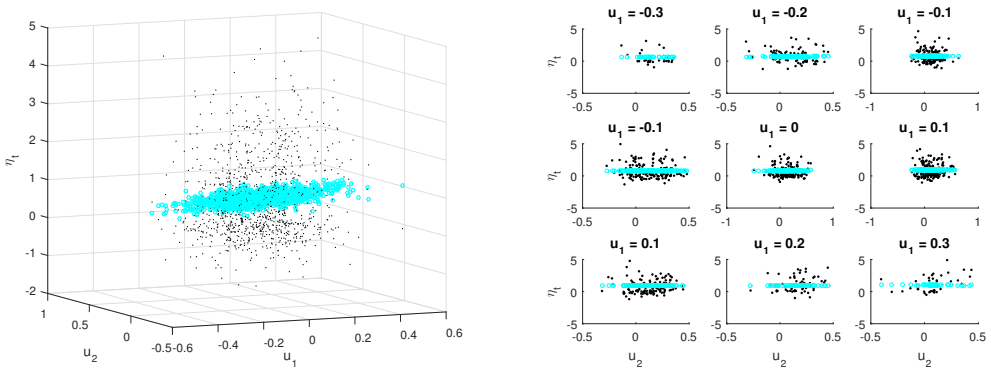



(a) Fitted surface. (b) Cross sections.

FIGURE 4.7: FNLR on $\tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)$ dataset.

assigning each data point to each cluster to come up with a final prediction for each data point. The cluster assignments are especially uncertain for data points near the boundary between two clusters, so averaging causes the predictions to be smoothed out, resulting in curves rather than jagged lines. The colors indicate the MAP cluster label for the data points.

Finally, figure 4.7 shows the dataset fit by FLNR. The data points are projected into the two dimensional space corresponding to the two dimensional receptive field. We see that the receptive fields are interacting nonlinearly, resulting in an S-shaped




(a) Fitted hyperplane. (b) Cross sections.

FIGURE 4.8: LR on $\tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)\mathbf{W}_{2,1:p}\mathbf{x}_t$ dataset.

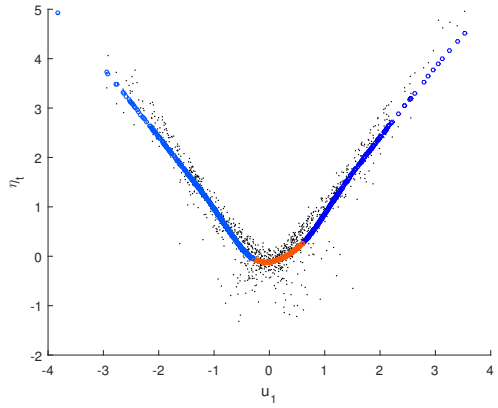
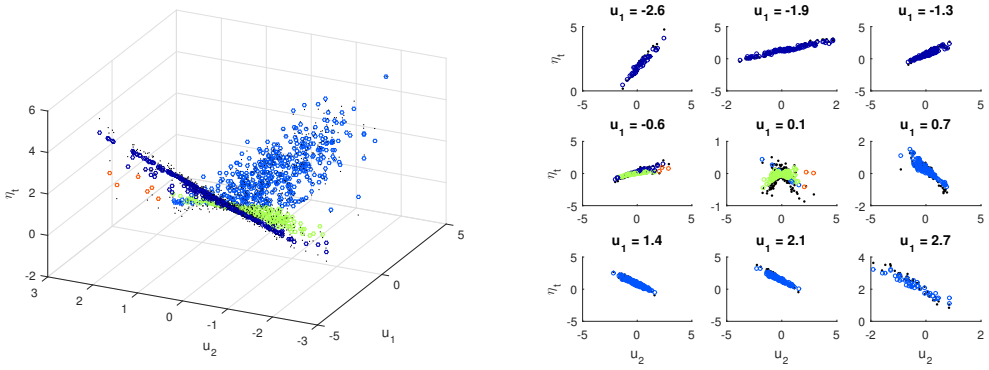



FIGURE 4.9: PNGLM on $\tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)\mathbf{W}_{2,1:p}\mathbf{x}_t$ dataset.

hyperbolic tangent surface of firing rates. Cross-sections show that a few of the points near the inflection and near the edges are missed, but the surface is overall very well-fit. The model uses only three clusters to fit the data. Introducing a second dimension of nonlinearity might require running the algorithm for more iterations, which is possibly why FNLR does slightly worse than PNGLM for this dataset.

For the saddle dataset, LR performs significantly worse than both PNGLM and FNLR. PNGLM does much better than LR but is outperformed by FNLR. Figures 4.8-4.10 show the predicted and true firing rates for models fitted to the saddle dataset. As shown in figure 4.1c, this dataset has a two-dimensional nonlinearity, one parabola going from left to right and one going into and out of the page. The point at which the parabolas meet is called the saddle point. Because this is a highly nonlinearly dataset, the LR model cannot project the data into a plane. Hence, LR predicts the mean of the data but cannot do any better. All cross sections show that a significant number of data points are missed by the model. PNGLM uses three clusters that result in relatively good predictions along the "long" side of the saddle (going from left to right). We see again that the model smoothes out predictions near the boundary between two clusters by taking a weighted average of the predictions made by assigning the data points to each of the clusters. Because the model is only




(a) Fitted surface. (b) Cross sections.

FIGURE 4.10: FNLR on $\tanh(\mathbf{W}_{1,1:p}\mathbf{x}_t)\mathbf{W}_{2,1:p}\mathbf{x}_t$ dataset.

capable of fitting a point nonlinearity, however, the model is not able to capture the "shorter" side of the saddle (going into and out of the page). This is remedied by FNLR, which is able to capture nonlinearities in both dimensions. We see that all cross sections show that predictions closely overlap with true values. Near the middle, we see that the short saddle is accurately predicted; running the model for more iterations may result in an even tighter fit. FNLR uses a total of four clusters to model the data.

## 4.2 Experimental Data

Experimental data was collected on monkeys performing a bimanual center-out task (Ifft et al. (2013)). In this task, the monkeys were trained to manipulate two virtual arms that appeared on a montior. The goal in each trial was to move both arms from the a central position to targets that appeared along the periphery. Figure 4.11 depicts the task. Arm positions were recorded and velocity was calculated by taking a two-point difference. These eight behavioral variables (position and velocity for both arms in 2-D space) were input as the regressors to each of the models. Firing rates from 377 neurons were recorded simultaneously. Tuning curves were fitted to

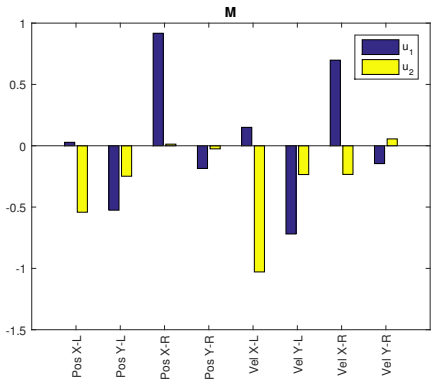
FIGURE 4.11: Bimanual center-out task (Ifft et al. (2013)). Monkey controls two virtual arms. The goal of the task is to move the arms into targets that appear on the periphery of the computer monitor.

each of these neurons using each of the three models.

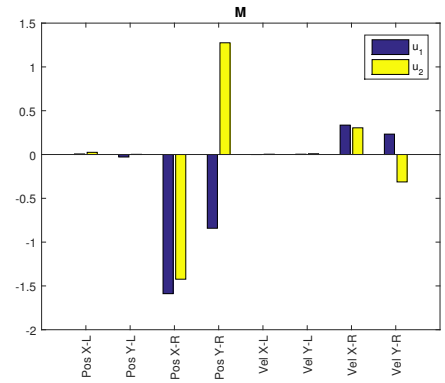
Neurons were ranked first in order of variance then in order of predictive power in a linear model consisting of all 377 neuron. Rankings were averaged, and the results for the top three neurons are reported here. All models were initialized with semi-empirical priors. Figure 4.12 gives the receptive fields of each of the neurons. The $x$-axis is labeled by the eight behavioral parameters described above ($x$- and $y$-position and velocity for right and left arms). Among the three neurons, neuron 3 has the simplest receptive field. The first dimension $u_1$ encodes primarily for right arm $x$- position, while the second dimension $u_2$ more or less averages over the eight parameters. Hence, this neuron can be called a "right arm $x$-position" neuron. The

Table 4.3: MSE on experimental data for Linear Regression (LR), Pointwise Nonlinearity (PNGLM), and Full Nonlinearity (FNLR).

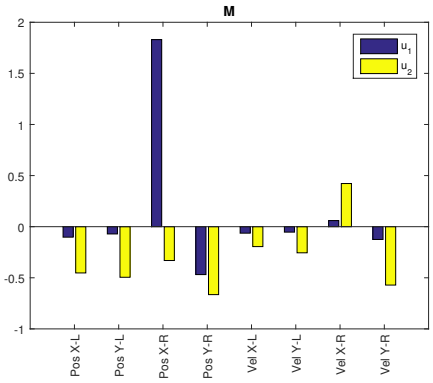
| | Neuron 1 | Neuron 2 | Neuron 3 |
|---|---|---|---|
| var($\eta_t$) | 34.2759 | 88.5033 | 24.2922 |
| LR | 31.0126 | 77.5427 | 22.0589 |
| PNGLM | 24.6845 | **66.9744** | 15.7245 |
| FNLR | **20.9112** | 72.5174 | **11.9120** |




(a) Neuron 1




(b) Neuron 2

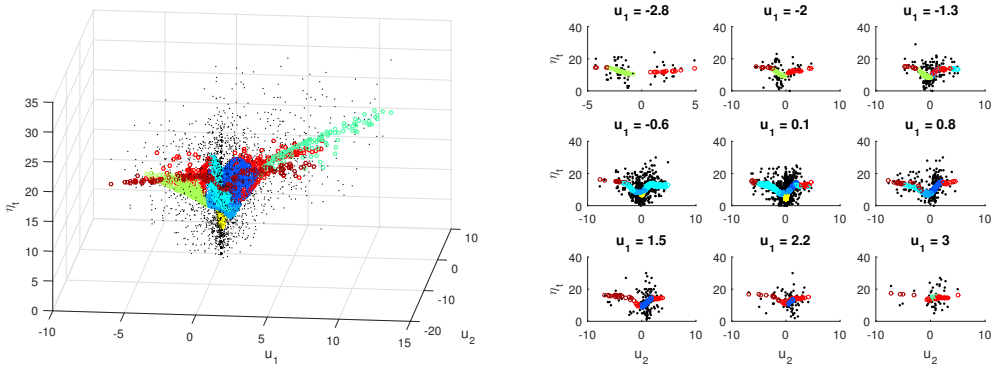



(c) Neuron 3

FIGURE 4.12: Receptive fields.

first dimension of neuron 2's receptive field performs an average over the $x$- and $y$-positions of the right arm while the second dimension performs a contrast between the two. Hence, we can call this a "right arm position neuron." Finally, neuron 1's receptive field is the most complicated. There is no simple way to describe the receptive field of the neuron, but in general, its first dimension encodes a contrast of right arm $x$-position and velocity against left arm $y$-position and velocity, and the second dimension encodes an average of left arm $x$-position and velocity.

Table 4.3 highlights the performance of each model on these three selected neurons. We see that for all three neurons, FNLR significantly outperforms LR, due to a better fit to the neurons' highly non-linear tuning curves. FNLR also outperforms

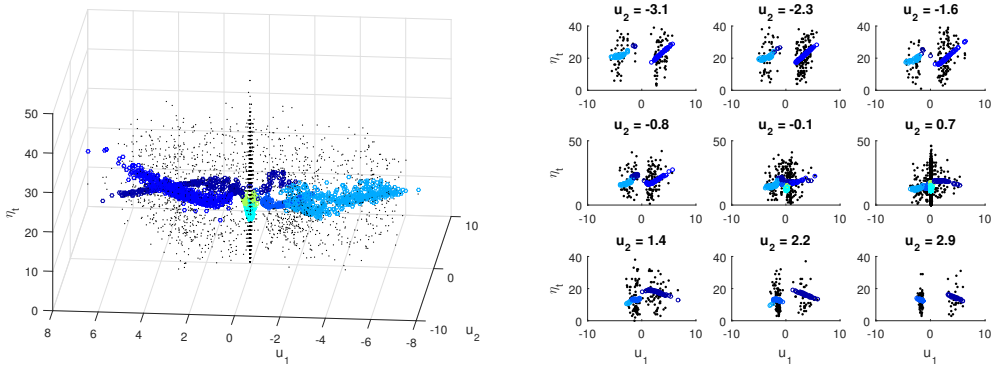



(a) Neuron 1 fitted with FNLR. (b) Cross sections.

FIGURE 4.13: FNLR fit on neuron 1.

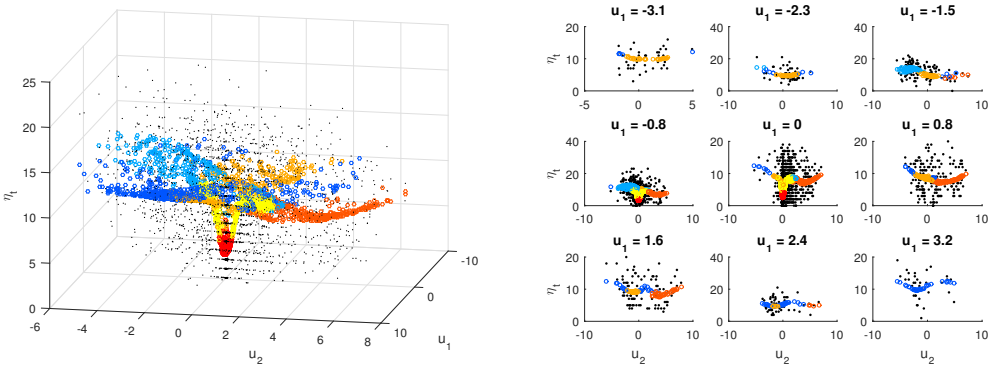
PNGLM for neurons 1 and 3, but PNGLM does better than FNLR for neuron 2. This is likely because neuron 2 only has a one-dimensional non-linearity; hence, fitting it with FNLR results in superfluous parameters that only widen the prediction error bars while biasing the parameters toward the prior mean.

For illustration, we show cluster-labeled predictions made by FNLR for these neurons in figure 4.13-4.15. Each of the neurons was fit with a maximum number of clusters equal to twelve but only eight components were needed for neuron 1, five for neuron 2, and six for neuron 3. From the cross sections, we see that FNLR




(a) Neuron 2 fitted with FNLR. (b) Cross sections.

FIGURE 4.14: FNLR fit on neuron 2.




(a) Neuron 3 fitted with FNLR. (b) Cross sections.

FIGURE 4.15: FNLR fit on neuron 3.

accurately captures many of the interesting properties of the dataset. For instance, in the first subplot of figure 4.13b, we see that there is a gap in between two regions of the stimulus space in which the neuron does not fire at all. The model is able to adapt to this by fitting separate lines to each of the regions surrounding the gaps. This is also evident in many of the subplots in figure 4.14b. Next, there are regions for which the tuning curve is nonlinear; see, for example, the $u_1 = -0.6$ subplot of figure 4.13b. The model captures the nonlinearity by fitting several piecewise components and smoothing areas at the boundary between two clusters. Finally, for each of the neurons, there is a large vertical cluster near the region where $u_1$ and $u_2$ both equal 0. This suggests that when the monkey is not moving ($u_1 = u_2 = 0$), there is a great deal of variability in the neural firing rates. The model attempts to fit this variability using a few clusters near the zero region (see, for example, the $u_1 = 0$ subplot of 4.15b), but does so poorly. This is purely a result of the fact that neurons are extremely noisy and suggests that good decoding of kinematics require information from multiple neurons.

# 5

# Conclusions and Future Work

Tuning curves of cortical neurons can be dynamic, highly nonlinear, and unique to each and every neuron in the motor cortex. Furthermore, the receptive fields to which neurons respond can be varied. In the context of BMI, fixing the same receptive field for all neurons and making poor approximations to tuning curves can result in poor decoding performance. Classically, BMI decoders have employed cosine, linear, and quadratic tuning functions for position and velocity. To account for varying nonlinearities among the many neurons from which recordings are taken, we have formulated here a nonparametric, nonlinear regression model. The model fits a piecewise linear function to a tuning curve for a low-dimensional projection of behavioral variables. Hence, the model not only discovers unique nonlinearities for each neuron but also discovers the receptive fields to which the neuron best responds. We fit this model using VB, which allows for fast approximation of the posterior distributions of model parameters. We show that the model accurately discovers nonlinearities in tuning functions in both simulated and experimental data.

In future work, we plan to make the model more robust to local modes by incorporating split and merge steps while clustering. Moreover, we plan to add an

additional layer to our model by treating the Gaussian firing rate as a parameter for a Poisson firing rate observation. This will tie in with existing literature on LNP models and will more accurately represent the firing properties of neurons. We also plan to measure the decoding performance using the newly learned tuning functions. This will require inverting the model in the sense that we will back out behavioral parameters from the firing rates rather than predict firing rates from behavioral parameters. Since this could result in multimodal distributions for the behavioral parameters, part of the challenge will be in integrating information across all neurons in a clever way to avoid taking a simple average over the modes for each neuron.

This work is highly interdisciplinary and has important implications outside of BMI. One could employ this model to discover whether or not circuit-level non-linear operations proposed by systems neuroscientists, such as divisive normalization, can be identified. It could also be used to identify dendritic non-linearities from spiking data and potentially be used to classify cell type based on a cell's non-linear transfer function. Figure 5.1 illustrates this with a cartoon. The gray boxes depict neurons which are interconnected in a complex neural circuit. The black triangles symbolize electrodes which are recording from the neurons. As stimuli are presented, the neurons respond, and receptive fields and tuning curves can be characterized. The analogous circuit is depicted on the right with tuning curves fitted by FNLR. Assuming that neurons with similar tuning curves have similar cellular structure, one could cluster tuning parameters to identify cell types of unlabeled neurons.

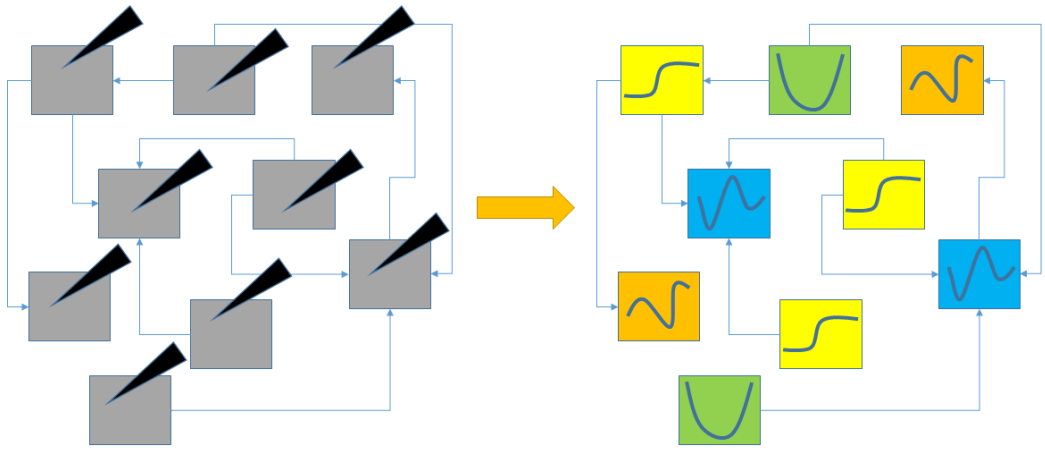


FIGURE 5.1: Explaining the diversity of neuron type characterized by a neuron's dendritic structure by clustering nonlinear tuning curves.

# Bibliography


Aldous, D. J. (1985), *Exchangeability and related topics*, Springer.

Asatryan, D. G. and Feldman, A. G. (1965), "Functional tuning of the nervous system with control of movement or maintenance of a steady posture: I. Mechanographic analysis of the work of the joint or execution of a postural task," *Biophysics*, 10, 925–934.

Beal, M. J. (2003), *Variational algorithms for approximate Bayesian inference*, University of London.

Bishop, C. M. (2006), *Pattern recognition and machine learning*, springer.

Bizzi, E., Accornero, N., Chapple, W., and Hogan, N. (1984), "Posture control and trajectory formation during arm movement," *The Journal of Neuroscience*, 4, 2738–2744.

Boyd, S. and Vandenberghe, L. (2004), *Convex optimization*, Cambridge university press.

Chapin, J. K., Moxon, K. A., Markowitz, R. S., and Nicolelis, M. A. L. (1999), "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nature Neuroscience*, 2, 664–670.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38.

El-Arini, K. (2008), "Dirichlet Processes," .

Ferguson, T. S. (1973), "A Bayesian analysis of some nonparametric problems," *The annals of statistics*, pp. 209–230.

Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., and Massey, J. T. (1982), "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex," *The Journal of Neuroscience*, 2, 1527–1537.

Hannah, L. A., Blei, D. M., and Powell, W. B. (2011), "Dirichlet process mixtures of generalized linear models," *The Journal of Machine Learning Research*, 12, 1923–1953.

Hubel, D. H. and Wiesel, T. N. (1959), "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, 148, 574–591.

Ifft, P. J., Shokur, S., Li, Z., Lebedev, M. A., and Nicolelis, M. A. (2013), "A brain-machine interface enables bimanual arm movements in monkeys," *Science translational medicine*, 5, 210ra154–210ra154.

Kullback, S. and Leibler, R. A. (1951), "On information and sufficiency," *The annals of mathematical statistics*, pp. 79–86.

Li, Z., O'Doherty, J. E., Hanson, T. L., Lebedev, M. A., Henriquez, C. S., and Nicolelis, M. A. (2009), "Unscented Kalman filter for brain-machine interfaces," *PloS one*, 4, e6243.

MacQueen, J. et al. (1967), "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA.

McCulloch, W. S. and Pitts, W. (1943), "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, 5, 115–133.

Moran, D. W. and Schwartz, A. B. (1999), "Motor cortical activity during drawing movements: population representation during spiral tracing," *Journal of neurophysiology*, 82, 2693–2704.

Nelder, J. A. and Baker, R. (1972), "Generalized linear models," *Encyclopedia of Statistical Sciences*.

O'Doherty, J. E., Lebedev, M. A., Ifft, P. J., Zhuang, K. Z., Shokur, S., Bleuler, H., and Nicolelis, M. A. (2011), "Active tactile exploration using a brain-machine-brain interface," *Nature*, 479, 228–231.

Pais-Vieira, M., Chiuffa, G., Lebedev, M., Yadav, A., and Nicolelis, M. A. (2015), "Building an organic computing device with multiple interconnected brains," *Scientific reports*, 5.

Parisi, G. (1988), *Statistical field theory*, Addison-Wesley.

Pillow, J. W., Paninski, L., and Simoncelli, E. P. (2003), "Maximum Likelihood Estimation of a Stochastic Integrate-and-Fire Neural Model." in *NIPS*.

Ramakrishnan, A., Ifft, P. J., Pais-Vieira, M., Byun, Y. W., Zhuang, K. Z., Lebedev, M. A., and Nicolelis, M. A. (2015), "Computing arm movements with a monkey brainet," *Scientific reports*, 5.

Rasmussen, C. E. (2006), "Gaussian processes for machine learning," .

Rasmussen, C. E. and Ghahramani, Z. (2002), "Infinite mixtures of Gaussian process experts," *Advances in neural information processing systems*, 2, 881–888.

Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N., and Wolpaw, J. R. (2004), "BCI2000: A general-purpose brain-computer interface (BCI) system," *IEEE: Transactions on Biomedical Engineering*, 51, 1034–1043.

Schwartz, O., Pillow, J. W., Rust, N. C., and Simoncelli, E. P. (2006), "Spike-triggered neural characterization," *Journal of Vision*, 6, 13.

Sergio, L. E., Hamel-Pâquet, C., and Kalaska, J. F. (2005), "Motor cortex neural correlates of output kinematics and kinetics during isometric-force and arm-reaching tasks," *Journal of neurophysiology*, 94, 2353–2378.

Snelson, E. and Ghahramani, Z. (2007), "Local and global sparse Gaussian process approximations," in *International Conference on Artificial Intelligence and Statistics*, pp. 524–531.

Taylor, D. M., Tillery, S. I. H., and Schwartz, A. B. (2002), "Direct Cortical Control of 3D Neuroprosthetic Devices," *Science*, 296, 1829–1832.

van Hemmen, J. L. and Schwartz, A. B. (2008), "Population vector code: a geometric universal as actuator," *Biological cybernetics*, 98, 509–518.

Velliste, M., Perel, S., Spalding, M. C., Whitford, A. S., and Schwartz, A. B. (2008), "Cortical control of a prosthetic arm for self-feeding," *Nature*, 453, 1098–1101.

Vintch, B., Zaharia, A., Movshon, J., and Simoncelli, E. P. (2012), "Efficient and direct estimation of a neural subunit model for sensory coding," in *Advances in neural information processing systems*, pp. 3104–3112.

Weiss, P. (1907), "L'hypothèse du champ moléculaire et la propriété ferromagnétique," *J. Phys. Theor. Appl.*, 6, 661–690.

Wu, A., Park, I. M., and Pillow, J. (2015), "Convolutional Spike-triggered Covariance Analysis for Neural Subunit Models," in *Advances in Neural Information Processing Systems*.