RICE UNIVERSITY

Selecting Models that Describe Neural Population Responses

by

Jyotirmai Uppuluri

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Science

APPROVED, THESIS COMMITTEE:

[Signatures]

Don H. Johnson, Chair
J.S. Abercrombie Professor of Electrical and
Computer Engineering and of Statistics

Richard G. Baraniuk
Victor E. Cameron Professor
of Electrical and Computer Engineering

Rudolf H. Riedi
Associate Professor of Statistics

HOUSTON, TEXAS

MAY 2007
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.
Abstract

Selecting Models that Describe Neural Population Responses

by

Jyotirmai Uppuluri

This research provides a new approach to analyzing the statistical properties of limited data sets. A lack of sufficient data is a pressing issue in the neuroscience community, where researchers want to understand the statistics of interactions between neurons. The methods used in this work involve an information-theoretic criterion for determining a class of models that can accurately describe the available neural data’s statistics. These methods combine principles from Minimum Description Length coding and an approach based on the Kullback-Leibler distance. In the context of limited data, this research provides a less forced or error-prone view of the statistical properties of the data. This provides researchers in the neuroscience community with a more robust approach to data analysis.
Acknowledgments

I would like to thank Dr. Richard Baraniuk and Dr. Rolf Riedi for serving on my thesis committee and for providing valuable feedback and kind words. I would also like to thank Ilan Goodman, who provided countless hours of his own time, wise counsel, and support when I needed it most. Thanks also go to my friends and family, without whom I would never have persevered. Finally, I would like to thank my advisor, Dr. Don H. Johnson, who gave me his time, attention, support, and great ideas at every turn and without hesitation.
Contents

1 Multivariate Models 5
  1.1 The Sarmanov-Lancaster Model .......................... 5
    1.1.1 Example with 3 Binary Random Variables ............. 7
    1.1.2 The Exponential Model .................................. 9
    1.1.3 Maximum Likelihood Estimation with the Sarmanov- Lancaster Model ........................................ 11
    1.1.4 Maximum Likelihood Estimation with the Exponential Model 13

2 Minimum Description Length 18
  2.1 Data Compression and MDL Codelengths .................... 19
    2.1.1 Two-Part Codes for MDL ................................. 20
    2.1.2 MDL in Practice ........................................... 22
    2.1.3 Normalized Maximum Likelihood and Regret .............. 23
  2.2 NML and Model Selection .................................... 25
    2.2.1 Asymptotic Formula for the Normalized Maximum Likelihood Model ................................ 26
    2.2.2 Application of NML ........................................ 28
    2.2.3 Invariance of the Fisher Information Term .............. 30
    2.2.4 The Exponential Model Family and NML .................. 32
    2.2.5 The Sarmanov-Lancaster Model Family and NML .......... 33
  2.3 Some Observations on NML .................................. 36

3 Model Selection 40
  3.1 Introduction ................................................... 40
  3.2 Kullback-Leibler-based Model Analysis ...................... 41
  3.3 Examples ....................................................... 45
    3.3.1 NML and Our Approach to Model Selection .............. 47
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Stochastic Complexity for Two Neurons</td>
<td>34</td>
</tr>
<tr>
<td>2.2</td>
<td>Stochastic Complexity for Three Neurons</td>
<td>34</td>
</tr>
<tr>
<td>2.3</td>
<td>Three Neuron Probability of Error for NML-based Model Selection</td>
<td>35</td>
</tr>
<tr>
<td>2.4</td>
<td>Two Neuron Probability of Error for NML-based Model Selection</td>
<td>36</td>
</tr>
<tr>
<td>2.5</td>
<td>Two Independent Neurons $o(1)$ Term</td>
<td>37</td>
</tr>
<tr>
<td>2.6</td>
<td>Two Dependent Neurons $o(1)$ Term</td>
<td>38</td>
</tr>
<tr>
<td>2.7</td>
<td>One Neuron $o(1)$ Term</td>
<td>39</td>
</tr>
<tr>
<td>3.1</td>
<td>Minimum KL Distances for Three Model Classes</td>
<td>43</td>
</tr>
<tr>
<td>3.2</td>
<td>Two Neuron Model Feasibility Regions</td>
<td>48</td>
</tr>
<tr>
<td>3.3</td>
<td>KL-based Contours versus Fisher-Information-based Contours</td>
<td>49</td>
</tr>
<tr>
<td>3.4</td>
<td>Model Feasibility Region for Real Data</td>
<td>50</td>
</tr>
<tr>
<td>3.5</td>
<td>Model Selection for 10 Data Samples</td>
<td>54</td>
</tr>
<tr>
<td>3.6</td>
<td>Model Selection for 40 Data Samples</td>
<td>55</td>
</tr>
<tr>
<td>3.7</td>
<td>Model Selection for 300 Data Samples</td>
<td>56</td>
</tr>
</tbody>
</table>
Introduction

Decoding the activity of neurons is the goal of computational neuroscience research. Neurons, both alone and in cooperation with one-another, encode and communicate information about the world around them in the form of spikes, electrical signals that originate in the body of the neuron and propagate toward other neurons in a population. Spike waveforms for a given neuron are almost always identical, meaning that the information encoded by a spike is when it occurs. With the advent of multi-electrode technology able to record data from several neurons at once, scientists would like to know if and how neurons cooperate with each other to encode information in the form of neural population codes. It is the inter-relationships among individual neural responses that determine such a joint population code. Thus, one must measure the entire population’s response to a stimulus instead of simply inferring population behavior from single-neuron responses. These recorded responses can be interdependent in a vast number of ways, and techniques to analyze this data face what statisticians call the “curse of dimensionality.”

In the usual procedure for statistically analyzing a neural population’s spike response, each neuron’s discharge (or spiking) pattern is binned in time, and the presence or absence of a spike in each time bin is represented by a binary symbol. From our population of $N$ jointly recorded neural responses, we want to mea-
sure inter-neuron correlations as well as determine individual spike discharge rates. Weinberger [21] shows that the amount of data needed for accurate data analysis must be proportional to $2^N$ samples [13]. Additionally, as another neuron is added to the population being observed, any procedure based on binned data will need twice as much data as before to sustain measurement accuracy. To analyze $B$-bin intra-neuron dependence as well as inter-neuron relationships, the amount of data needed is proportional to $2^{BN}$ samples. This heavy data requirement arises because we typically have no model for the joint distribution of the population, and we are forced to use non-parametric techniques. Oftentimes, it is impossible to obtain such large datasets for analysis. However, neuroscience researchers still use techniques such as cross-correlation to test for statistical dependence between pairs of neurons in available datasets [9]. This cross-correlation type of analysis, however, is not appropriate for measuring complicated, non-linear relationships between more than two neurons. Additionally, while statistics like cross-correlations may be measured for a dataset of any size, simply analyzing these statistics in isolation precludes a more comprehensive, and realistic, view of neural population dynamics. That is, estimating individual statistics like cross-correlations and placing confidence intervals around these estimates does not tell us whether the larger picture we have effectively constructed about the population (i.e., the assumption of the presence or absence of each pairwise interaction) is viable. To construct an accurate picture of a population's statistics, we must either limit the scope of our analysis to very
small populations over small time windows, or we must analyze the available data in a manner that better reveals underlying structure.

In this thesis, I suggest a different approach to population data analysis. Due to the lack of data that prevents non-parametric data analysis, I will take an approach that involves the use of parametric model families that encompass non-linear, higher-order interactions between several neurons and, thus, form a more cohesive picture of a population’s dynamics. If we have such a model for a population response, we may write the probability of the response as follows. At each time bin, we concatenate the symbols from all of the neurons to form the ordered response measurement $R$ to describe the entire population’s response in this bin [13]. We perform our analysis at each bin separately, allowing us to consider one bin at a time. We assume we have $M$ such measured responses produced by repeating a stimulus $M$ times; these together form the response vector $\mathbf{R} = \{R_1, \ldots, R_M\}$ which represents a dataset for the population. We further assume the $M$ responses are produced by carefully repeating the stimulus in a manner which minimizes sequential response effects. Consequently, the $M$ responses are considered statistically independent of each other. If $P(R)$ denotes the probability of producing one of the $M$ population responses, the probability of the entire set of measurements $P(\mathbf{R})$ thus equals $\prod_{m=1}^{M} P(R_m)$.

With the ability to write the probability of a response as described above, a host of statistical techniques may be employed to analyze the available data. The
questions that must be answered now are, with limited data available, how can one choose an appropriate parametric model family to describe the data, and more specifically, how can one choose reasonable models within that family? In this thesis, I provide a method for model selection that seeks to answer these two questions. Additionally, I provide a way to evaluate a chosen model (and not simply accept or reject hypotheses about individual statistics) based on the data.

In Chapter 1, I discuss two multivariate models for neural population responses, the Sarmanov-Lancaster model and the exponential (or log-linear) model. Although both are capable of describing the full range of binary-valued neural responses, the exponential model proves to be better-suited to our method for model selection.

In Chapter 2, I discuss minimum description length methods, and specifically, I show how Normalized Maximum Likelihood-based model selection can be used in conjunction with the exponential model to choose a model family for observed neural data.

In Chapter 3, I discuss an information-theoretic method that evaluates model family choices and also finds all models in the given family that are likely to have produced the observed data. I also show how combining this method with Normalized Maximum-Likelihood-based model-selection allows one to first determine which of increasingly complex model families best fits the available data, and then, within this model family, which models are deemed likely to have produced the observed data.
Chapter 1

Multivariate Models

In this chapter, we outline two multivariate descriptions for single-bin neural population responses - the Sarmanov-Lancaster model and the exponential (or log-linear) model; and discuss various properties of both models. Additionally, we present the problem of maximum likelihood parameter estimation in the context of both models, a task we must be able to perform to implement the model-selection methods presented in Chapter 2.

1.1 The Sarmanov-Lancaster Model

Goodman, in his thesis, developed the Sarmanov-Lancaster model for multivariate probability distributions by extending a series expansion of bivariate distributions introduced first by Sarmanov and later by Lancaster [9; 19; 16]. Let $\mathbf{X} = (X_1, ..., X_N)$ be a discrete random vector with joint probability distribution
(probability mass function) \( p_X(x) \) and marginal distributions \( p_{X_i}(x_i) \). Define the quantity \( \phi^2 = \sum_x p^2_X(x)/\prod_{i=1}^N p_{X_i}(x_i) - 1 \). Provided that \( \phi^2 < \infty \), Goodman states that the joint probability function can be represented by the expansion

\[
p_X(x) = \prod_{i=1}^N p_{X_i}(x_i) \left[ 1 + \sum_{i_1,...,i_N} a_{i_1...i_N} \prod_{n=1}^N \psi_{i_n}^{(n)}(x_n) \right]. \tag{1.1}
\]

Here \( \psi_{i_n}^{(n)} \) form a complete set of orthonormal functions with respect to the marginal distribution \( p_{X_n}(x_n) \) with \( \psi_0^{(n)} = 1, n = 1,...,N \). Various orthonormal functions such as orthogonal polynomials and normalized Legendre polynomials have been used by Bahadur and Lancaster to expand continuous bivariate distributions [3; 16]. What is arguably most important about this expansion is that it explicitly codes the dependence at various levels in the hierarchy of interactions. Let \( i = i_1...i_N \), and define \( \psi_1(x) = \prod_{n=1}^N \psi_{i_n}^{(n)}(x_n) \). Goodman notes that the indices \( i \) can be partitioned into disjoint sets (each representing two or more of the random variables), which he denotes \( \beta_k, k = 1,...,2^N-N-1 \), such that the functions \( \psi_j, j \in \beta_k \) are functions only of the \( k^{th} \) unique subset of the \( N \) random variables. For example, if \( \beta_1 \) represents the subset \( (X_1, X_2) \), then for all \( j \in \beta_1, \psi_j(x) = \psi_j(x_1, x_2) \). Equation (1.1) may then be rewritten as

\[
p_X(x) = \prod_{i=1}^N p_{X_i}(x_i) \left[ 1 + \sum_{k=1}^{2^N-N-1} \sum_{i \in \beta_k} a_i \psi_i \right]. \tag{1.2}
\]

With this new equation, the coefficients \( a_i, i \in \beta_k \) represent only those interactions that occur within the \( k^{th} \) subset, separate from lower order interactions or interac-
tions with variables outside of the $k^{th}$ subset.

1.1.1 Example with 3 Binary Random Variables

Goodman provides an example that proves to be valuable for the three neuron population case. Here, we have three jointly defined binary random variables $X_1, X_2,$ and $X_3$, with $p_n = \Pr[X_n = 1]$ and $\sigma_n^2 = p_n(1 - p_n), n = 1, 2, 3$ (the usual marginal probabilities and variances of each of the variables). Goodman uses the orthogonal polynomials as a basis set for the marginal probability distributions, obtaining, for each $X_n$, $\psi_0(x_n) = 1$ and $\psi_1(x_n) = (x_n - p_n)/\sigma_n$. Using this expansion, the coefficients $a_{i_1i_2i_3}$ may be calculated, and one obtains

$$a_{12} = \rho_{12}$$

$$a_{13} = \rho_{13}$$

$$a_{23} = \rho_{23}$$

$$a_{123} = \rho_{123} - \frac{p_1}{\sigma_1} \rho_{23} - \frac{p_2}{\sigma_2} \rho_{13} - \frac{p_3}{\sigma_3} \rho_{12}. $$

Of great value in this model is the fact that $\rho_{ij} = (E[X_iX_j] - p_ip_j)/\sigma_i\sigma_j$ are the familiar product-moment correlation coefficients. Interpreting these coefficients is naturally intuitive to one looking to the Sarmanov-Lancaster model to yield an understanding of data. Additionally, $\rho_{ijk} = (E[X_iX_jX_k] - p_i p_j p_k)/\sigma_i\sigma_j\sigma_k$ represents third-order dependence and is a quantity similar to correlation.
The Sarmanov-Lancaster model is a useful modeling tool in that it consists only of univariate parameters and dependence parameters. Most importantly, the dependence parameters represent specific interactions between the variables and cover the full range of dependencies achievable for a given set of marginals [9]. One can decipher the level of dependence within various subsets of variables intuitively, particularly for the case of a bivariate subset (in which the dependence is equivalent to the typical correlation). The Sarmanov-Lancaster model is particularly useful for understanding the interactions of two and three neurons, but when dealing with larger populations, the number of parameters increases exponentially in the number of neurons. Additionally, the dependence parameters do not vary independently of each other. For example, Joe [12] states the admissible range of second-order correlation in the bivariate Bernoulli case to be

$$\rho \geq \max \left\{ -\sqrt{\frac{p_1 p_2}{(1 - p_1)(1 - p_2)}}, -\sqrt{\frac{(1 - p_1)(1 - p_2)}{p_1 p_2}} \right\}. $$

The range of correlation coefficients is thus limited by the marginal probabilities, which complicates the interpretation of correlation values. Particularly, finding a maximum-likelihood estimate of the parameters of the Sarmanov-Lancaster problem proves to be problematic. While it is numerically calculable in some cases, this approach does not always prove to be stable, nor do numerical methods converge to an answer quickly, to be described in further detail in the Maximum Likelihood section of this chapter. For these reasons, we use a different, but equivalent, multivariate
model for data analysis.

1.1.2 The Exponential Model

The exponential or log-linear model of multivariate data is frequently used, but does not provide as useful a separation of interactions as does the Sarmanov-Lancaster model. Specifically, the parameters of the exponential model do not correspond to interactions of unique subsets on the joint distribution, and thus, do not correspond to the parameters of the Sarmanov-Lancaster model in an intuitive way [9]. Goodman describes the exponential model for an $N$-variate distribution as follows. Let $\mathbf{X} = (X_1, \ldots, X_N)$ be a discrete random vector with joint probability distribution (probability mass function) $p_{\mathbf{X}}(\mathbf{x})$ and let $|\mathbf{X}|$ be the cardinality of the sample space of $\mathbf{X}$. Additionally, let $\hat{i} = (i_1, \cdots, i_j)$ such that $|\hat{i}| = j$. Assuming that $p_{\mathbf{X}}(\mathbf{x}) > 0$ for all $\mathbf{x}$, the log-likelihood function may be written as:

$$\log p_{\mathbf{X}}(\mathbf{x}) = \sum_{|i|=1}^{|\mathbf{X}|-1} \theta_i g_i(\mathbf{x}) - \psi(\theta).$$

(1.3)

The functions $g_i(\cdot)$ are specific to the model and are not necessarily orthogonal. The term $\psi(\theta)$ is a normalization term, chosen so that the probability function sums to one. For the three neuron case (trivariate binary distribution), we use the following version of the exponential model. Let $P[X_n = 1] = p_n$. The log likelihood may then
be written as

\[
\log p_\mathbf{x}(\mathbf{x}) = \sum_i \theta_i x_i + \sum_{i<j} \theta_{ij} x_i x_j + \theta_{123} x_1 x_2 x_3 - \psi(\theta). \tag{1.4}
\]

The \( \theta_i \) represent the effects of marginal events on the total probability, \( \theta_{ij} \) represent the effects of dependencies or interactions between pairs of random variables in the set and marginal probability effects, and \( \theta_{123} \) represents the effect of dependence among all three of the variables. Although modes of interaction between various subsets of the variables are represented by the different order parameters of the exponential model, these parameters do not represent truly separate orders of interaction as in the Sarmanov-Lancaster model: the parameters of the exponential model are not projections onto orthogonal bases, easily seen when one notes that the \( \psi(\theta) \) term is a function of all of the individual \( \theta \) parameters [9; 12]. Due to this lack of orthogonality, in the three neuron case, having \( \theta_{123} = 0 \) and \( \theta_{12} = 0 \) does not imply that neurons 1 and 2 (corresponding to variables \( X_1 \) and \( X_2 \)) are independent: these two terms do not represent or contain all of the interactions between neurons 1 and 2. Even the marginal probabilities \( p_i \) of the exponential model depend on all the parameters of the joint probability function in a nonlinear and complicated way, again emphasizing that the parameters of the exponential model cannot be interpreted as separate modes or orders of interaction. It should be noted, however, that when all of the \( \theta_{ij} \) are zero and \( \theta_{123} \) is zero, \( \{X_1, X_2, X_3\} \) are truly independent. That is, even though there is a one-to-one transformation between
the Sarmanov-Lancaster and exponential models for the full parametrization, setting certain parameters of either model to zero does not imply that similar order parameters in the other model will be zero. Despite the exponential model's lack of true separation of interaction orders, it proves to be significantly easier to use in certain applications, especially for parameter estimation, as is discussed in the next section.

1.1.3 Maximum Likelihood Estimation with the Sarmanov-Lancaster Model

When analyzing data with a parametric model, finding the maximum likelihood estimates of the parameters is almost always of interest. Specifically in our case, we are interested in finding maximum likelihood parameter estimates to use Normalized Maximum Likelihood-based model selection, detailed in Chapter 2. Thus, we would like to calculate ML parameter estimates for whichever parametric model we are using.

For the trivial case of a single neuron, the ML estimate of the neuron's firing probability within a bin is \( \hat{p}_{ML} = \frac{m}{M} \). That is, the ML estimate of a neuron's marginal probability is the number of times \( m \) a 1 is observed over the total number of trials \( M \). This result extends for \( N \) independent neurons such that each neuron's marginal probability is the number of times \( m_n \) the \( n^{th} \) neuron produces spikes divided by the total number of trials \( M \). Once we incorporate dependence into the Sarmanov-Lancaster model, however, things become complicated quickly. Even for
the simple case of 2 neurons, the maximum likelihood estimate of $\theta = (p_1, p_2, \rho)$ cannot be analytically calculated. However, we are able to use the invariance property of ML estimates to our benefit in certain situations. Namely, for $N$ neurons, the transformation between the $2^N - 1$ letter probabilities and the $2^N - 1$ parameters of the full-dependence model is one-to-one. Thus, we may simply find the ML estimate of the letter probabilities and transform the result to obtain the ML estimate of the Sarmanov-Lancaster parameters. It is well known that the ML estimate of the letter probabilities is simply the type, which is easily calculated [6].

While this method works in the case of the full-dependence model, i.e. the model in which we use all $2^N - 1$ parameters, we cannot employ this same transformation when we begin to analyze the data with different dependence models in which certain parameters are constrained to be zero. There is no current method for finding the ML estimate of the letter probabilities under the constraint that certain specified parameters of the Sarmanov-Lancaster model are zero, leaving us unable to use the invariance property of the ML estimate. It is possible to use numerical methods (such as Matlab's \texttt{fmincon}) to directly estimate the parameters of the Sarmanov-Lancaster model, but such methods often take a prohibitively long time to run. Additionally, we have found that such methods are not always able to obey the constraints set forth and return maximum likelihood estimates in which constrained parameters are incorrectly nonzero.
1.1.4 Maximum Likelihood Estimation with the Exponential Model

For the exponential model, the ML estimates of the parameters are analytically calculable for the independent and full-interaction models for any $N$ neuron population. However, analytically calculating the ML parameter estimates for constrained models where one or more of the parameters is constrained to be zero, proves to be difficult. One may not simply find the ML estimate for the full-interaction model $\hat{\theta}_{ML}$ and then zero out the components which are constrained to be zero. The solution must be directly found. We turn to the works of Amari and Nemenman for a solution.

In his recent paper, Amari studies structures of hierarchical systems of joint probability distributions with information geometry [2]. Without going into full detail of Amari's results, we simply state the main results of importance here. First, the exponential model described in the preceding sections is an example of a hierarchical system of joint probability distributions that forms a dually flat manifold in the sense described in detail by Amari. In the case of three binary random variables, the full-interaction exponential model has $2^3 - 1 = 7$ parameters $\theta = (\theta_1, \theta_2, \theta_3, \theta_{12}, \theta_{13}, \theta_{23}, \theta_{123})$, which creates a 7-dimensional manifold. Amari partitions the parameter vector $\theta$ into three distinct subsets such that $\theta = (\theta_1, \theta_2, \theta_3)$ with $\theta_1 = \theta_1, \theta_2, \theta_3$, $\theta_2 = (\theta_{12}, \theta_{13}, \theta_{23})$, and $\theta_3 = \theta_{123}$. That is, the parameter vector $\theta$ has been partitioned based on the three orders of interaction represented: marginal (or no interaction), pairwise and triple-wise interactions. Amari then de-
cribes sub-manifolds $E_k$ that consist of only those joint distributions that have at most $k^{th}$ order interactions but do not have intrinsic dependence or interactions of order higher than $k$. Each sub-manifold $E_k$ represents joint distributions having interactions only of the orders represented by $\theta_i, i \leq k$. So, $E_2$, for example, would include only joint distributions having interactions between at most 2 neurons and not having any triple-wise or third-order interactions, meaning that for $E_2$, $\theta_3 = 0$. He also describes sub-manifolds $M_k$ that consist of all joint probability distributions having the same $k^{th}$ order marginals as the full joint distribution. Thus, $M_2$ includes all joint probability distributions having the same second order, or pairwise, marginals as the full distribution. Amari proves that the sub-manifolds $M_k$ and $E_k$ are complementary and orthogonal at any point.

The fundamental results we take from Amari are as follows. Given a joint distribution $p$ that resides on the full joint distribution manifold, Amari defines the information projection of $p$ onto the sub-manifold $E_k$ as being the unique point $p_k$, where:

$$p_k = \arg\min_{q \in E_k} D(p||q).$$

Thus, $p_k$ is the point in $E_k$ that is closest to $p$ in terms of the Kullback-Leibler distance. It is clear, then, that $p_k$ is the maximum likelihood estimate of $p$ given the constraint that there are no interactions of order higher than $k$. Additionally, Amari states that the projection $p_k$ of $p$ onto $E_k$ also has maximum entropy among all distributions in $M_k$, that is, distributions having the same $k^{th}$ order marginals
as \( p \):

\[
p_k = \arg \max_{q \in M_k} H[q].
\]

Thus, \( p_k \) satisfies both the constraint that there are no interactions of order higher than \( k \) and also that all \( k^{th} \) order marginals are preserved.

These results are of direct importance in calculating the ML parameter estimates for constrained versions of the exponential model in which one or more of the parameters in the parameter vector \( \theta \) are zero. Nemenman summarizes these two results from Amari by stating that the maximum entropy distribution consistent with some marginals of the joint probability distribution introduces no statistical interactions beyond those present in the marginals. A solution of any maximum entropy problem with marginal constraints has the form of a product of terms depending on the constrained variables [11]. For example, if the constraint for the three neuron case is \( \theta_{123} = 0 \), then the maximum entropy distribution satisfying the second-order marginal constraints has the form:

\[
p(x_1, x_2, x_3) = a(x_1, x_2) \times b(x_1, x_3) \times c(x_2, x_3).
\]

It should be noted that satisfying the second-order marginal constraints automatically satisfies the individual marginal constraints. In general, no analytic solution for the functions \( a, b, \) and \( c \) exists, but an algorithm called the iterative proportional fitting procedure (or IPFP) converges to the true solution for the maximum entropy
distribution [7; 8]. IPFP iteratively adjusts a trial solution to satisfy each of the marginal constraints one at a time. All marginals of the original distribution $p$ are preserved except those marginals involving all of the variables being analyzed for interaction. Thus, if we constrain $\theta_{12} = 0$, all marginals are preserved except for $p_{12}$.

With the results from Amari and Nemenman, we are now in a position to find the ML estimates of the parameters of the exponential model in cases when certain parameters are constrained to be zero. From Amari, we know that the maximum likelihood distribution under the interaction-order constraints minimizes the KL distance to the full (unconstrained) maximum likelihood distribution (also known as the type). Additionally, this same ML constrained distribution maximizes the entropy among all distributions satisfying the marginal constraints. Thus, we may use IPFP to calculate our constrained maximum likelihood distribution.

Unlike the Sarmanov-Lancaster model, we are able to take advantage of the invariance of the ML estimate for the exponential model's parameters even for constrained models. For $N$ neurons, the transformation between the $2^N - 1$ letter probabilities and the $2^N - 1$ parameters of the full-interaction exponential model is one-to-one. Thus, we may simply find the ML estimate of the letter probabilities and transform the result to obtain the ML estimate of the exponential parameters. We can employ this same transformation when we begin to analyze the data with different dependence models in which certain parameters are constrained to be zero:
we simply calculate the ML estimate of letter probabilities (equivalent to finding the constrained maximum likelihood distribution via IPFP) and transform back to the exponential model parametrization. As certain parameters have been constrained to zero in finding the ML distribution, the end result of the transformation will zero the appropriate parameters. We have a fast and reliable method for finding the ML parameters for further use in Normalized Maximum Likelihood model selection.
Chapter 2

Minimum Description Length

The Minimum Description Length, or MDL, Principle is a valuable tool for model selection problems. We will specifically use an MDL-based approach to choose between models of the same family with differing numbers of parameters representing the presence (or absence) of interactions among neurons in a population.

MDL takes the view that data can be compressed, and that the model that compresses the data best corresponds to the best fitting model. MDL does not postulate that the model chosen is the true underlying model generating the data or even that such a true model exists. Rather, MDL states that by compressing the data efficiently with a given model, we have learned more about the regularities present in the data [10]. MDL also does not assume that a so-called true data-generating distribution lies in any of the model families under consideration, a fundamental assumption that other methods (such as hypothesis testing) implicitly make.

MDL is a model inference strategy that chooses models in a manner that trades
off between goodness of fit and complexity, thereby implementing a form of Occam’s Razor and protecting against the perils of overfitting. In the face of a small number of data samples, MDL-based model selection tends to favor simpler models, a phenomenon we will discuss towards the end of this chapter.

MDL may be used to estimate both the structure of a model, i.e. the number of parameters in a parametric model, and the values of parameters themselves. MDL parameter estimation, however, in many cases results in the maximum likelihood parameter estimate within a chosen model structure [10]. In this thesis, we are concerned more with using MDL to estimate model structure.

2.1 Data Compression and MDL Codelengths

Underlying MDL’s view of learning as data compression is the concept of uniquely mapping data to a description [10]. The shorter the description is, the better the data has been learned by the model producing the description. Idealized MDL was extended from the concept that the best model for a given sequence of data may be identified with the shortest program which prints the data, a concept directly related to the concept of Kolmogorov Complexity [10]. In this thesis, however, we are concerned with non-idealized MDL and its practical applications in the specific form of Normalized Maximum Likelihood, or NML, methods.

Before proceeding, I would like to mention two pitfalls encountered in modeling data. First, there is the danger of overfitting data. In the extreme case, one can
always describe the data with a model containing as many parameters are there are data samples. This method, of course, does not reveal anything about any underlying regularity (if any) in the data. Second, even when one finds a model that describes a given dataset well, it may not work as well for describing subsequent data or other datasets. That is, there may be no model that describes all data having regularity equally well, which is another way of stating a Kolmogorov Complexity result that there exists no computer program that for every set of data, when given that set of data as input, returns the shortest program that prints the data [10; 17]. MDL attempts to model data while avoiding these pitfalls.

2.1.1 Two-Part Codes for MDL

Original MDL methods used a two-part codelength to choose among candidate models. The first portion of the codelength represents the length (in bits) of the description of a point in the model-space, also thought of as a specific hypothesis from the model family being considered. The second portion represents the length of the description of the data as encoded (or viewed) with this hypothesis. Thus, the best hypothesis (from any model family) to describe the data is the one that results in the shortest two-part codelength, i.e., the smallest sum of the first and second portions described above. Additionally, the model, or parametric family, containing this hypothesis is the best model to explain the data [10].

In the case of two-part MDL, it is easy to see the trade-off between model complexity and goodness-of-fit. Often, a model (and thus, a hypothesis within the
model-space) we are considering is very simple, meaning that the first part of the
two-part codelength, the description of the hypothesis alone, will be very small.
However, such a simple model may not describe (or compress) the data very well,
leading to a large codelength for the description of the data as encoded by the
hypothesis. The converse is also true many times – a complex model will often have
a large model description length but a short data description length. To see this,
recall that one can always overfit the data such that the hypothesis \textit{perfectly} fits the
data at hand, but is usually \textit{out of hand} in terms of complexity.

It should be noted that MDL is only concerned with codelengths, and not en-
codings themselves. Additionally, MDL is concerned only with the length of prefix
codes, but that does not cause any loss of generality [6].

Use of the two-part codelength version of MDL proves to be problematic, how-
ever. It is simple enough to define the second part of the codelength, the length
of the description of the data as encoded by the hypothesis. For a hypothesized
probability distribution, it is always feasible to code the data (as viewed through
the hypothesis) with a code having lengths \( L(D|H) = -\log P(D|H) \), where \( P \)
is the probability mass function over the data \( D \) given the hypothesis \( H \). However,
efficiently describing the complexity of the hypothesis \( H \) via \( L(H) \), the first part
of the two-part codelength, proved to be a challenge for early proponents of MDL
including Rissanen and Barron. The difficulty in creating a non-arbitrary and yet
practically feasible method for calculating two-part codelengths for data motivated
the development by Rissanen of one-part codelengths [18].

2.1.2 MDL in Practice

MDL, as it is used today for practical model-selection, came about with the use of universal codes for a one-part codelength, developed by Barron, Rissanen, and Yu [1]. Instead of choosing a model based on the performances of specific hypotheses within the model, Barron, Rissanen, and Yu choose a model based on properties of the model as a whole. The code corresponding to an encoding based on the entire model is designed to be a universal code. The codelength of this universal code is called the stochastic complexity of the data as viewed through the model. Additionally, this universal code may be mapped to a universal model, the probability distribution corresponding to a universal code.

Rissanen defines the parametric complexity of a parametric model. Grunwald refers to the parametric complexity as a measure of the richness of the parametric model, indicating the model's ability to fit random data [10]. Grunwald provides several interpretations of the parametric complexity. We will not discuss all of those interpretations here, but will mention two of his main points. First, he states that the parametric complexity of a model is the logarithm of the number of essentially different, distinguishable point hypotheses within the model. Additionally, he notes that for a finite model $M$ and for a large number of data samples, the parametric complexity is essentially the log of the number of distributions in $M$ [10].

Rissanen's stochastic complexity is equal to the sum of the parametric complexity
of the model and the minimum of the codelengths of the data as viewed through the various hypotheses of the model we are examining. When the codelengths comes from a code with lengths $L(D|H) = -\log P(D|H)$, the minimum codelength will be achieved by the maximum-likelihood parametric hypothesis $\hat{H}$ in the model. Thus, the stochastic complexity of the data given the model is equal to the sum of $L(D|\hat{H})$ and the parametric complexity of the model. Model selection techniques using this form of MDL choose the model that has the smallest stochastic complexity given the data at hand $D$. The trade-off still exists between goodness of fit (in the codelength term) and complexity of the model (in the parametric complexity term).

2.1.3 Normalized Maximum Likelihood and Regret

Barron, Rissanen and Yu discuss the concept of regret as a way to measure the performance of a universal model $\bar{P}$ relative to a model family $M$ containing probabilistic distributions [1; 10]. It should be noted that $\bar{P}$, a probability distribution over the data-outcome space $D^n$, may not reside in $M$. For observed data sequence of length $n$, $d^n$, the regret of $\bar{P}$ relative to $M$ is

$$-\log \bar{P}(d^n) - \min_{P \in M} \{-\log P(d^n)\}$$

Thus regret measures the extra bits needed to encode the data we have seen with the universal model as opposed to the best fitting distribution in the model family. Since we are concerned in this thesis with parametric model families, we may assume that
there exists for model $M$ a single $\hat{\theta}(d^n)$ that is the maximum likelihood parameter (or parameter vector) for observed data $d^n$. Consequently, the equation becomes

$$ - \log \bar{P}(d^n) - \{ - \log P(d^n|\hat{\theta}(d^n)) \}. $$

Barron, Rissanen and Yu look at the worst-case regret over all possible $n$-length data sequences $d^n$ and would like to find the universal model which minimizes this worst-case regret, called $R_{max}$. That is, the optimal universal model relative to model family $M$ for a given sample size $n$ is the distribution minimizing (over all possible distributions on $D^n$)

$$ \min_{\bar{P}} R_{max}(\bar{P}) = \min_{\bar{P}} \max_{d^n \in D^n} \{ - \log \bar{P}(d^n) - \{ - \log P(d^n|\hat{\theta}(d^n)) \} \}. $$

Shtarkov solved this minimization under a specific condition on the parametric complexity of a model $M$ [20]. Specifically, if

$$ Complexity(n, M) \equiv \log \sum_{d^n \in D^n} P(d^n|\hat{\theta}(d^n)) < \infty $$

then the minimax regret as defined above is uniquely achieved for the distribution $\bar{P}_{nml}$ given by

$$ \bar{P}_{nml}(d^n) := \frac{P(d^n|\hat{\theta}(d^n))}{\sum_{x^n \in D^n} P(x^n|\hat{\theta}(x^n))}. $$
This minimax optimal probability distribution, known as the normalized maximum likelihood distribution, achieves regret equal to the parametric complexity of the model regardless of what the observed data is. $\tilde{P}_{nml}$ produces the only complete prefix code with this property, effectively giving all distributions in $M$ equal footing. Grunwald provides a simple proof of why the maximum regret achieved by $\tilde{P}_{nml}$ is indeed the overall minimax regret [10].

It should be noted that for any universal model $\tilde{P}$ relative to a model family $M$, for every $P \in M$, the largest codelength difference $\max_{d^n \in D^n} - \log \tilde{P}(d^n) + \log P(d^n)$ increases sublinearly in $n$ (and, for finite $M$, it is in fact bounded by a positive constant) [10].

### 2.2 NML and Model Selection

The minimax optimal universal model $\tilde{P}_{nml}$ can be used in model selection as follows. Given data $d^n = (d_1, ..., d_n)$, and a finite number of parametric models $M_i$ to choose among (each having finite parametric complexity), pick the model that maximizes the normalized maximum likelihood given the data. That is, maximize

$$\tilde{P}_{nml}(d^n | M_i),$$

or, equivalently, minimize

$$- \log \tilde{P}_{nml}(d^n | M_i) = - \log P(d^n | \hat{\theta}_i(d^n)) + \text{Complexity}(n, M_i).$$
Thus, we are picking the model which has the smallest stochastic complexity given the data we see \( d^n \), in essence picking the model which minimizes the codelength of the data. Grunwald discusses an extension to NML for choosing among an infinite number of models, but we are only concerned with selection among a finite number of models in this thesis [10].

2.2.1 Asymptotic Formula for the Normalized Maximum Likelihood Model

The stochastic complexity of the normalized maximum likelihood universal model is often difficult to evaluate even numerically, owing to the difficulty in calculating the \( \text{parametric} \) complexity of the NML model. Kontkanen has proposed a fast algorithm for evaluating the parametric complexity for conditionally independent multinomial variables [14]. While this has proven to be a successful model class for many applications, it does not provide insight in our neural population setting, in which the neurons (or subsets of neurons) in a population could be correlated. It is precisely the presence and absence of these correlations that we seek to determine, so it is clear that the independence assumptions required by Kontkanen’s methods cannot assist us.

What is of use in the neural setting is an asymptotic formula for the NML model. Under certain conditions, for a \( k \)-dimensional parametric model \( M \) with parametrization \( \Theta \), the parametric complexity (as the sample size \( n \) approaches
\[ \text{Complexity}(n, M) = \frac{k}{2} \log \frac{n}{2\pi} + \log \int_{\theta \in \Theta} \sqrt{|I(\theta)|} d\theta + o(1), \]

with \( o(1) \) meaning it goes to zero as \( n \) approaches infinity. The quantity \( k \) is the number of parameters in the model, and \( |I(\theta)| \) is the determinant of the \( k \times k \) Fisher information matrix \( I \) evaluated at \( \theta \). This asymptotic equation holds only under the following circumstances, discussed by Grunwald.

1. The parametric complexity of \( M \) is finite, as is the integral of the Fisher information term.

2. The maximum-likelihood parameter estimate remains sufficiently within the boundaries of \( \Theta \) for all large \( n \).

Since we are dealing with neural population data, as stated above, the parametric complexity will be finite. There exist further conditions for the usability of the asymptotic form, but these are satisfied if \( M \) is an exponential family, which is true for the Sarmanov-Lancaster and the exponential models [5].

Grunwald states that experiments suggest that for exponential-like models, the approximation given by the asymptotic equation above is reasonably accurate for small numbers of data samples. Additionally, as \( n \) grows very large, we know that the codelongth term \(- \log P(d^n|\hat{\theta}(d^n))\) typically grows linearly while the parametric complexity term for a fixed \( k \)-dimensional parametric model generally grows only
logarithmically [10]. That is, for large $n$, the portion of the stochastic complexity due to the parametric complexity term (containing the Fisher information integral) is much smaller than the portion due to the data-encoding codelength term, which we see in figures 2.1 and 2.2. When $n$ is large enough, the term containing the Fisher information integral can thus be ignored in selecting between models. Finally, it should be noted that the asymptotic formula for the NML model should ideally be used only when the number of parameters in the models of interest are much less than the number of data samples.

2.2.2 Application of NML

We shall now discuss our application of Normalized Maximum Likelihood-based model selection methods in the context of the exponential (or log-linear) and Sarmanov-Lancaster model families discussed in the previous chapter. To choose between various parametric models within each of these model families, we analyze the data through the "lenses" of the different models. We do this by calculating the stochastic complexity of the data under the view of each of these models. That is, for the Sarmanov Lancaster model family we calculate the stochastic complexity for the independent model and fully dependent model, for example. For the exponential model, we are able to calculate the stochastic complexity for the various constrained models (in which certain parameters are assumed to be zero) in approaching the data. Thus, we turn now to the task of calculating stochastic
complexity. The stochastic complexity is, asymptotically,

\[- \log P(d^n | \hat{\theta}(d^n)) + \frac{k}{2} \log \frac{n}{2\pi} + \log \int_{\theta \in \Theta} \sqrt{I(\theta)} d\theta + o(1).\]

Though with neural data we are not in the realm of large sample size \(n\) (which is where this asymptotic formula is meant to apply), calculating the parametric complexity directly for several Bernoulli random variables is often intractable, even for any reasonable number of variables, and additionally, there are currently no algorithms for calculating the stochastic complexity in the case of models having possibly correlated variables, as we have with neural populations. Thus, we are left to analyze how the asymptotic formula for stochastic complexity performs even in non-asymptotic regimes to understand how NML-based model selection fares with limited data.

The first term of the stochastic complexity, \(- \log P(d^n | \hat{\theta}(d^n))\) is the negative log likelihood function of the length-\(n\) data using the ML estimates of the parameters in the model we are considering. For the exponential family, this term is always calculable using the Iterative Proportional Fitting Procedure detailed in the previous chapter. For the Sarmanov-Lancaster model family, this term is only readily calculable for the full-dependence and full-independence models, thus limiting our analysis to these models for this family. The second term of the stochastic complexity is simply a function of the data length \(n\) and the number of parameters \(k\) in the model being analyzed. The term involving the Fisher information,
\( \log \int_{\theta \in \Theta} \sqrt{I(\theta)} d\theta \), proves to be very difficult to compute in all but very specific model types. For a model (in either the exponential or Sarmanov-Lancaster family) in which we assume the \( K \) variables are fully independent, i.e. all interaction or dependence parameters are constrained to be zero, the value of the Fisher information term is \( \log(\pi^K) \). The Fisher information term is also readily calculated to be 

\[
\frac{\pi^{(k+1)/2}}{\Gamma((k+1)/2)}
\]

for the full-interaction (for the exponential family) or full-dependence (for the Sarmanov-Lancaster family) model, where \( k = 2^K - 1 \) for \( K \) neurons [14]. This result for the full-interaction or dependence model is calculable as a direct result of the invariance of the Fisher information term to reparametrization, detailed below. Finally, the last term in the asymptotic formula, the \( o(1) \) term, is shown to be much smaller than any of the other terms and may be safely ignored even when the number of data samples is not very large. This is easily calculated in the case of a single neuron and two neurons, and shown in figures 2.5, 2.6, and 2.7, but not easily calculated for reasonable sample sizes for models having 3 neurons.

### 2.2.3 Invariance of the Fisher Information Term

The Fisher information matrix under the \( \theta \)-parameterization is 

\[ I(\theta) = -E(\nabla_\theta^2 \log P(d^n|\theta)). \]

If we then reparametrize to the \( p \)-parametrization, which is the result of a one-to-one mapping from \( \Theta \)-space to \( P \)-space, we have by the vector chain rule that 

\[ \nabla_p = J^T \nabla_\theta, \]

where \( J \) is the Jacobian matrix \( J = \frac{\partial \theta}{\partial p} \) with components \( J_{ij} = \frac{\partial \theta_i}{\partial p_j} \). Thus, the Fisher information matrix \( I(p) = J^T I(\theta) J \). Taking the determinant, we see 

\[ |I(p)| = (|J|)^2 |I(\theta)|, \]

meaning that 

\[ \sqrt{|I(p)|} = \sqrt{|I(\theta)|} |J|, \]

and
\[ \int_{p \in P} \sqrt{I(p)} dp = \int_{\theta \in \Theta} \sqrt{I(\theta)} \frac{\partial \theta}{\partial p} dp, \]
which is simply \[ \int_{\theta \in \Theta} \sqrt{I(\theta)} d\theta, \] showing the invariance property of the Fisher term when the mapping between the parameter spaces is one-to-one.

In our case, the \(2^K - 1\) parameters of both the full-interaction exponential model and the full-dependence Sarmanov-Lancaster model are a one-to-one function of the \(2^K - 1\) letter probabilities of the response alphabet. For example, for 3 neurons, there are eight letter probabilities (000, 001, 010, etc.), seven of which are true parameters and the eighth being constrained such that the probabilities sum to one. Because the Fisher information term is invariant to this one-to-one reparametrization, we may calculate this term easily for either the full exponential or Sarmanov-Lancaster model using the much simpler letter-probability parametrization. For constrained models of either model family, however, we may not use the same technique. This is because the letter-probability parameter space is always of dimension \(2^K - 1\) (because there is no way to “reduce” this parametrization of observed letter probabilities), and thus it may not be mapped in a one-to-one manner to the parameter space of a constrained model that has less than \(2^K - 1\) parameters. In this case, the Jacobian matrix would not be square, and the parametrization-invariance property of the Fisher information term does not hold. Thus, in our approach to NML-based model selection in which we choose between various constrained models of the exponential family, we are not able to consider the Fisher information term of the stochastic complexity. We see that the Fisher information term, like the \(o(1)\) term, may be
dropped in model selection if we are willing to increase the risk of choosing the wrong model.

2.2.4 The Exponential Model Family and NML

For the exponential model family, we may analyze and choose between various constrained versions of a full-interaction model if we ignore the (incalculable) Fisher information and $o(1)$ terms for the constrained models. For $K$ neurons, there are $k = 2^K - 1$ free parameters in the full-interaction exponential model. However, with the aid of the iterative proportional fitting procedure (IPFP) detailed in the previous chapter, we may also analyze so-called constrained models to see which model fits the data best in the sense of minimizing regret (without the Fisher information regret term). We may calculate the first term of the stochastic complexity, the likelihood term, in a straightforward manner for the full-interaction and zero-interaction (or independent) models, and with the aid of IPFP, we are also able to calculate this likelihood for constrained models as well since IPFP provides a way of calculating the ML estimates of the parameters under these constraints. The second term of the stochastic complexity is very simple and is smaller in models with parameters constrained to be zero. We must ignore the incalculable Fisher information term in the constrained cases. And finally, the last term, the $o(1)$ term proves to be very small in relation to the other terms and also may be ignored in choosing between models. Thus, for the exponential model family, we may choose between a full-interaction model and various models in which certain interaction parameters are
constrained to be zero. Although setting certain interaction parameters to zero does not imply independence of the variables governed by the parameters, as discussed in Chapter 1, it allows us to have a better understanding of what type of exponential model interactions we are encountering in the data.

2.2.5 The Sarmanov-Lancaster Model Family and NML

For the Sarmanov-Lancaster model family, we are not only unable to calculate the Fisher information term for so-called constrained models (as with the exponential model), but we also are not able to obtain maximum likelihood parameter estimates for constrained versions of the full-dependence model, as described in Chapter 1. Namely, there is not a tool such as IPFP for the Sarmanov-Lancaster model family. We may only analyze and choose between a full-dependence model and a full-independence model. In these two cases, we are able to calculate the Fisher information term as described above. For \( K \) neurons, there are \( k = 2^K - 1 \) free parameters in the full-dependence Sarmanov-Lancaster model and \( K \) free parameters in the full-independence model. As with NML-based model selection in the exponential model, we calculate the stochastic complexity of the independent and dependent models given the data at hand and choose the model, either "fully independent" or "fully dependence", which minimizes the regret.
Figure 2.1: Stochastic Complexity terms for (a) Two neuron dependent model (b) Two neuron independent model. These were calculated using the true parameters \( p_1 = .25 \), \( p_2 = .4 \), and \( \rho = 0 \). Even with as few as 10 samples, it is clear that the first two terms of the asymptotic expansion of the stochastic complexity far outweigh the Fisher information and \( o(1) \) terms.

Figure 2.2: Stochastic Complexity terms for (a) Three neuron model with interactions (b) Three neuron full independence model. These were calculated using the true parameters \( \theta_1 = -1.785, \theta_2 = -1.785 \), and \( \theta_3 = -1.5 \) with all interaction parameters set to zero. Even with as few as 10 samples, it is clear that the first two terms of the asymptotic expansion of the stochastic complexity far outweigh the Fisher information and \( o(1) \) terms.
Figure 2.3: This plot shows the probability of error achieved by employing NML-based model selection (with the asymptotic formula for stochastic complexity without the Fisher information or o(1) terms). The probabilities were simulated using 10000 trials. The parameter vector in each of the cases was $\theta = [\theta_1, \theta_2, \theta_3, \theta_{12}, \theta_{13}, \theta_{23}, \theta_{123}]$, being set to (a) $[-1.9750 -1.9750 -1.9750 1 1 1 0]$, i.e., all 2-way interactions are present in the model (b) $[-1.7850 -1.7850 -1.5000 1.1960 0 0 0]$, i.e., only the 2-way interaction between variables 1 and 2 is present, and in (c) $[-1.7850 -1.7850 -1.5000 0 0 0]$, i.e., no interactions exist among any of the variables. In (d), we see an overlay of all of these error probabilities. It becomes clear from looking at (d) that with a limited amount of data (less than 100 data samples) in the three neuron case, our NML-based method performs better for "simpler" models, but in any case still does not achieve error probabilities less than 0.1 til after 400 samples are available, which is rarely the case with neural data.
Figure 2.4: This plot shows the probability of error achieved by employing NML-based model selection (with the asymptotic formula for stochastic complexity). Since we are only dealing with 2 neurons and the simplified situation of a full independence model and a full dependence model (as the 2 neurons are simply either correlated or not), we are able to use the known Fisher information and $o(1)$ terms here. The probabilities were simulated using 10000 trials. The true parameter vector is in (a) $p_1 = p_2 = .25$, with various values of $\rho$ being displayed, and in (b) $p_1 = .25$, and $p_2 = .025$, with various values of $\rho$ being displayed. Note that until a critical amount of data is available, our NML-based method tends to favor the independence model whether it is the true model or not – an embodiment of the parsimonious tendencies of NML.

2.3 Some Observations on NML

NML-based model-selection coincides with Bayes factor model selection when using a non-informative prior, namely, the Jeffrey’s prior. However, MDL and Bayesian inference are very different in philosophy and in practice outside of this specific choice of distribution. The same difference in philosophy applies to NML and use of the Bayesian Information Criterion, which asymptotically coincides with the first two terms of the stochastic complexity of the NML distribution. Additionally, a so-called modified maximum likelihood plug-in distribution asymptotically performs the same as the NML distribution [10].

While we have found the minimax optimal distribution relative to a given model
**Figure 2.5:** Here, we see that for a model having two uncorrelated neurons (or Bernoulli random variables), the $o(1)$ term decays as the amount of data samples increases. Even after having just 20 samples, the relative importance of this term is quickly diminished. The maximum value of this term (for just 1 data sample) is 0.9347, whereas the Fisher information term is $\log(\pi^2)=2.2895$.

family $M$, there is no indication that our choice in $M$ is a good one, and there may always exist an unknown model which will better compress the data we have. However, if the data are generated by some *true* distribution which is contained in one of the model families we are considering, then NML-based inference presents a rational way of choosing the model containing this true distribution [4; 22]. The main contribution that we take away from NML-based model selection is the ability to choose a parsimonious model class (or parametric model family) to describe the data available to us. In the next chapter, we will discuss how to evaluate whether the choice that NML-based model selection has made is a reasonable choice in light
Figure 2.6: Here, we see that for a model having two correlated neurons (or Bernoulli random variables), the $o(1)$ term decays as the amount of data samples increases. Even after having just 20 samples, the relative importance of this term is quickly diminished. The maximum value of this term (for just 1 data sample) is $1.8537$, whereas the Fisher information term is $\log(\pi^2)=2.2895$.

of the data at hand. Additionally, we will discuss how to analyze models within the model class selected by NML-based inference.
Figure 2.7: Here, we see that for a model having a single neuron (or Bernoulli random variable), the $o(1)$ term decays as the amount of data samples increases. Even after having just 20 samples, the relative importance of this term is quickly diminished. The maximum value of this term (for just 1 data sample) is 0.4674, whereas the Fisher information term is over 3 times larger, at $\log(\pi)=1.14475$. 
Chapter 3

Model Selection

3.1 Introduction

While Normalized Maximum Likelihood-based model selection is able to select a parametric model class for a given dataset, it does not provide a way to determine if the class selected is a reasonable one. That is, NML-based model selection simply choose the best model class among available model classes (i.e., model classes that are provided) and does not make a judgment about the range of possible models within this model class that can describe the data well. Ideally, we would like the data we have to reveal the best model within a parametric model class for the population response. If a model for the population response were somehow available, the amount of data needed would not necessarily follow Weinberger’s result. Essentially, a model puts available data in context and allows us to focus our data analysis. When we have a parametric model, we can find the maximum likelihood
estimate of the parameters, which will give us a single model most consistent with the data within those models having the assumed parametric form. It would still be unknown, however, whether any of these models actually describe the data well. In this chapter, we suggest a different approach to population data analysis where we will consider all models within a model class (chosen by NML-based methods) and find those models most likely to have produced the data at hand.

3.2 Kullback-Leibler-based Model Analysis

The joint probability of a measured neural response $\prod_{m=1}^{M} P(R_m)$ can be written in a simple way using the theory of types [6].

$$P(R) = \prod_{m=1}^{M} P(R_m) = \exp \left\{ -M \left[ H(\hat{P}) + D \left( \hat{P} \parallel P \right) \right] \right\} \quad (3.1)$$

where $\hat{P}$ denotes the type of the data. Result (3.1) implies that the data measurements are completely summarized by the type and that the type is all that is needed in subsequent analysis. $H(\hat{P})$ is the entropy of the type [6] and $D \left( \hat{P} \parallel P \right)$ is the Kullback-Leibler distance [13; 15] between the type and a probability distribution that describes how the data were produced.

The key idea for our approach to data analysis rests on the fact that any statistical model, $P$, of how the data were produced, whether it is right or wrong, has a probability of having produced the data. Assuming we have many possible models, we augment the notation for the response probability by a parameter vector $\theta$
to indicate that possible probability models within a given parametric model class result from specific parameter choices.

\[ P(\mathbf{R}; \theta) = \exp \left\{ -M \left[ \mathcal{H}(\tilde{P}) + \mathcal{D} \left( \tilde{P} \| P(\theta) \right) \right] \right\} \]  

(3.2)

Clearly, the maximum likelihood choice of model parameters is

\[ \hat{\theta}_{\text{ML}} = \arg\min_{\theta} \mathcal{D} \left( \tilde{P} \| P(\theta) \right) \]

It is important to note that the minimum value of the Kullback-Leibler distance is usually not zero, unless the parametric model class contains the type. For example, a possible model class one might use would describe the responses of two neurons as being statistically independent. This model class would have two parameters: the two single-neuron response probabilities. Even if the statistically independent model class were correct, the measured joint response probabilities would only rarely be exactly consistent with the independence model class. As the number of measured responses increases, the fit would improve, and the minimum value of the Kullback-Leibler distance between the measured type and the model class would decrease. We can see this is the case in figure 3.1.

The smallest the KL distance can be is zero, corresponding to an exact fit between data and model, when the model class under consideration contains the type. Therefore, the theoretical maximal value of the likelihood function (occurring when
Figure 3.1: We generate Bernoulli data and analyze it with a geometric model class, a Poisson model class, and a binomial model class by calculating the minimum KL (for a given amount of data) between the data and the “best” model in each model class. The distance between this best model and the type of the data is the minimum KL distance. It is not always zero even when the type is a member of one of the model classes. Also, it should be noted that as the number of data samples increases, the minimum value of the KL distance decreases as the fit improves. Finally, and most importantly, the minimum KL is smallest for the correct model class (the binomial), providing us a way to assess the viability of a parametric model class.

there is an exact fit between the data and the model) corresponds to the entropy term. We define the operator

\[ t_{\theta}^\max P(R; \theta) = \exp \left\{ -\mathcal{H}(\hat{P}) \right\} \]

The ratio of the model’s probability function (evaluated at the population’s response measurements) and the theoretical maximum can be interpreted as the relative probability that the measurements are consistent with the given model. The more
this model disagrees with the data, the smaller this relative probability will be. By accepting those models for the data having a relative probability greater than some threshold value of our choosing, we can determine all models consistent with the data. Because this ratio is related only to the Kullback-Leibler term in (3.2), this computation amounts to

\[ -\log \frac{P(R; \theta)}{\max_\theta P(R; \theta)} = MD \left( \hat{P} \parallel P(\theta) \right) \leq -\log P_0 , \]  

where \( P_0 \) is the threshold value for the relative probability. We can thus search for all parametric models consistent with the data by computing the Kullback-Leibler distance between the type derived from the measurements and all models in the assumed parametric model class. The region of parameter space for which the Kullback-Leibler distance is sufficiently small describes likely population models. We call this region the \textit{model feasibility region}: all models within this region have a relative probability of producing the measurements that is greater than the chosen threshold value \( P_0 \). The smaller this probability threshold, the greater the number of models that we want to consider. Note that even the maximum likelihood model within a model class may not yield a sufficiently small Kullback-Leibler distance; in this case, all models in this particular parametric class fail to describe the data sufficiently well (even though we are still able to calculate our ML estimate of the "best" model). In this manner, we can accept or reject models within a given class. We may also compare the quality of our choice of parametrization by looking at
the model feasibility region under each choice. For a given dataset, threshold, and number of parameters, a model class with a larger feasibility region is interpreted to be a more likely descriptor of how the data were produced.

3.3 Examples

Figure 3.2 illustrates our approach to classifying models. The plots show model feasibility regions for two model classes that can describe the single-bin response of a two-neuron population. In the simulation, the response probabilities for a neural spike were both 0.2, and the correlation coefficient was 0.5. In the left column, $M=20$ responses were used in the analysis; in the right, one hundred. For each dataset, the type was accumulated, and the Kullback-Leibler distance between the type and all models in the model class were computed. The parameter values of models likely to have produced the data are enclosed within the indicated boundaries. In the top row, the neural responses were analyzed using a model class that considered the neurons to be independent, and the model class parameters consisted of the two neurons’ response probabilities $p_1$ and $p_2$. In the left column, we also show with the blue elliptical contour the error bounds induced by the Fisher information matrix. The cross indicates the maximum likelihood estimate and the irregular green contour encloses the parameter values for those models consistent with the data at a threshold probability $P_0 = .0000454$. No contour appears in the right column because no model was consistent with the larger dataset at the required
probability level.

In the bottom row, the model class allowed the responses to be correlated. Here, \( \theta = \{p_1, p_2, \rho\} \), where \( \rho \) is the correlation coefficient. Here, surfaces enclose parameter values for models likely to have produced the simulated data at the same relative probability level as used in the top row. As the left panel shows, the surface may not be entirely closed. In the right panel, the parameter surface shrinks because more data are available, but it does not vanish. The maximum likelihood parameter estimates are indicated by the intersecting lines. For the left panel, they were \( \hat{\rho}_1 = 0.25 \), \( \hat{\rho}_2 = 0.1 \), and \( \hat{\rho} = 0.55 \); on the right, \( \hat{\rho}_1 = 0.21 \), \( \hat{\rho}_2 = 0.17 \), and \( \hat{\rho} = 0.5 \). It should also be noted that using NML-based model selection on the same datasets yields the choice of the dependent model class (bottom row) as being more descriptive of the data in both cases. This is consistent with the fact that for the independent model class, with more data, there was no set of models consistent with the data, and thus, no contour appeared. In other words, the contours shown here agreed with the model class chosen by NML.

This simulation shows that this approach of finding all the likely models has several interesting properties. First of all, as the number of responses increases, the boundary shrinks, which restricts the models capable of producing the data. In the top row, which used an incorrect model class, no boundary existed when enough responses were present because the Kullback-Leibler distance exceeded the threshold in all cases: no model that assumed independent neural responses could have
produced the data at the specified probability level. Despite the model class misfit, note that the maximum likelihood estimate of the response probabilities in this case is quite accurate. Consequently, just computing the maximum likelihood estimate and judging parameter estimate precision with the Cramér-Rao bound would not have revealed the model class' inadequacies. Secondly, note that the boundary is not ellipsoidal, as in the case of the blue quadratic form of the Fisher information, a Gaussian-motivated quantity similar to mean-squared error. Furthermore, the maximum likelihood parameter estimate, which always corresponds to the most likely model, is not necessarily located in the center of the boundary. This is also in contrast to the quadratic form of the estimation error given by $\varepsilon F\varepsilon'$, where $\varepsilon$ is the error incurred between the maximum likelihood estimate of the parameters and their true value and $F$ is the Fisher information matrix for the parameters. We see more of this difference between the contours of our method and the Fisher information-based contours in figure 3.3. In Figure 3.4, we see the same analysis conducted on a real dataset taken from a two-neuron population in the crayfish visual system.

### 3.3.1 NML and Our Approach to Model Selection

We will now discuss how to use the tools of NML and our KL-distance-based approach to select a model for neural data. In a typical case, a researcher might have, in a single time bin, on the order of 30 samples of data from 30 stimulus repetitions. While one may always choose a model class that might seem reasonable and calculate maximum likelihood estimates of the parameters of this model class
Figure 3.2: This figure shows a comparison of model feasibility regions for two different model classes for a two-neuron population: the independent and dependent model classes, shown in the top row and bottom rows, respectively. Additionally, the contours in the left column was calculated using only 20 data samples, and those in the right column using 100 data samples (from the same larger dataset).

Based on the data, this does not always lead to the choice of the best model class and model within the class. For example, whether one assumes the data came from two independent neurons or two dependent neurons (i.e., having correlated firings), the ML estimates of the neurons' marginal probabilities will be the same in either case. Thus, one may not simply rely on the ML estimate as a way to verify model
Figure 3.3: In the top-left figure, we simulated a binomial dataset of size 30, with \( p_1 = .9, p_2 = 9, \) and \( \hat{p}_1 = \hat{p}_2 = 0.9 \) as well. Using the same threshold for our method and for the Fisher-information-based error contour, we see that the Fisher-based contour is always an ellipsoid centered around the ML parameter estimate, even if the ellipse encompasses unallowable models (such as marginal probabilities greater than 1). In the top-right figure, we see that with 300 data samples, the two contours (with similar but not identical threshold values) do appear to track each other. Finally, in the bottom figure the threshold for Fisher-based contour was a fraction of the threshold for our contour. As expected, our contour (the green) was much larger, allowing more “likely” models.

class choice. The method described above, in which we consider a whole array of possible models within a model class, has the advantage over simply calculating the ML estimate of the parameters in that it takes into account the fact that the data are truly limited and thus, we cannot make definite claims as to an underlying model. That is, there are models that are close enough to what has been observed to still be reasonable models for describing the data at hand.

The shortcoming of the KL-based approach is that there is no way to compare between two model classes of differing dimensions. That is, while a two-dimensional model class (i.e., a class having two parameters such as the two marginal firing
Figure 3.4: These plots show model feasibility regions for two model classes that can describe the single-bin response of a two-neuron population. Real data from a two-neuron population in the crayfish visual system were used. Additionally, only 30 stimulus repetitions (or data samples) were available. When presented with this data, the NML-based method chose a model class in which the two neurons are dependent. In the figure on the left, even though the dependence model class was selected by NML, we analyze the data with a 2-parameter independence model class and see that a group of models in this class are still considered to be possible underlying distributions for the data (represented by the presence of the irregular curve). The elliptical contour represents the error between the parameter estimate and the true model, calculated from the quadratic form of the Fisher information matrix. The maximum likelihood estimate of the two marginal firing probabilities of each neuron, $p_1$ and $p_2$ are shown with the x. We see that $\hat{p}_{1ML}$ is .1, and $\hat{p}_{2ML}$ is very small, at .025. On the right, we see the region of likely models for the model class that NML chose — i.e., the dependent model class. Here, the maximum likelihood estimate of the three parameters is $\hat{p}_1 = .1$, $\hat{p}_2 = .025$, and $\hat{p} = .4750$, showing that according to the data, the neurons are very correlated. This is why, even with just 30 samples of the data, NML was able to distinguish the data as being dependent.

probabilities) can be compared to another two-dimensional model class by choosing the class that produces a larger contour for a given probability threshold, a two-dimensional class cannot be compared to a three-dimensional class directly. Additionally, the most complicated model class for a given number of neurons (i.e., the class having the full number of interaction parameters) will always contain the type of the observed data, and thus, the theoretical maximum value of the proba-
bility function will be achieved. So, this method will always favor the model class having the full number of parameters without regard for the actual number of parameters truly needed to describe the data. Thus, we need a way to first determine how many parameters our choice of model class should have, and then analyze the models within that class that could have produced the data with some likelihood.

The Normalized-Maximum Likelihood approach to model selection is well-suited to the task of choosing the model class that best fits the data. Given a set of data from a known number of neurons, we use NML-based model selection to determine the number of parameters truly needed to describe the data. For example, if we have data from a two-neuron population, we choose between the model class in which both neurons are independent and the model class in which both neurons are dependent. In the case of a three-neuron population, we choose between model classes ranging from full independence (in which all three neurons are independent) to partial dependence (in which a neuron may depend only on one of the other two neurons) to a full-dependence model class (in which all three neurons are dependent on each other in some manner). The NML-based approach takes the available data and computes the stochastic complexity for each of the possible model classes for the data and chooses the model class that has the smallest stochastic complexity. This NML-based approach will always choose one of the available model classes as the so-called best choice for the data, but it provides no insight as to whether this choice is actually reasonable. Thus, we look to the KL-based approach to see if the
model class chosen fits the data well enough.

The KL-based approach allows one to see if any model in a given parametric class is likely enough to have produced the data. We see this in Figure 3.2, in which the independent model class was rejected when enough data became available. We would like to use the KL-based approach in a similar manner in conjunction with NML-based model selection. Namely, once the NML-based approach has chosen a model class, we then use the KL-based approach to reject or accept this model class. The ratio described in (3.3) is compared to a threshold in order to accept or reject models in the model class, amounting to a decision whether the KL distance between the model and the type is sufficiently small.

It is not obvious, however, how to choose an acceptable threshold. We will now discuss one approach we have taken in choosing such a threshold, illustrated in Figures 3.5, 3.6, and 3.7. Given a set of data, we use NML to choose a model class. We calculate the KL distance between the data’s type and the maximum likelihood distribution for this model class, i.e., the best model within the class. These are not necessarily the same, as noted above, since the model class might be for independent models and the data (and the type) might reflect correlations. We would like a way to reject or accept the chosen model class based on the KL distance from the type to the ML distribution in the model class. One way to do this is to compare this KL distance to the distribution of KL distances measured between artificial data generated according to the ML distribution and the ML distribution itself. That is,
if the data we have is generally as close to the ML model (of the chosen model class) as the artificial data (that were generated according to the ML model), then it is reasonable to say that the model class is a good one for the data. If, however, the data are much farther away from the ML model than data truly generated by the ML model, then the data do not likely belong to the chosen model class, implying an error on the part of the model class choice.

Thus, our approach allows an exact specification of the range of models that can describe the data reasonably well rather than relying on confidence intervals for individual parameters or a Gaussian-type conception of mean-squared error. This approach consists of selecting a parametric model class using NML-based model selection and then accepting or rejecting this choice. We may then use the KL-based approach to find parametric probabilistic descriptions that are consistent with the data to a specified level. The KL-based approach amounts to finding a Kullback-Leibler distance between the measurements and all possible models, then focusing on those models for which the distance is sufficiently small (equation 3.3). The best model fitting the data presumably has the smallest minimum value for the Kullback-Leibler distance. We do not need to find the variance of our parameter estimates; the parameter feasibility surface does that for us.
Figure 3.5: We generated 10 samples of data for three neurons according to an exponential model class in which only neurons 1 and 2 interact, and all other two- and three-way interactions between the neurons are zero. NML-based model selection incorrectly chose a model class in which all order interactions occur between the 3 neurons (a higher-dimensional, or more complex, model class than the true model class). In the figure, the column of data points on the left shows the distribution of KL distances between the ML estimate in the model class NML chose and datasets generated according to this ML model. The center column shows the 99% confidence interval for the data of the left column, and the red x in this column represents the KL distance between the type of the true data and the ML estimate in the chosen model class. In this case, we see that the distance between the true data and the ML estimate of the chosen model class is within the confidence interval, implying that NML’s choice of model class was a reasonable one. In the rightmost column, we also plot the KL distances between the type and the ML estimates in the other model classes that NML did not choose. With such a limited amount of data as 10 samples, all of the model classes are deemed reasonable because a rejection of any of these classes would warrant more data. However, it is clear that several of the model classes are much worse than others.
Figure 3.6: We generated 40 samples of data for three neurons according to an exponential model class in which only neurons 1 and 2 interact, and all other two- and three-way interactions between the neurons are zero. NML-based model selection incorrectly chose a model class in which interactions also occur between neurons 2 and 3 (but no 3-way interaction occur still), a higher-dimensional, or more complex, model class than the true model class. In the figure, the column of data points on the left shows the distribution of KL distances between the ML estimate in the model class NML chose and datasets generated according to this ML model. The center column shows the 99% confidence interval for the data of the left column, and the red x in this column represents the KL distance between the type of the true data and the ML estimate in the chosen model class. In this case, we see that the distance between the true data and the ML estimate of the chosen model class is within the confidence interval, implying that NML's choice of model class was a reasonable one. In the rightmost column, we also plot the KL distances between the type and the ML estimates in the other model classes that NML did not choose. Even with a limited amount of data like 40 samples, several of the model classes are well outside of the confidence interval, making it clear that only certain model classes (namely, those in which there is interaction between neurons 1 and 2) are "reasonable" and others are not.
Figure 3.7: We generated 300 samples of data for three neurons according to an exponential model class in which only neurons 1 and 2 interact, and all other two- and three-way interactions between the neurons are zero. NML-based model selection correctly chose the model class in which only neurons 1 and 2 interact. In the figure, the column of data points on the left shows the distribution of KL distances between the ML estimate in the model class NML chose and datasets generated according to this ML model. The center column shows the 99% confidence interval for the data of the left column, and the red x in this column represents the KL distance between the type of the true data and the ML estimate in the chosen model class. In this case, we see that the distance between the true data and the ML estimate in the chosen model class is within the confidence interval, implying that NML’s choice of model class was a reasonable one. In the rightmost column, we also plot the KL distances between the type and the ML estimates of the other model classes that NML did not choose. We see that with more and more data, the confidence interval shrinks, and more model classes are easily rejected. In this case, all model classes lacking interaction between neurons 1 and 2 are easily rejected (cluster at the top of the figure), and model classes in which this interaction exists are still well within the confidence interval.
Conclusion

Understanding the dependence structure of a neural population is not an easy task when the amount of data at hand is limited. Non-parametric estimation is a nearly impossible task given the amount of data that would be required, so we turn to parametric model for data and conduct statistical analysis with their aid. In Chapter 1, I presented the Sarmanov-Lancaster and exponential multivariate models for neural responses. The Sarmanov-Lancaster model is an orthogonal and intuitive expansion: correlations between two neurons are easily understood and expressed in this model. However, the Sarmanov-Lancaster model is not easily used for maximum likelihood parameter estimation for so-called embedded models (or reduced-order models) in which we believe certain interactions between neurons do exist while others do not. The exponential model is more amenable to maximum likelihood estimation in these cases due to the Iterative Proportional Fitting Procedure. However, the exponential model’s parameters do not neatly separate interactions into orthogonal components, and as a result, the exponential model’s parameters are less intuitive than the Sarmanov-Lancaster model.

In Chapter 2, I discussed minimum description-length (or MDL) techniques and how they might apply to model selection in the realm of neural population data. If we use the exponential model detailed in Chapter 1, then we would like to select the
exponential model of the smallest order (i.e., having the least number of parameters) that still describes the data well. The Normalized Maximum Likelihood method for model selection accomplishes this goal by balancing a model's fit to the data with the model's complexity. While NML-based methods are easy to understand, calculating various components of NML are difficult. Since the NML distribution itself is not easily calculable, we turn to an asymptotic approximation. Even though we are not operating in the limit of large data samples, we have shown that the terms of the approximation that are difficult to calculate (the Fisher information term and constant term) do not greatly affect our ability to select an appropriate model class. We have also seen that the more complicated a model is, that is, the more parameters a model has, the more data that is required to accurately select this model. This also implies that if real neural data has many interactions present, it will require many more data samples than we have to accurately capture this dependence structure. Finally, if the correlations or dependence present in a data sample are very small or subtle, it will not be possible to accurately detect these interactions with less than thousands of data samples, which we simply do not yet have access to.

In Chapter 3, I discussed our method for finding all models within a parametric model class that are likely to have produced that observed data. We do this by calculating the Kullback-Leibler distance from the observed type to each model in the class, and those models that are close enough to the type are considered likely. I
discussed how we might then formulate a full model-selection method by combining this KL-based approach with that of the NML-based approach. First, analyze data with NML to select a model class (or model order). Then, accept or reject this choice by examining the KL distance between the type and the ML model in the selected model class. If the model class is accepted, then examine various models within the class by considering their KL distance to the type.

Although I have presented a model-selection method, a lack of data still plagues our analysis: if the data contain many dependencies between neurons, our model-selection method may deem all model classes to be reasonable (displaying a tendency not to reject models), and thus choose the simplest model due to the parsimonious nature of NML. However, since we have limited data, it is better to have a range of likely models within a simple model class to begin analysis, rather than to put full faith into an ML estimate which does not convey any information about the verity of the model class that was assumed. Our analysis centered around pairs or groups of three neurons, and even with these small population sizes, our ability to accurately detect interactions of all orders between various neurons was severely limited with data sample sizes that are realistic in neuroscience settings. With the addition of every single neuron to a population, we require double the amount of data, so it is not obvious that even with four neurons we will ever have enough data for proper analysis. With increasing data sample sizes, however, the method presented does fare better in its ability to accurately represent population dynamics.
Bibliography


