RICE UNIVERSITY

Analyzing Statistical Dependencies in Neural Populations

by

Ilan N. Goodman

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
Professor of Electrical and Computer Engineering

Rudolf H. Riedi
Assistant Professor of Statistics

Houston, Texas

January, 2005
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

Analyzing Statistical Dependencies in Neural Populations

by

Ilan N. Goodman

Neurobiologists recently developed tools to record from large populations of neurons, and early results suggest that neurons interact to encode information jointly. However, traditional statistical analysis techniques are inadequate to elucidate these interactions. This thesis develops two multivariate statistical dependence measures that, unlike traditional measures, encompass all high-order and non-linear interactions. These measures decompose the contributions of distinct subpopulations to the total dependence. Applying the dependence analysis to recordings from the crayfish visual system, I show that neural populations exhibit complex dependencies that vary with the stimulus. Using Fisher information to analyze the effectiveness of population codes, I show that optimal rate coding requires negatively dependent responses. Since positive dependence through overlapping stimulus attributes is an inherent characteristic of many neural systems, such neurons can only achieve the optimal code by cooperating.
Acknowledgments

I would like to express my appreciation to the following people: Dr. Richard Baraniuk and Dr. Rolf Riedi, for serving on my thesis committee - their feedback improved this thesis immensely, and gave me some interesting ideas for extending my research; Dr. Raymon Glantz, for providing the data for my analysis; my family and friends, who distracted me when I needed distracting, and whose questions caused me constantly to put my research into perspective; the members of my research group, whose questions and suggestions enabled me properly to develop my ideas; and finally, my advisor and mentor Dr. Don H. Johnson, who is continually opening my eyes to new concepts, and whose guidance has been invaluable.
Contents

Abstract ii

Acknowledgments iii

List of Illustrations vi

1 Introduction 1

1.1 Neurons, Spikes and the Neural Code ......................... 3

1.2 Multivariate Statistical Dependence .......................... 9

2 Multivariate Models and Dependence Measures 12

2.1 Multivariate Models ........................................... 14

2.1.1 Sarmanov-Lancaster Model .............................. 14

2.1.2 Log-linear Model ........................................... 17

2.2 Dependence Measures and Decompositions .................. 18

2.2.1 Phi-Squared Dependence .................................. 20

2.2.2 Kullback-Leibler Dependence ............................ 22

2.2.3 Comparing Dependence Measures ........................ 30

3 Neural Data Analysis 33

3.1 Estimating Probability Distributions .......................... 34

3.2 Bootstrap De-Biasing and Confidence Intervals............... 37
3.3 Data Analysis: The Crayfish Visual System .............................. 40
  3.3.1 Sine Wave Impulse Displacement Data Set ......................... 41
  3.3.2 Sine Wave Motion Data Set ........................................... 43
  3.3.3 Triangle Wave Motion Data Set ...................................... 45

4 Optimal Population Codes ................................................. 49
  4.1 Fisher Information ......................................................... 50
  4.2 Rate Codes ................................................................. 52
  4.3 Dependence Codes ........................................................ 53

5 Conclusion ........................................................................ 58

Bibliography ........................................................................ 63

A Derivation of the Sarmanov-Lancaster Model ....................... 72
Illustrations

1.1 Anatomy of a neuron ........................................ 4
1.2 Hodgkin-Huxley circuit model of cell membrane potential .... 6
1.3 Idealized Hodgkin-Huxley action potentials .................... 6
2.1 Dependence measures as a function of the Sarmanov-Lancaster parameters 22
2.2 Dependence measure decompositions for three Bernoulli random variables . 23
3.1 Estimated vs. exact dependence measures ...................... 38
3.2 Variability of dependence measure estimates ................... 39
3.3 Sine wave and triangle wave grating stimuli .................... 42
3.4 Stimulus timing for impulse displacement and sine wave grating data sets . 43
3.5 Impulse displacement data set .................................. 44
3.6 Sine wave grating data set ..................................... 46
3.7 Triangle wave motion data set ................................... 47
4.1 System model of stimulus encoding in a neural population ........ 50
4.2 Cramér-Rao bound for rate parameters ........................ 54
4.3 Cramér-Rao bound for correlation parameters .................. 56
Chapter 1

Introduction

The fundamental aim of computational neuroscience is to understand how the brain represents information about the external world. Researchers have determined much about how individual neurons behave. Experimental methods for recording neural signals, beginning with Lord Adrian’s first experiments in the 1920s, have become increasingly accurate, with advances in technology for preparing specimens and recording cell impulses, and with improved methods for low-noise amplification and signal denoising [1]. The quantitative model proposed by Hodgkin and Huxley in 1952 [2] provided a simple and accurate mathematical description of the action potential dynamics in a single neuron. Their findings illustrate a universal property of neural systems, that neurons produce sequences of identical spikes that differ only in when the spikes occur. Yet with everything that we know about the neural response, the answer to the central question in computational neuroscience remains largely a mystery: How does the nervous system represent sensory information in these sequences of identical electrical impulses?

Until recently, researchers were unable to record from more than a single neuron at a time; as a result, most research focussed on the ability of a single neuron to encode information in its discharge pattern. Now, multi-electrode recording techniques enable neuroscientists to record simultaneously from many neurons, and the resulting data suggest that
neurons do not process their inputs independently; rather, sensory information is encoded in the coordinated action of many neurons [3–5]. Researchers are now developing techniques to analyze multi-neuron recordings in the hopes of understanding the *population code*, the mechanism by which groups of neurons collectively represent information about the sensory environment.

The first step toward understanding the population code is characterizing statistical dependencies between neural discharge patterns. To that end, most studies use traditional techniques such as cross-correlation to test for statistical dependence between pairs of neurons [3, 6, 7]. However, correlation analysis is inappropriate for populations of more than two neurons, which may include more complicated interactions than simple pairwise linear dependence. In Chapter 2 I develop a new multivariate statistical model based on the work of Sarmanov and Lancaster [8, 9] with an intuitive dependence structure that applies quite well to the analysis of neural populations. I also develop two new statistical dependence measures that have a direct connection to the parameters of the Sarmanov-Lancaster model. Using this model, together with the more traditional log-linear model, I decompose the two dependence measures so that each component represents the contribution of distinct subsets to the total dependence. The resulting techniques thus provide not only a measure of the statistical dependence in a population, but also a detailed picture of the different interactions between neurons that contributed to that dependence. In Chapter 3 I apply the dependence measures to multi-neuron recordings from the crayfish visual system.

Beyond analyzing multi-neuron recordings to determine how neurons interact, we re-
quire the complementary theoretical work to understand why they interact. A common assumption in neuroscience is that through evolution the brain has found optimal or nearly optimal solutions to the problems of perception [10]. Significant evidence supports this assumption, with experiments showing that a variety of different neural systems approach the theoretical limit of performance for their specific functions [11]. Applying this principle to population codes suggests that determining why neurons interact is tantamount to an optimization problem. Understanding the conditions under which a population achieves optimal performance will give insight to explain the observed behavior. Taking this approach, in Chapter 4 I use Fisher information and the Cramér-Rao bound on mean-square estimation error to characterize the optimal population code, and I compare the theoretical results with the observations from the crayfish recordings.

In trying to understand how neural signals represent information, computational neuroscientists exploit concepts from a variety of fields, including neurophysiology, biophysics, and statistics. Before describing my approach to this challenging problem, I must give some attention to the relevant concepts in these fields, and introduce the theoretical tools that I use in the subsequent chapters. I begin with an overview of the neuron and neural signals.

1.1 Neurons, Spikes and the Neural Code

Neurons are specialized cells that encode and process information about the world, ultimately translating sensory information into a behavioral response. Consequently, neurons
Figure 1.1 : Anatomy of a typical neuron. Inputs from other cells are synapsed on dendrites and integrated in the soma. Inputs exceeding a threshold generate action potentials, which propagate along the axon to the presynaptic terminals where they are transmitted to neighboring neurons. *Adapted from [13].*

must have a method to encode, process, and transmit huge amounts of information quickly and efficiently. To accomplish this formidable task, these cells have developed a unique form of communication: neurons encode sensory information in sequences of identical electrical impulses that are propagated efficiently over large distances. Figure 1.1 illustrates a typical neuron. The neuron receives inputs from other neurons (as many as $10^3$) at its *dendrites*, which are integrated in the cell body or *soma*. When sufficient inputs cause the membrane voltage to exceed a threshold, the neuron generates an electrical spike or *action potential*; these signals actively propagate along the *axon* to the *presynaptic terminals*, which form the connections with other neurons[12].

The Hodgkin-Huxley (H-H) theory [2] provides a simple electric circuit model for the
flow of ion currents through the nerve fibre membrane. Through the H-H model, using circuit theory, we can write the corresponding cell dynamics as a system of ordinary differential equations. Since its introduction in 1952, most neuron models have been based on the H-H model [14].

Figure 1.2 shows a circuit diagram of the H-H model. According to this model, the membrane current is the sum of separate currents carried by sodium, potassium, and other ions that flow across the nerve membrane, as well as a capacitive current due to the membrane capacitance. Figure 1.3(a) shows a typical action potential produced by this model. The resting potential difference across the cell membrane is roughly $-70$ mV [14]. When a sufficiently large current (stimulus) is applied, the membrane voltage will exceed a certain threshold, producing a spike that lasts roughly 1-5 ms. The amplitude of the spike is independent of the strength of the stimulus, being rather a property of the neuron that produced it [12].

This last observation gives insight into how neurons encode information: since the size and shape of the spikes are independent of the strength of the stimulus, the time-dependent output of a neuron is essentially a sequence of identical events. Consequently, the only interesting feature of the neural response is when the events occur. Complicating any analysis is the fact that the neural response is not deterministic: repeating a stimulus under identical conditions does not result in identical responses. Rather, the neural response is a stochastic sequence of stereotyped events, usually modeled as a point process [15]. Figure 1.3(b) illustrates a typical response with its corresponding points.
Figure 1.2: Hodgkin-Huxley circuit model of cell membrane potential. The voltage $V$ is the potential difference between the inside and the outside of the neuron. The membrane current is a sum of the current due to membrane capacitance ($I_C$) and currents carried by sodium ions ($I_{Na^+}$), potassium ions ($I_{K^+}$), and a leakage current carried by chloride and other ions ($I_L$). The sodium and potassium conductances are voltage-dependent, modeled as variable resistors. The leakage conductance is fixed. The membrane capacitance is given by $C_m$, and the batteries represent the equilibrium potentials of each ion, i.e., the voltages for which the net membrane current carried by each ion would be zero. Redrawn from [2].

Figure 1.3: Idealized Hodgkin-Huxley (H-H) action potentials. Panel (a) shows a single spike, generated by solving the H-H equations using Euler’s method, with model parameters as in [2]. Panel (b) shows an idealized spike train, with the event times denoted below the plot by an “x”.
Viewing the neural response as a point process, there are traditionally two main categories of coding schemes through which a neural response can represent information: *rate codes*, where information is encoded in the time-varying spike frequency; and *timing codes*, where the information is encoded in the precise timing of the action potential events [4]. Experiments have shown instances where each of these schemes are used. For example, Henry, Dreher, and Bishop [16] found that the average firing rate of certain neurons in the cat visual cortex varies as a function of the orientation angle of a light bar in the visual field; in contrast, in studies of the cat auditory system, Cariani and Delgutte [17] showed that complex acoustic stimuli are encoded in the inter-spike interval statistics of a spike train, a form of timing code. In essence, these two coding schemes are the same, differing only in the temporal resolution at which they are viewed. Thus, viewing the neural response on a very coarse time scale yields only the average rate effects, while higher temporal resolution admits more precise timing information into the analysis.

For a single neuron, the definition of a spike code is problematic [14]. The usual way to estimate the spike timing from data is through repetition of the stimulus and binning: first, present the same stimulus \( R \) times and record the neuron’s response for \( T \) seconds in each trial; second, divide the responses into \( M \) bins of length \( \Delta t \), i.e. \( M = T/\Delta t \). Finally, in each bin, count the number of spikes \( n \) that occurred across all \( R \) trials. Then, the average firing rate in a given bin is \( n/R\Delta t \). As the number of stimulus repetitions increases and the bin width decreases, this quantity approaches the instantaneous firing rate of the neuron. While this method is simple to compute, from the point of view of the
organism it is not possible; the average spike rate must be estimated quickly and reliably from a single instance of the response. The alternative is to count the number of spikes that occur in a certain time interval, and then divide by the length of the interval. However, obtaining a reliable estimate here would require a lengthy observation interval, making the delay between stimulus and response impractically long. This observation lead to the idea of population codes [1]: encoding a stimulus requires the coordinated action of many neurons. If so, the rate, timing, and any other parameters of interest could be estimated reliably over short time scales by observing the entire population response and applying the appropriate filter (e.g. averaging).

The simplest and most popular model of population coding is the additive noise model [18–21]. In this model, the spike rate of the \( i \)\textsuperscript{th} neuron is

\[
r_i = f_i(s) + n_i.
\]  

(1.1)

Here, \( f_i(s) \) is the average rate response for the stimulus \( s \), and \( n_i \) is a random noise term, usually assumed to be Gaussian. The function \( f_i(s) \) is sometimes called a tuning curve, and it characterizes exactly how the neuron's response is related to some feature of the stimulus [14]. Various decoding schemes have been proposed under different assumptions on the form of the tuning curve and the distribution of the noise; examples include maximum likelihood methods [20], maximum a posteriori methods [14], and population vector methods [22].

An important question in population coding is whether the neurons are statistically
dependent; for example, the noise term in equation 1.1 could be drawn from a correlated Gaussian distribution. In that case, the inter-neuron correlations may have a significant effect on both the form and accuracy of the optimal decoder. There have been two main approaches to determining how dependencies affect decoding accuracy. The first approach uses information theoretic quantities such as mutual information to measure synergy and anti-synergy, whether the population represents information better or worse than the sum of the individual neurons’ contributions [5]. For example, Panzeri et al. [23] and Pola et al. [24] use mutual information to measure the extent to which the population encodes the stimulus. Then, by breaking the mutual information into separate rate and dependence effects, they determine how dependencies contribute to encoding the stimulus. The second approach uses Fisher information and the Cramér-Rao bound to determine the optimal code in terms of the minimum mean-square estimation error [19, 25, 26]; I take this approach in Chapter 4. This method assumes a parametric model for the population response and asks under what conditions is the optimal decoding error minimized.

1.2 Multivariate Statistical Dependence

To discuss statistical dependence between neural responses, we need to arrive at a concrete formulation of what we mean by dependence. Probability theory gives a concise technical definition of statistical independence: the random variables \( X = (X_1, \ldots, X_N) \) are statistically independent if and only if \( p_X(x) = \prod_{i=1}^{N} p_{X_i}(x_i) \), where \( p_X(x) \) is the joint probability distribution and \( p_{X_i}(x_i) \) is the marginal distribution of the \( i^{th} \) random vari-
able [27]. For Gaussian random variables, the correlation coefficient uniquely determines independence; if the correlations between all pairs of variables are zero, then the random variables are statistically independent. In general, however, zero correlation does not imply independence. The correlation coefficient is a measure of linear relationship between two random variables only; for most multivariate models, non-linear relationships may exist, and furthermore, the variables may not be restricted to pairwise interactions. Rather, most multivariate statistical models reflect dependence in a much more complicated way.

Joe [28] discusses four properties that are desirable for any multivariate statistical model. First, the model should be easy to interpret, and model parameters should each relate to a specific type of statistic; for example, there may be dependence parameters, as well as parameters that relate strictly to the univariate marginal probabilities. Second, all marginal distributions should also be members of the same parametric family; this is called the closure property. Third, the model should be flexible enough to capture a wide range of dependence. Finally, there should be a closed-form expression for the model’s probability function. Joe demands that there be a closed-form expression for the model’s cumulative density function; however, I relax that requirement since there is no known model that satisfies all four conditions otherwise. Assuming a less stringent version of the last condition, the multivariate Gaussian model satisfies all the conditions, and several other multivariate statistical models in the literature satisfy at least three of the conditions. Examples include the multivariate Poisson distribution [29] and multivariate exponential distributions, which satisfy all but the last condition, and certain families of copulas, which are multivariate
distributions with marginals that are uniform on the interval \([0,1]\)\(^*\) [28].

One method of analyzing statistical dependencies from data is simply to fit the data to a multivariate model. However, we often desire a method to measure dependence without requiring an explicit model. Appropriate dependence measures must capture the main characteristics of the dependence structure. For example, the sample correlation coefficient may not reflect the true dependence if the samples were not generated from a bivariate Gaussian distribution, and in that case would not be an appropriate measure. Non-parametric variants of the correlation coefficient such as the \textit{Spearman rank correlation} and \textit{Kendall’s tau} measure dependence between random variables corresponding to the cumulative density functions. However, such methods do not generalize easily to more than pairs of random variables, and only apply to continuous distributions [30]. Some measures such as the extremal dependence measure [31] focus on \textit{tail dependence}, the tendency of certain distributions to generate simultaneously large values for all the variables. Other traditional dependence measures include Pearson’s chi-square statistic, and variations such as Yule’s measure of association [32]. The following chapter develops some multivariate models and dependence measures that are appropriate for analyzing non-Gaussian data, apply to both continuous and discrete distributions, and generalize to groups of more than two random variables.

\(^*\)For a rigorous treatment of copulas, see [28, 30].
Chapter 2
Multivariate Models and Dependence Measures

The theory of statistical dependence for Gaussian random vectors is simple and well established; dependence is characterized completely by the product-moment correlations between each pair of random variables [30]. In contrast, correlation is not a good description of dependence for non-Gaussian random variables. For example, consider $N$ Bernoulli distributed random variables. Fully specifying their joint distribution requires $2^N - 1$ parameters; there are $N(N - 1)/2$ pairwise correlations, which, together with the $N$ marginal probabilities, can specify the joint distribution only when $N = 2$. For larger numbers of variables, third and higher order dependencies also need to be determined. In general, when dealing with non-Gaussian random variables, many different dependence concepts could be used. The challenge is to find an unambiguous, meaningful description of dependencies flexible enough to apply to a useful class of problems. This chapter introduces two non-parametric measures of dependence having this flexibility that apply to a large class of multivariate distributions, and that are detailed enough to give a thorough understanding of the interactions between random variables that cause the dependencies.

The dependence measures presented here result from a very basic definition of dependence. The components of a random vector $\mathbf{X} = (X_1, \ldots, X_N)$ are said to be statistically dependent if and only if $p_{\mathbf{X}}(\mathbf{x}) \neq \prod_{i=1}^{N} p_{X_i}(x_i)$, where $p_{\mathbf{X}}(\mathbf{x})$ is the joint probability
distribution and $p_{X_i}(x_i)$ is the marginal distribution of the $i^{th}$ random variable [27]. Complete dependence occurs when the random variables are single-valued functions of each other; hence, a natural measure of dependence should be related to the divergence between the joint distribution and the product distribution. The techniques developed in this chapter quantify this divergence using a class of information-theoretic distances defined by Ali and Silvey [33]. However, quantifying the dependence in this manner does not provide the detailed understanding of the underlying interactions causing the dependence. To obtain that level of detail, the dependence measures must relate to a model’s parameters so that different types of interactions can be identified and their contribution to the overall dependence determined. To that end, this chapter develops two multivariate models that are highly flexible and whose parameters relate directly to the different types of interactions between variables that contribute to statistical dependence.

Since all practical applications of the techniques developed here will involve either a fully discrete system or a discrete approximation to a continuous system, I restrict the discussion in this chapter to the case of discrete random variables only. Consequently, the probability functions will by convention be assumed to be probability mass functions, i.e. $p_X(x) = \Pr[X = x]$. Unless otherwise stated, the equations presented here can all be applied to the continuous random variable case simply by replacing the mass functions with continuous densities.
2.1 Multivariate Models

Multivariate models for most collections of non-Gaussian random variables are extremely complicated, and, in most cases, are mathematically intractable for more than two or three random variables [28]. To fully understand multivariate dependence we need a statistical model for the multivariate probability function that incorporates an explicit dependence structure. I begin by developing the Sarmanov-Lancaster model, which is extended from an expansion of bivariate distributions that was introduced by Sarmanov [8, 34] and later elaborated by Lancaster [9, 35–37].

2.1.1 Sarmanov-Lancaster Model

Sarmanov [8] and Lancaster [35] independently came up with a series expansion for a family of bivariate distributions. Here I extend the expansion for the general case of $N$-vari ate distributions. Let $X = (X_1, \ldots, X_N)$ be a discrete random vector with joint probability distribution $p_X(x)$ and marginal distributions $p_{X_i}(x_i)$, and define the quantity $\phi^2 = \sum_x p_X^2(x) / \prod_{i=1}^N p_{X_i}(x_i) - 1$. This quantity is extensively explored in section 2.2.1.

Provided that $\phi^2 < \infty$, the joint probability function can be expanded as

$$p_X(x) = \left[ \prod_{i=1}^N p_{X_i}(x_i) \right] \left[ 1 + \sum_{i_1, \ldots, i_N} a_{i_1, \ldots, i_N} \prod_{n=1}^N \psi^{(n)}_{i_n}(x_n) \right].$$ (2.1)

The functions $\{\psi^{(n)}_{i_n}\}$ form a complete set of orthonormal functions with respect to the marginal distribution $p_{X_n}(x_n)$, and $\psi^{(n)}_0 = 1$, $n = 1, \ldots, N$. This expansion is derived in detail in Appendix A. The choice of orthonormal functions is not unique; for example,
Bahadur [38] uses orthogonal polynomials in his expansion of a multivariate binary distribution, and Lancaster [9] uses the normalized Legendre polynomials to expand several continuous bivariate distributions.

The choice of unity as the zero-order function lends a particularly interesting property to the expansion. For simplicity, let \( i = i_1 \cdots i_N \), and define \( \psi_i(x) = \prod_{n=1}^{N} \psi_{i_n}^{(n)}(x_n) \). The indices \( i \) can be partitioned into disjoint sets \( \beta_k, \ k = 1, \ldots, 2^N - N - 1 \) such that the functions \( \psi_j, \ j \in \beta_k \) are functions of the \( k \)th unique subset of the \( N \) random variables only. For example, if \( \beta_1 \) represents the subset \((X_1, X_2)\), then for all \( j \in \beta_1 \), \( \psi_j(x) = \psi_j(x_1, x_2) \). We can then write equation 2.1 as

\[
p_x(x) = \left[ \prod_{i=1}^{N} p_{X_i}(x_i) \right] \left[ 1 + \sum_{k=1}^{2^N - N - 1} \sum_{i \in \beta_k} a_i \psi_i \right]. \tag{2.2}
\]

Now the coefficients \( a_i, \ i \in \beta_k \) represent the interactions within the \( k \)th subset, separate from any lower order interactions and interactions with other variables not in the \( k \)th subset.

For example, consider three jointly defined binary random variables \( X_1, X_2 \) and \( X_3 \), with \( p_n = \Pr[X_n = 1] \) and \( \sigma_n^2 = p_n(1 - p_n), \ n = 1, 2, 3 \). Using orthogonal polynomials as a basis set for the marginal probability distributions, we have for each \( X_n \)

\[
\psi_0(x_n) = 1 \quad \psi_1(x_n) = (x_n - p_n)/\sigma_n.
\]

Now the coefficients \( a_{i_1 i_2 i_3} \) in the expansion of the joint distribution function are easy to
compute:

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

Here, \( \rho_{ij} = (\mathbb{E}[x_i x_j] - p_i p_j)/(\sigma_i \sigma_j) \) are the usual product-moment correlation coefficients and \( \rho_{ijk} = (\mathbb{E}[x_i x_j x_k] - p_i p_j p_k)/(\sigma_i \sigma_j \sigma_k) \) is a third-order correlation-like quantity.

In light of Joe's [28] desiderata (Section 1.2), the Sarmanov-Lancaster model represents an extremely powerful model: it consists of strictly univariate parameters and dependence parameters, and furthermore each dependence parameter refers to a very specific mode of interaction between variables. The resulting dependence structure captures the entire range of dependencies that is achievable for a given set of marginals; it possesses the closure property for all univariate, bivariate, and higher order marginals; and there is a closed-form expression for its probability function. The main drawback to this model is the large number of parameters; the number of parameters increases exponentially with the number of variables. For that reason, fitting this model to high dimensional data is unrealistic. Instead, in Section 2.2.1 I develop a dependence measure to quantify indirectly the contribution of the Sarmanov-Lancaster parameters. First, however, I will describe a second multivariate model with properties similar to, though less ideal than the Sarmanov-Lancaster model.
2.1.2 Log-linear Model

The log-linear or exponential model of multivariate data is frequently used in the analysis of multivariate categorical data [39–42]. The dependence structure defined by this model is somewhat less useful than the Sarmanov-Lancaster dependence structure, since the parameters here do not correspond to interactions of unique subsets on the joint distribution. However, I will use this model in section 2.2.2 to elaborate the dependence measure \( \nu \).

As before, let \( \mathbf{X} = (X_1, \ldots, X_N) \) be a discrete random vector with joint probability distribution \( p_{\mathbf{X}}(\mathbf{x}) \), and let \( |\mathbf{X}| \) be the cardinality of the sample space of \( \mathbf{X} \). With minimal loss of generality, assume that \( p_{\mathbf{X}}(\mathbf{x}) > 0 \) for all \( \mathbf{x} \). Then we can write the log-likelihood function as a linear sum:

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

The functions \( g_i \) are model specific and are not required to be orthogonal. The quantity \( \psi(\theta) \) is chosen to make the probability function sum to unity. For example, consider the three-variable binary distribution with \( \Pr[X_n = 1] = p_n \) and \( \Pr[X_1 = x_1, X_2 = x_2, X_3 = x_3] > 0, \forall x_1, x_2, x_3 \). Then,

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

Here the \( \theta_i \) express the effect of \( p_i \) on the joint probability, \( \theta_{ij} \) are measures of correlation between pairs of random variables, and \( \theta_{123} \) expresses third-order dependence.

As in the Sarmanov-Lancaster model, the parameters of the log-linear model can be partitioned into subsets representing interactions of specific subsets of random variables,
as they are in equation 2.4. However, since the parameters of the log-linear model are not projections onto orthogonal bases, the parameters do not represent strictly independent modes of interaction [28]. For example, having $\theta_{12} = 0$ in equation 2.4 does not imply zero dependence between the variables $X_1$ and $X_2$, unless $\theta_{123} = 0$. In general, the marginal probability functions of a log-linear model depend on all the parameters of the joint probability function in a complicated way, and hence the log-linear parameters can not be interpreted strictly in terms of different modes of interaction. Consequently, the log-linear model does not have the closure property, and so the marginal distributions may not have the same parametric form as the joint distribution.

While the Sarmanov-Lancaster model, and, to a certain extent, the log-linear model provides a powerful description of multivariate statistical dependencies, fitting data to the model is no easy task, requiring a huge amount of data when the number of variables is more than two or three. Rather than applying the models directly, in the following section I develop two dependence measures that are related to the model parameters but do not require explicit parameter estimation.

### 2.2 Dependence Measures and Decompositions

A convenient way to quantify statistical dependence is to measure the dissimilarity between the joint probability function and the independent product distribution. For example, the *mutual information*, which is a popular measure of dependence between two random variables, is really the Kullback-Leibler (KL) divergence between the joint dis-
tribution and the independent product distribution. The KL divergence is a member of a general class of coefficients of divergence known as Ali-Silvey distances[33]. Let $p^{(1)}(x)$ and $p^{(2)}(x)$ be two distributions for the random vector $X$, and define the likelihood ratio
\[ \Lambda(x) = \frac{p^{(1)}(x)}{p^{(2)}(x)}. \]
Ali-Silvey distances have the form
\[ d(p^{(1)}, p^{(2)}) = f(\mathbb{E}[c(\Lambda(x))]), \] (2.5)

where $c(\cdot)$ is a continuous convex function, $f(\cdot)$ is an increasing real function, and the expectation is with respect to $p^{(2)}$.

Taking $p^{(1)}$ to be a joint distribution function and $p^{(2)}$ to be the corresponding independent product distribution in equation 2.5, we obtain a general class of dependence measures that includes the mutual information. This definition illustrates a fundamental characteristic of statistical dependence: as dependence between random variables increases, the joint distribution becomes concentrated on an increasingly smaller set in the joint sample space relative to the independent product distribution [43]. Ali and Silvey [44] proved that in the case of a bivariate Gaussian distribution, all dependence measures in the Ali-Silvey class are non-decreasing functions of each of the correlation coefficients. Difficulties prevent obtaining similar results for more complicated non-Gaussian models; differences in how dependence relates to various multivariate parameters makes a generalization of that result impossible. However, in the following section I will develop an Ali-Silvey dependence measure called the phi-squared dependence, and prove that the measure is an increasing function of the Sarmanov-Lancaster dependence coefficients.
2.2.1 Phi-Squared Dependence

Pearson defined the mean-square contingency $\phi^2$ as a generalization of his $\chi^2$ test statistic, which he used to test for significant departures of a variable from its expectation. I give $\phi^2$ a different interpretation: I define it as an Ali-Silvey measure of dependence, so that it might inherit the properties of that class. Let $\mathbf{X} = (X_1, \ldots, X_N)$ be a random vector with joint probability distribution $p_{\mathbf{X}}(\mathbf{x})$ and marginal distributions $p_{X_i}(x_i)$, and define the likelihood ratio $\Lambda = p_{\mathbf{X}}(\mathbf{x})/ \prod_{i=1}^{N} p_{X_i}(x_i)$. Then,

$$\phi^2 = \mathbb{E}[(\Lambda - 1)^2] = \sum_{\mathbf{x}} \frac{p_{\mathbf{X}}^2(\mathbf{x})}{\prod_{i=1}^{N} p_{X_i}(x_i)} - 1. \tag{2.6}$$

For a given set of random variables, $\phi^2 = 0$ if and only if the variables are statistically independent, and $\phi^2$ increases with increasing dependence. When $\mathbf{X}$ is a continuous random vector with corresponding continuous probability density functions, the same definition applies by making the sum in equation 2.6 an integral.

When the variables in $\mathbf{X}$ are all discrete, $\phi^2$ can be transformed to the interval $[0, 1]$ using a simple bound known as the Frechét upper bound [28]. Using Bayes’ Theorem [27], for any $i$ we can write the joint distribution as $p_{\mathbf{X}}(\mathbf{x}) = p_{X_i}(x_i)p(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N \mid x_i)$. Since for discrete distributions the conditional probabilities are all less than or equal to 1, the joint distribution is upper
bounded by \( \min_i p_{X_i}(x_i) \). Hence,
\[
\phi^2 \leq \sum_{x:p_X(x) > 0} \frac{\min_n p_{X_n}(x_n)}{\prod_{i=1}^N p_{X_i}(x_i)} - 1 \equiv \phi_{\max}^2. \tag{2.7}
\]
The normalized measure is then \( \overline{\phi^2} = \phi^2 / \phi_{\max}^2 \). Complete dependence occurs when the random variables are one-to-one functions of each other. In that case, the marginal distributions are equal, and \( p_X(x) = p_{X_i}(x_i) \) at exactly \( N \) points and \( p_X(x) = 0 \) everywhere else; hence \( \overline{\phi^2} = 1 \) if and only if the variables in \( X \) are completely dependent.

Since \( \phi^2 \) is an Ali-Silvey measure, when \( X \) is a Gaussian random vector \( \phi^2 \) is a non-decreasing function of the correlation coefficients. For example, if \( (X_1, X_2) \) are jointly Gaussian random variables with correlation coefficient \( \rho \), the dependence is readily computed to be \( \phi^2 = \rho^2 / (1 - \rho^2) \). A more general result is obtained by using the Sarmanov-Lancaster model. Substituting equation 2.1 into equation 2.6 we obtain the simple relation
\[
\phi^2 = \sum_i a_i^2. \tag{2.8}
\]
Thus, \( \phi^2 \) is an increasing function of each of the dependence parameters in the Sarmanov-Lancaster model. Figure 2.1(a) illustrates this result for the simple three-variable Bernoulli case.

Equation 2.8 gives rise to an additional interesting property of \( \phi^2 \): The dependence measure can be decomposed into separate components, each representing the contribution of a different subset of variables independently to the total dependence. Using the partitioning of the coefficients defined in equation 2.2, we can rewrite \( \phi^2 \) as the sum \( \phi^2 = \sum_{k=1}^{N^2-N-1} \phi_k^2 \)
where \( \phi_k^2 = \sum_{i \in \beta_k} a_i \), and \( \beta_k \) defined as in section 2.1.1. It is convenient for the discussion
Figure 2.1: (a) The normalized phi-squared dependence and (b) the normalized KL dependence measure is shown as a function of the Sarmanov-Lancaster parameters for a set of three Bernoulli random variables, \( (X_1, X_2, X_3) \). The set is homogeneous, with \( P[X_1 = 1] = P[X_2 = 2] = P[X_3 = 3] = .35 \) and \( \rho_{12} = \rho_{13} = \rho_{23} \). This illustrates that \( \phi^2 \) and \( \nu \) are increasing functions of each of the Sarmanov-Lancaster dependence parameters. Note that the cut-off regions in the corner correspond to invalid sets of model parameters.

To re-index the components to indicate directly which subset they represent:

\[
\phi^2 = \sum_{i<j} \phi^2_{i,j} + \sum_{i<j<k} \phi^2_{i,j,k} + \cdots + \phi^2_{1,2,3,\ldots,N} .
\] (2.9)

For example, \( \phi^2_{12} \) is the component of \( \phi^2 \) due entirely to pairwise interactions between the variables \( X_1 \) and \( X_2 \).

Continuing the three-variable Bernoulli example of section 2.1.1, the decomposition is

\[ \phi^2 = \rho^2_{12} + \rho^2_{13} + \rho^2_{23} + \rho^2_{123} . \]

This decomposition is illustrated in Figure 2.2.

2.2.2 Kullback-Leibler Dependence

The best known dependence measure in the Ali-Silvey class is the mutual information for two random variables. Here I generalize it for multiple random variables. Let
Figure 2.2: Decomposition of dependence measures for three jointly distributed Bernoulli random variables. The joint distribution $p(x_1, x_2, x_3)$ was simulated in each bin. The top plot shows the marginal distributions $P[X_n = 1]$ for each variable; the marginal distributions are constant throughout the simulation. The middle plot shows the components of the normalized $\phi^2$ dependence measure, computed as described in section 2.2.1. The bottom plot shows the components of the normalized dependence measure $\nu$, computed as described in section 2.2.2. Between bin 0 and bin 50 there is a constant level of 2\textsuperscript{nd} order dependence between $X_1$ and $X_2$. Between bin 25 and bin 75 there is also 2\textsuperscript{nd} order dependence between $X_1$ and $X_3$. Between bins 50 and 100 there is 3\textsuperscript{rd} order dependence between the random variables.
\( \mathbf{X} = (X_1, \ldots, X_N) \) be a random vector with joint probability distribution \( p_{\mathbf{X}}(\mathbf{x}) \) and marginal distributions \( p_{X_i}(x_i) \), and define the likelihood ratio \( \Lambda = p_{\mathbf{X}}(\mathbf{x}) / \prod_{i=1}^{N} p_{X_i}(x_i) \). The KL dependence \( \nu \) is the KL divergence between the joint probability function and the product distribution,

\[
\nu = \mathbb{E}[\Lambda \log \Lambda] \\
= \sum_{\mathbf{x}} p_{\mathbf{X}}(\mathbf{x}) \log \frac{p_{\mathbf{X}}(\mathbf{x})}{\prod_{i=1}^{N} p_{X_i}(x_i)}. \tag{2.10}
\]

For a given set of random variables, \( \nu = 0 \) if and only if the variables are independent, and \( \nu \) increases with increasing dependence. This measure is also defined for continuous densities, in which case the sum in equation 2.10 is an integral.

Like \( \phi^2 \), \( \nu \) can be normalized when the variables in \( \mathbf{X} \) are all discrete. In that case, using the log-sum inequality [45] we find

\[
\nu = \sum_{j \neq i} H(X_j) - H(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_N | X_i) \\
\leq \min_i \sum_{j \neq i} H(X_j) \\
\equiv \nu_{\text{max}} \tag{2.11}
\]

where \( H(X_j) = -\sum_{x_j} p_{X_j}(x_j) \log p_{X_j}(x_j) \) is the entropy of the random variable \( X_j \).

We could alternatively derive this bound using the Frechét upper bound as in the previous section. When the random variables are completely dependent, the conditional entropies are all zero and so the inequality becomes an equality; thus the normalized quantity \( \bar{\nu} = \nu / \nu_{\text{max}} = 1 \) if and only if the random variables are strictly dependent.
In the case of two jointly Gaussian random variables, \( \nu \) is an increasing function of the correlation coefficient, \( \nu = -\frac{1}{2} \log (1 - \rho^2) \). While a formal proof has not been forthcoming, simulations have shown that for non-Gaussian multivariate distributions, \( \nu \) is an increasing function of the Sarmanov-Lancaster parameters. Figure 2.1(b) illustrates this relationship for the three-variable Bernoulli case.

Amari [40] describes a decomposition of \( \nu \) such that each component is due to interactions of a different order. For example, interactions of order \( k \) are mutual interactions between \( k \) random variables. The decomposition is constructed through information geometry, so it is necessary to introduce a few basic concepts before developing it. Consider a family of parametric joint probability distributions \( \mathcal{M} = \{p(x; \xi)\} \), where \( \xi = (\xi_1, \ldots, \xi_m) \) is a vector of real-valued parameters that fully specify the distribution of the random vector \( X \). In information geometry, this family of distributions forms an \( m \)-dimensional manifold with \( \xi \) as its coordinate system. For example, the joint distribution of \( N \) binary random variables is specified by \( 2^N - 1 \) parameters and thus lies on an \( m = 2^N - 1 \) dimensional manifold.

To begin, consider a parametric curve on the manifold \( \xi(t), t \in \mathbb{R} \). When the curve is a linear function of \( t \), it is called a geodesic. The geodesic minimizes the arc length over all paths between two points in a manifold. Amari [46] proved the following Pythagoras theorem: Given a manifold \( \mathcal{M} \) and three points \( p_1, p_2, p_3 \in \mathcal{M} \), when the geodesic connecting \( p_1 \) and \( p_2 \) is orthogonal to the geodesic connecting \( p_2 \) and \( p_3 \), then \( d^2(p_1, p_3) = d^2(p_1, p_2) + d^2(p_2, p_3) \). Here, \( d^2(\cdot, \cdot) \) is the squared distance along the
geodesic. When the points on the manifold are probability distributions, the squared distance is equal to twice the Kullback-Leibler distance between the corresponding distributions [40]; thus,

$$D(p_1 || p_3) = D(p_1 || p_2) + D(p_2 || p_3).$$ (2.12)

Next, consider two dual geometric properties of certain manifolds. A manifold is called e-flat if there is a parametrization $\theta$ such that

$$\mathbb{E} \left[ \frac{\partial^2}{\partial \theta_i \partial \theta_j} \log p_X(x; \theta) \frac{\partial}{\partial \theta_k} \log p_X(x; \theta) \right] = 0$$ (2.13)

for all $i, j, k$. Any $\theta$ coordinate curve is a geodesic (which is why the manifold is called flat), and the connection between two points on the manifold along these coordinates is called an e-geodesic. An example of such a parametrization is the log-linear model, written here as an exponential:

$$p_X(x; \theta) = \exp \left( \sum_i \theta_i g_i(x) - \psi(\theta) \right).$$ (2.14)

A manifold is called m-flat if there is a parametrization $\eta$ such that

$$\mathbb{E} \left[ \frac{1}{p_X(x; \eta)} \frac{\partial^2}{\partial \eta_i \partial \eta_j} p_X(x; \eta) \frac{\partial}{\partial \eta_k} \log p_X(x; \eta) \right] = 0$$ (2.15)

for all $i, j, k$. Again, all $\eta$ coordinate curves are geodesics, and two points on the manifold are connected along the $\eta$ coordinates by an m-geodesic. An example of an m-flat manifold is the mixture family $p_X(x; \eta) = \sum_i \eta_i q_i(x)$, where $\{q_i\}$ are probability distributions, $0 < \eta_i < 1$, and $\sum \eta_i = 1$. While it is not true in general that log-linear distributions can be expressed as mixture distributions, it is always true that manifolds that are e-flat are also
$m$-flat (and vice-versa). In other words, if an $e$-flat parametrization exists on a given manifold, there also exists an $m$-flat parametrization. Consequently, such manifolds are called dually flat.

We proceed by partitioning the parameters such that each partition contains only parameters that describe the same dependence order. Let $X = (X_1, \ldots, X_N)$ be a random vector with $x_n \in \mathcal{X}$ and joint probability distribution $p_X(x)$. If $p$ lies on an $e$-flat manifold $\mathcal{E} = \{p(x; \theta)\}$ with coordinates $\theta$, the manifold is also $m$-flat and can be written as $\mathcal{M} = \{p(x; \eta)\}$ with coordinates $\eta$. We rewrite the coordinates in terms of the ordered partitions $\theta = (\theta_1, \ldots, \theta_N)$ and $\eta = (\eta_1, \ldots, \eta_N)$, where $\theta_k$ and $\eta_k$ are the set of all parameters that describe the interactions of order $k$; in other words, these parameters are functions of exactly $k$ variables. Using this partition, define the submanifolds

$$\mathcal{E}_{k+} = \{p(x; \theta) : \theta_1 = 0, \ldots, \theta_k = 0\}$$

$$\mathcal{M}_{k-} = \{p(x; \eta) : \eta_{k+1} = 0, \ldots, \eta_N = 0\}.$$  \hfill (2.16)

Thus the submanifold $\mathcal{E}_{k+}$ contains only probability distributions with non-zero parameters of order higher than $k$, whereas $\mathcal{M}_{k-}$ contains distributions with non-zero parameters of order lower than or equal to $k$. It can be shown that the two submanifolds are complementary and orthogonal at every point. Hence, we can define a new mixed coordinate system that orthogonally separates interactions of different orders; this new coordinate system is written $(\eta_{k-}, \theta_{k+})$. Note that in the original coordinate systems, the coordinates are not necessarily orthogonal; for example, in the three variable log-linear model, $\theta_{123} = 0$ does
not imply zero third-order dependence. However, in the mixed coordinate system, since the $\theta_3$ coordinates are orthogonal to the $\eta_{3-}$ coordinates, $\theta_3 = 0$ does in fact imply zero third-order dependence.

Letting $p(x)$ be a distribution on a dually flat manifold with coordinates $(\theta_1, \ldots, \theta_N)$ and $(\eta_1, \ldots, \eta_N)$, we define the distribution $p^{(k)}(x)$ to be the point on the manifold closest to $p(x)$ that contains no dependencies of order $k$ or higher. Using the mixed coordinate system, this point corresponds to the coordinates $(\eta_{k-}, \theta_{k+})$, with $\theta_{k+} = 0$. Similarly, define a third distribution $p^{(j)}(x)$ with $1 \leq j \leq k$. The points $p$ and $p^{(k)}$ lie on the same $e$-flat manifold, and consequently are connected by an $e$-geodesic. Similarly, $p^{(k)}$ and $p^{(j)}$ lie on the same $m$-flat manifold and are connected by an $m$-geodesic. Because of the orthogonality of the two coordinate systems, the $e$-geodesic connecting $p$ and $p^{(k)}$ is orthogonal (at $p^{(k)}$) to the $m$-geodesic connecting $p^{(k)}$ and $p^{(j)}$, and consequently the Pythagoras theorem of equation 2.12 holds. Iterating this process, we obtain the decomposition

$$\nu = D(p || p^{(1)}) = \sum_{k=2}^{N} D(p^{(k)} || p^{(k-1)}) = \sum_{k=2}^{N} \nu_k,$$

(2.17)

Thus $\nu$ is decomposed into $N - 1$ components, where each component is the contribution of interactions of a different order to the overall dependence between the random variables.

Consider once more the three-variable binary example. We obtain the parametrization of equation 2.4, which has the coordinates $\theta = (\theta_1, \theta_2, \theta_3)$, where $\theta_1 = (\theta_1, \theta_2, \theta_3)$, $\theta_2 = (\theta_{12}, \theta_{13}, \theta_{23})$, and $\theta_3 = (\theta_{123})$. Similarly, define an $\eta$ parametrization with the coordinates $\eta = (\eta_1, \eta_2, \eta_3)$, where $\eta_1 = (\eta_1, \eta_2, \eta_3)$, $\eta_2 = (\eta_{12}, \eta_{13}, \eta_{23})$, and $\eta_3 = (\eta_{123})$. 
Here, $\eta_i = \mathbb{E}[x_i]$, $\eta_{ij} = \mathbb{E}[x_i x_j]$, and $\eta_{123} = \mathbb{E}[x_1 x_2 x_3]$. We need to obtain the coordinates of $p^{(2)}$, which in the mixed coordinate system correspond to the coordinates $(\theta_1, \theta_2, \eta_3)$ with $\eta_3 = 0$. We thus find the distribution $p^{(2)}$ by setting $\theta_3 = 0$ and solving for the new coordinate $\eta_3$.

To find the transformation between $\theta$ and $\eta$, first let $p_{x_1 x_2 x_3} = P[X_1 = x_1, X_2 = x_2, X_3 = x_3]$. Rewriting $\theta$ in terms of these probabilities, we find that

$$
\theta_{123} = \log \frac{p_{111} p_{100} p_{010} p_{001}}{p_{000} p_{110} p_{101} p_{011}}. \quad (2.18)
$$

Since $\eta$ consists of expectations, we can also write $\eta$ in terms of the probabilities

$$
\eta = A p, \quad (2.19)
$$

where $A$ is a $(2^N - 1) \times (2^N - 1)$ invertible matrix and $p = (p_{x_1 x_2 x_3})_{x \in \mathcal{X}, x \neq (0,0,0)}$ is the vector of probabilities excluding $p_{000}$. Hence, to find $p^{(2)}$, set equation 2.18 equal to zero and solve for the new $\tilde{\eta}_{123}$. Then simply substitute $\tilde{\eta}_{123}$ back into equation 2.19 and solve for $p$. Finally, we note that $p^{(1)}$ is simply the independent distribution, the product of the marginal probabilities. Thus we obtain the components of $\nu = \nu_2 + \nu_3$,

$$
\nu_2 = D(p^{(2)} || p^{(1)})
$$

$$
\nu_3 = D(p || p^{(2)}).
$$

This decomposition is illustrated in figure 2.2.

In this section we developed two dependence measures with similar properties. While both measures are in the Ali-Silvey class, each one concentrates on a different aspect of
statistical dependence. Much like different distortion measures highlight different kinds of error, $\phi^2$ tends to emphasize big differences between the joint and product distributions and deemphasize small ones, while $\nu$ emphasizes small departures. In choosing an appropriate dependence measure to use in data analysis, we require an objective way of comparing the two quantities.

2.2.3 Comparing Dependence Measures

Rényi [47] discussed seven properties that an appropriate measure of dependence should have. Here I generalize them for multivariate dependence measures. Letting $\delta(X)$ be any dependence measure for the random vector $X$, the following conditions should be satisfied:

1. $\delta(X)$ is defined for any $X$ whose members are non-deterministic.

2. $\delta(X)$ does not depend on the ordering of the variables in $X$; eg.

$$\delta(X_1, X_2, X_3) = \delta(X_2, X_1, X_3).$$

3. $\delta(X) \in [0, 1].$

4. $\delta(X) = 0$ if and only if the random variables are independent.

5. $\delta(X) = 1$ if and only if there is strict dependence between the random variables, ie.

$$X_1 = f_2(X_2) = f_3(X_3) \cdots \text{ where } \{f_i\} \text{ are real single-valued functions.}$$

6. If $\{f_i(\cdot)\}$ are real single-valued functions, then $\delta(f_i(X_i)_{i=1}^N) = \delta(X)$.

7. If $X$ is a bivariate Gaussian, then $\delta(X) = |\rho|.
Both dependence measures $\phi^2$ and $\nu$ satisfy postulates 1 and 2 and 6 trivially. Conditions 3, 4, and 5 are satisfied for the normalized measures discussed in the discrete case; when the random variables are continuous, however, there is no meaningful upper bound for either dependence measure. Rényi’s last postulate concerning the correlation coefficient bears little importance in the multivariate case. Joe [48] proposed the transformations $\nu^* = [1 - \exp(-2\nu)]^{1/2}$ and $\phi^{2*} = [\phi^2/(1 + \phi^2)]^{1/2}$, so that the transformed measures satisfy postulates 3 and 7 for the bivariate case; however, the transformations do not satisfy postulate 5 for non-Gaussian random variables, and consequently are not of much use.

The chief difference between the two dependence measures lies in their decompositions, which provide a more detailed view of the dependencies in a set of variables. While $\nu$ is an attractive measure because of its relation to information theory [4], the $\nu$ decomposition is somewhat less detailed than the $\phi^2$ decomposition since it does not distinguish between different subsets of variables within each order of interaction. Moreover, extending the three variable example to four variables, we find that determining the third-order effects in the decomposition requires solving a system of four non-linear equations; thus, in general, computing the decomposition for $\nu$ requires solving large sets of nonlinear equations, which is computationally expensive compared to the quadratic function evaluations required to compute the $\phi^2$ decomposition.

Using the simple bound $\log x \leq x - 1$, we see that $\phi^2$ is an upper bound on $\nu$; however, this bound is tight for small dependencies only. The two dependence measures are also related to each other through the Taylor series expansion of $\nu$. Using the definition of $\nu$
in equation 2.10 and taking the Taylor series expansion about the product distribution, we
obtain
\[ \nu = \frac{1}{2} \phi^2 + \cdots. \] (2.20)
Thus \( \frac{1}{2} \phi^2 \) is the first term in the Taylor series expansion for \( \nu \). Note that since the expansion
is about the product distribution, \( \frac{1}{2} \phi^2 \) is only a good approximation for \( \nu \) when the depen-
dence is small. Additionally, computational examples have shown that this relationship is
preserved under the decomposition; for example, if \( \phi_2^2 = \sum_{i<j} a_{ij} \) is the total second-order
component of \( \phi^2 \) and \( \nu_2 \) is the second-order component of \( \nu \), then \( \nu_2 \approx \frac{1}{2} \phi_2^2 \). This result
is somewhat surprising, given the very different methods of obtaining the decompositions.
However, it is also reassuring, as it indicates that the decompositions do in fact correspond
to the same types of interactions in each case.

This chapter took a fairly broad approach to the problem of quantifying multivariate sta-
tistical dependence. Two multivariate models were discussed that specify a very large class
of multivariate distributions, and two non-parametric dependence measures were de-
veloped along with decompositions into components representing interactions within unique
subsets of variables. Although the concepts discussed here could be used in a variety of
applications (cf. [49]), we chose the models and dependence measures with a specific ap-
plication in mind: the analysis of neural population data. In the next chapter I show exactly
how the concepts introduced here are applied to this analysis, with examples using real data
recorded from the crayfish visual system. For the analysis I use the \( \phi^2 \) measure, both for
the ease with which it is computed and the greater detail the decomposition provides.
Chapter 3

Neural Data Analysis

Given simultaneous spike train recordings from several neurons comprising a population, we would like to compute a dependence measure and its decomposition. To analyze a neural response, the spike trains must be converted into a rather simple symbolic form: Divide the response time into discrete bins of length $\Delta$, and count the number of spikes that occur in each bin. For a population response, this procedure must be repeated for each of the $N$ neurons in the population. The response can then be represented by the sequence $\mathbf{R} = \{R_t\}_{t=1}^T$, where $R_t$ is a $D$-ary code representing the spike count in the $t^\text{th}$ bin for every neuron, and $T\Delta$ is the total length of the response. Here, $D$ is the maximum number of spikes produced by any neuron in any bin. For example, in a two-neuron population with a maximum of one spike per bin, if neuron 1 fired and neuron 2 did not, the symbolic representation would be $R_1 = 10_2 = 2$. The $R_t$ thus take on values from the set $\mathcal{A} = \{0, \ldots, D^N - 1\}$, called an alphabet [4]. Data analysis then proceeds by analyzing the statistics of the sequence $\mathbf{R}$.

A limitation often experienced by researchers in neuroscience is the lack of sufficient data to produce any meaningful analysis. This problem is compounded in the case of population analysis, where instead of a single neuron’s response, we would like to analyze many neurons’ simultaneous responses. In that case, the amount of data needed for the
analysis grows exponentially in the population size; thus for a population of \( N \) neurons, more than \( D^N \) samples per bin would be required [4]. The techniques presented in this chapter overcome this problem in part by using improved methods for estimating probabilities and bootstrap re-sampling methods for reducing bias and estimating confidence intervals.

### 3.1 Estimating Probability Distributions

The simplest estimate of the probability distribution of a discrete random variable is the empirical distribution, known as the type. This estimate is the maximum likelihood estimate of the probability of each symbol [45]. Given a sequence \( R = \{R_n\}_{n=1}^N \) of independent, identically distributed samples of a discrete random variable \( X \) that takes values \( x \in \mathcal{A} \), the type \( \hat{p}_X(x) \) is the relative frequency of each symbol in the sequence,

\[
\hat{p}_X(x) = \frac{1}{N} \sum_{n=1}^N I(R_n = x).
\]  

(3.1)

Here, \( I(\cdot) \) is an indicator function which equals 1 if its argument is true, and 0 otherwise.

The Law of Large Numbers assures us that the type converges to the true distribution as the number of samples \( N \) increases [27]; however, some problems emerge when \( N \) is small or when the total number of symbols \( A = |\mathcal{A}| \) is large relative to \( N \) [50]. To illustrate why, consider the problem of estimating the distribution of English letters in written text from a small sample, e.g. the word “sample.” The type estimate would assign the probability \( 1/6 \) to each of the letters \{s, a, m, p, l, e\} and 0 to the remaining 20 letters in the alphabet.
Thus the maximum-likelihood estimate of the probability of observing a new letter in the next sample is zero. However, in this case the type estimate seems excessively naïve, since we would certainly expect more than 6 letters to appear in a large sample of text. When the alphabet size is large relative to the sample size, the type tends to overestimate the probabilities of symbols that do appear in the sample, and assign zero probability to those symbols that do not appear.

Several methods have been proposed to improve probability estimates when the amount of data is limited. The simplest of these is the \textit{add-constant} estimators [51],

\[
\hat{p}_X^{(ac)}(x) = \frac{\sum_{n=1}^{N} I(R_n = x) + \beta}{N + \beta A}.
\]  

The add-constant estimators assign a non-zero probability to all symbols that do not appear in the sample. The constant \(\beta\) effectively constrains the shape of the estimated distribution, which is equivalent to placing a prior distribution on the observations [52]. Krichevsky and Trofimov [53] showed that the choice of \(\beta = 1/2\) is asymptotically optimal for certain applications; using this choice results in the Krichevsky-Trofimov (K-T) estimator [4].

Good and Turing [52] suggested an alternative to add-constant estimators that results in improved probability estimates for symbols not appearing in the sample. Letting \(\mu_x = \sum_{n=1}^{N} I(R_n = x)\) be the number of times the symbol \(x\) appears in the sample, define the variable \(q(\mu) = \sum_{x \in A} I(\mu_x = \mu)\). In other words, \(q(\mu)\) is the number of symbols
which appear exactly $\mu$ times in the sample. The Good-Turing estimator is then [52]

$$
\hat{p}_X^{(gt)}(x) = \frac{1}{K} \begin{cases} 
q(1) & \text{if } \mu_x = 0 \\
(\mu_x + 1)q(\mu_x + 1)/q(\mu_x) & \text{if } \mu_x > 0,
\end{cases} 
$$

(3.3)

where the factor $K$ makes the probabilities sum to one. Like the add-constant estimators, this method amounts to placing a prior distribution on the observations. However, here the prior distribution is empirically learned from the observations themselves [54]. Good [52] showed that this estimator converges to the true probabilities, and provides better estimates for the probability of unseen symbols when the sample size is small. This estimator does not perform well, however, for symbols that appear frequently in the sample, and variants of this method have been proposed to improve the probability estimates obtained in that case [50].

Estimating the dependence measures from data essentially amounts to estimating the probability distribution of the joint response. The dependence measures are then computed directly from the estimated distribution using equations 2.6 and 2.10. The quality of the estimate thus depends on the method used to estimate the probability distribution. Figures 3.1 and 3.2 compare the dependence measures computed using four different probability estimators: the unmodified type, the K-T method, the Good-Turing method, and a modified Good-Turing method proposed by Orlitsky, Santhanam and Zhang [50]. Figure 3.1 shows that the K-T method produces the least bias of all the methods when the sample size is small. When the sample size is large, all methods yield approximately the same perfor-
mance, except the G-T method which performs significantly worse. Figure 3.2 shows the variability $\sigma/\mu$, where $\sigma$ is the standard deviation and $\mu$ is the mean of the estimates. For the small sample size, the K-T method produces the lowest variability in the estimates. For the large samples, all methods perform well except for the O-S-Z method, which performs significantly worse for small values of dependence. The K-T method works well for both small and large sample sizes, and consequently K-T is the method used for the population analysis in section 3.3.

The figures also illustrate a further difficulty with using the unmodified types for small samples; when any of the estimated marginal distributions are impulses (i.e. the variable is deterministic), the normalized dependence measures are undefined. The K-T method ensures that all estimated probabilities are greater than zero, and hence the normalized measures can always be computed.

### 3.2 Bootstrap De-Biasing and Confidence Intervals

Although the estimates obtained for $\nu$ and $\phi^2$ using any of the empirical density methods described in section 3.1 are asymptotically unbiased, the small number of samples that can be obtained from neural recordings can lead to significant bias in the estimates [4]. The bootstrap method [55] is a re-sampling technique that can be employed to estimate and remove this bias. Let $R = \{R_1, \ldots, R_N\}$ be the length $N$ dataset from which a population parameter $\theta(R)$ is to be estimated. A bootstrap sample $R^* = \{R^*_1, \ldots, R^*_N\}$ is a random sample of size $N$ drawn with replacement from the original dataset. Thus some elements of
Figure 3.1: Estimated dependence measures for a three variable binary population (alphabet size = 8) averaged over 1000 trials. The panels on the left show the normalized dependence measures $\phi^2$ and $\nu$ estimated from 16 samples; the panels on the right show the dependence measures estimated from 160 samples. The graphs show estimates obtained using four methods: unmodified types, Krichevsky-Trofimov (K-T), Good-Turing (G-T), and Orlitsky-Santhanam-Zhang (O-S-Z). The K-T estimator performs well for small and large sample sizes.
Figure 3.2: Variability ($\sigma/\mu$) of the dependence measure estimates for a three variable binary population (alphabet size = 8). Variability was estimated from 1000 independent trials. Good performance is indicated by low variability. The panels on the left show the variability of the estimates computed from 16 samples; the panels on the right show the variability of estimates from 160 samples. The graphs show estimates obtained using four methods: unmodified types, Krachevsky-Trofimov (K-T), Good-Turing (G-T), and Orlitsky-Santhanam-Zhang (O-S-Z). The K-T estimator performs well for small and large sample sizes.
the original dataset may appear in the bootstrap sample more than once, and some not at all. Taking $B$ bootstrap samples, the bootstrap estimate of bias for $\theta$ is the average bootstrap estimate minus the original estimate:

$$\text{bias} = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}(\mathbf{R}^*_b) - \theta(\mathbf{R}),$$

(3.4)

where $\{\hat{\theta}(\mathbf{R}^*_b), b = 1, \ldots, B\}$ are the estimates of $\theta$ computed from each bootstrap sample. The debiased estimate of $\theta$ is then $2\theta(\mathbf{R}) - \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}(\mathbf{R}^*_b)$. Around 200 bootstrap samples are required to give a good estimate of bias in most cases [55].

We can also use the bootstrap method to compute confidence intervals for the estimates. Let $\hat{F}(a) = \frac{1}{B} \sum_{b=1}^{B} I[\theta(\mathbf{R}^*_b) < a]$ be the cumulative density function of the bootstrap estimates $\hat{\theta}(\mathbf{R}^*_b)$. A simple estimate for the $100\alpha\%$ confidence interval is $[\hat{F}^{-1}((1 - \alpha)/2), \hat{F}^{-1}((1 + \alpha)/2)]$. This corresponds to taking the $(1 - \alpha)/2$ and $(1 + \alpha)/2$ percentiles of the bootstrap replications. For example, the 90% confidence interval would be $[\hat{F}^{-1}(.05), \hat{F}^{-1}(.95)]$. The confidence interval for the debiased estimate is then $[2\theta(\mathbf{R}) - \hat{F}^{-1}((1 - \alpha)/2), 2\theta(\mathbf{R}) - \hat{F}^{-1}((1 + \alpha)/2)]$. While 200 bootstrap samples is sufficient for estimating bias, estimating confidence intervals requires on the order of 2000 samples [56].

3.3 Data Analysis: The Crayfish Visual System

The initial processing stage of the crayfish optic nerve is made up of 14 distinct sustaining fibers (SFs). These neurons are responsible for converting the analog signals received from
photoreceptors in the eye into spike trains [57]. The SFs are also partially responsible for behavior, being functionally connected to the optomotor neurons responsible for eyestalk movement [58].

The data analyzed in this section were recorded in three different experiments*. In these experiments, the crayfish was secured with its eye position fixed. Light stimuli were then projected onto a surface completely surrounding the subject’s visual field; consequently, distances and speeds of the stimulus are measured in terms of degrees on the circular projection surface. Micro-electrodes were used to record from two optomotor neurons and from up to seven sustaining fibers in the crayfish optic nerve. For the analysis I use the K-T method described in section 3.1 to estimate probability distributions from repeated presentations of the stimuli.

### 3.3.1 Sine Wave Impulse Displacement Data Set

This data set consists of two optomotor neurons responding to an impulse displacement of a sine wave grating in the visual field. Figure 3.3(a) illustrates the sine wave grating. Figure 3.4(a) shows the timing of the stimulus presentation: in each trial, a sine wave light grating with an angular period of $60^\circ$ was projected onto the visual field, and after 5 seconds the grating was instantaneously displaced through an angular distance $\Delta \theta \in \{+10, -10, +20, -20, +30, -30\}$, measured in degrees. For example, with

*The author is indebted to Dr. R.M. Glantz, Professor of Biochemistry and Cell Biology, Rice University for providing the data analyzed here. For more experimental details, see [59].
Figure 3.3: Examples of light grating stimuli presented to the crayfish visual field. Panel (a) shows a sine wave grating; light intensity varies sinusoidally around the projection surface. Panel (b) shows a triangle wave grating. Here, the light intensity varies linearly over each half-cycle.

A displacement of $+10^\circ$ the light intensity at a fixed point in the visual field would have changed by $1/3$ of a period of the sine wave.

Figure 3.5 shows the post-stimulus time (PST) histograms or time-dependent average firing rate, along with the estimated dependence measures for this experiment. In each trial the stimulus was presented 20 times. There is a constant low level of dependence throughout the stimulus presentation for every trial, between .005 and .01 on the normalized scale; while small, this quantity is statistically significant as indicated by the bootstrap confidence intervals. The negative displacement angle trials show some interesting effects at the onset of the stimulus: at 5 seconds when the grating is displaced, these trials show a decrease in dependence over one or two bins, but the dependence quickly returns to previous levels as the subject adapts to the new stimulus condition.
Figure 3.4: Timing of stimulus presentation for the sine wave data sets. Panel (a) shows the stimulus timing for the impulse displacement data set; after 5 seconds, the grating is instantaneously displaced through an angular distance of $\Delta \theta$. Panel (b) shows the stimulus timing for the motion data set; after a 2 second pause, the grating is displaced at a constant spatial frequency through an angular distance of $\Delta \theta$, and then is returned to its original orientation.

3.3.2 Sine Wave Motion Data Set

This data set consists of two optomotor neuron responses to a moving sine wave light grating, shown in Figure 3.3(a). Figure 3.4(b) shows the stimulus timing for this experiment. At the beginning of each trial, a grating with angular wavelength of $\lambda \in \{30, 120, 60, 40, 80, 22.5, 11.25, 5.625\}$, measured in degrees, was projected onto the crayfish visual field. After a 2 second pause, the grating was displaced with a constant temporal frequency of 0.1 Hz for 5 seconds; after another 2 second pause, the grating was returned to its original orientation with the same temporal frequency. For example, for a 30° sine wave, the light intensity at a fixed point in the visual field would have gone through 6 cycles of the sine wave during each displacement portion of the trial.

Figure 3.6 shows the results of the sine wave grating experiment. For the 30° grating,
Figure 3.5: Results of the sine wave impulse displacement experiment. Each panel shows the results of a single trial, averaged over 20 repetitions of the stimulus. The displacement angle for each trial is indicated underneath the corresponding panel. The top two plots in each panel are the PST histograms for each of the two optomotor neurons. Note that the PST histograms have been scaled differently to show detail in the low-rate neurons. The bottom plot in each panel shows the normalized $\phi^g$ dependence, computed from the K-T estimate of the population response distribution in each 20 ms bin. The bootstrap method was used to remove bias and compute 90% confidence intervals for the estimates; confidence intervals are enclosed by the dotted lines.
the population response exhibits higher dependence during the stimulus adapting periods, between 0 and 2 seconds and between 7 and 9 seconds. Thus the effect of the moving stimulus was to decrease dependence in the population response. The $60^\circ$ and $11.25^\circ$ trials exhibit similar effects.

### 3.3.3 Triangle Wave Motion Data Set

In this data set, a population of 2 optomotor neurons and 7 sustaining fibers was recorded responding to a triangle wave grating stimulus (figure 3.3(b)) moving with constant spatial frequency $f \in \{0.1, 0.05\}$, measured in Hz. For example, for the 0.1 Hz grating, the light intensity at a fixed point in the visual field would vary like a triangle wave with period of 10 seconds.

Figure 3.7 shows the results of the triangle wave motion experiment for a small population of two optomotor neurons and two sustaining fibers. The higher frequency grating here elicits a stronger rate response than the low frequency grating. However, the overall dependence levels are similar for both frequencies. For both the high and low frequency gratings, the second-order subpopulations exhibit around twice as much dependence ($4 \times 10^{-5}$) as the third-order subpopulations; thus almost $3/4$ of the total dependence is due to pairwise dependencies only. In all cases, dependence levels are roughly constant throughout the stimulus presentation.

The dependence analysis conducted in this chapter served to illustrate two main points. First, using the techniques from the previous chapter, we were able to gain a detailed per-
Figure 3.6: Results of the sine wave grating displacement experiment. Each panel shows the results of a single trial, averaged over 30 repetitions of the stimulus. The angular wavelength of the grating for each trial is indicated underneath the corresponding panel. The top two plots in each panel are the PST histograms of each of the two optomotor neurons. Note that the PST histograms are scaled differently to show detail in the low-rate neurons. The bottom plot in each panel shows the normalized $\phi^2$ dependence, computed from the K-T estimate of the population response distribution in each 20 ms bin. The bootstrap method was used to remove bias and compute 90% confidence intervals for the estimates; confidence intervals are enclosed by the dotted lines.
Figure 3.7: Results of the triangle wave grating experiment. The left panels show the population response to a 0.1 Hz triangle wave light grating. The right panels correspond to a 0.05 Hz grating. The top four plots in each panel show the PST histograms of two optomotor neurons and two sustaining fibres. The fifth plot shows the total normalized $\phi^2$ dependence, computed from the K-T estimate of the population response distribution in each 10 ms bin. The remaining plots show the components of the total $\phi^2$ corresponding to each unique subpopulation, indicated to the left of each graph. Dependence estimates were de-biased and 90% confidence intervals were computed using the bootstrap method. Confidence intervals are enclosed by the dotted lines.
spective on the population dependencies exhibited in a real neural system. Second, while small, the dependencies are statistically significant, and in some cases vary over the course of the stimulus presentation. Taken alone, these results are somewhat startling; given that the neurons examined have overlapping inputs as well as lateral connections, we might have expected a higher level of dependence. In the following chapter I provide a theoretical characterization of the optimal population code that may explain what we observed from the crayfish data.
Chapter 4

Optimal Population Codes

So far this thesis has been primarily concerned with measuring statistical dependence and determining the dependence structure of a neural population from recordings. The results in chapter 3 show that neural populations can exhibit complicated dependencies that may vary with the stimulus. It is natural, then, to wonder how dependencies affect the efficacy of the population code — does dependence enhance or diminish the ability of the neurons to represent the stimulus effectively? Many studies have concluded that simple dependencies can improve the accuracy of the population code [19, 25, 60], while others concluded that dependence limits coding accuracy [6, 26, 61]. These studies either used mutual information to judge under what conditions the information capacity of a population would be enhanced or diminished, or they used Fisher information to determine the accuracy of the optimal decoding method given a specific model for the population response. Using mutual information is problematic and the results can be misleading*, and consequently in this chapter I take the second approach: Given a model for the population response, namely the Sarmanov-Lancaster model of section 2.1.1, I use Fisher information and the Cramér-Rao bound to characterize the optimal population code. I analyze two types of codes here: rate codes, where the stimulus is encoded by the neurons’ marginal spike probabilities; and

*For a detailed explanation, see [61, 62].
Figure 4.1: Model of stimulus encoding and decoding by neural populations. A single stimulus $s$ is encoded as a vector parameter $\theta = f(s)$ of the joint distribution of the spiking responses of the neurons in a population. After transmitting through an unknown channel, the parameter $\hat{\theta}$ is estimated from the received spikes $\hat{X}$, and the stimulus is recovered by applying the decoding function $\hat{s} = f^{-1}(\hat{\theta})$.

dependence codes, where the stimulus is encoded by the interneuron dependencies.

4.1 Fisher Information

Figure 4.1 shows a model for stimulus encoding by a population of neurons. Here, $S$ represents a scalar valued stimulus and $X = \{X_1, \ldots, X_N\}$ is the joint response (spikes) of the population of $N$ neurons. The stimulus is encoded by the joint statistics of the neural spike trains, which are expressed by the response probability law. Given a specific stimulus $s$, the encoding is described parametrically by the probability function

$$p_X(x|s) = p_X(x; \theta(s)), \quad (4.1)$$

where $\theta(s) = \{\theta_1(s), \ldots, \theta_K(s)\}$ is a vector parameter encoding the stimulus. Since the stimulus is encoded in the parameter $\theta$, decoding the population response amounts to estimating $\hat{\theta}$ from a particular response $\hat{X}$. Consequently, the optimal decoding scheme would minimize the estimation error over all decoding schemes.

Fisher information is a useful way of quantifying the accuracy with which a parameter can be estimated from data. Given a probability distribution $p_X(x; \theta)$ with scalar parameter
\( \theta \), the Fisher information is defined as the negative expected value of the second derivative of the log-likelihood function [45]:

\[
J(\theta) = -\mathbb{E} \left[ \frac{\partial^2}{\partial \theta^2} \log p_X(x; \theta) \right].
\]  (4.2)

The expectation here is taken with respect to \( p_X(x; \theta) \). When the components of \( X \) are statistically independent, the Fisher information of the vector is equal to the sum of the individual components, i.e. \( J(\theta) = \sum_{i=1}^{N} J_i(\theta) \). The Fisher information is related to the accuracy of the optimal estimator for the parameter \( \theta \) through the Cramér-Rao inequality, which states that the mean-squared estimation error \( \mathbb{E} \left[ \hat{\theta} - \theta \right]^2 \) for any unbiased estimator is greater than or equal to the reciprocal of the Fisher information [45]:

\[
\mathbb{E} \left[ \hat{\theta} - \theta \right]^2 \geq \frac{1}{J(\theta)}.
\]  (4.3)

Any unbiased estimator that achieves the Cramér-Rao bound (CRB) with equality is called efficient. While the Cramér-Rao inequality gives a bound on the estimation error for all unbiased estimators of \( \theta \), it does not guarantee the existence of an efficient estimator. Nevertheless, the Fisher information is a useful quantity for comparing various encoding schemes, as it provides a direct measure of the accuracy of the optimal decoder.

When the parameter being estimated is a vector, the **Fisher information matrix** is the expected value of the Hessian of the log-likelihood function [45]:

\[
J(\theta) = -\mathbb{E} \left[ \nabla_{\theta} \nabla_{\theta}^T \log p_X(x; \theta) \right].
\]  (4.4)

In that case, the lower bound on the estimation error for the parameter \( \theta_i \) is \([J^{-1}(\theta)]_{ii}\), the \( i^{\text{th}} \) diagonal element of the inverse Fisher information matrix.
Using equation 4.4, we can compute the Fisher information matrix for the parameters of the Sarmanov-Lancaster model, \( \theta = \left( (p_{X_n}(x_n))_{n=1}^{N}, (a_{i_1 \ldots i_N}) \right) \).

\[
J(\theta) = -\mathbb{E} \left[ \nabla_\theta \nabla_\theta^T \sum_{n=1}^{N} \log p_{X_n}(x_n) + \log \left( 1 + \sum_{i_1, \ldots, i_N} a_{i_1 \ldots i_N} \prod_{n=1}^{N} \psi^{(n)}_{i_n}(x_n) \right) \right] \\
= \sum_{n=1}^{N} J_n(\theta) - \mathbb{E} \left[ \nabla_\theta \nabla_\theta^T \log \left( 1 + \sum_{i_1, \ldots, i_N} a_{i_1 \ldots i_N} \prod_{n=1}^{N} \psi^{(n)}_{i_n}(x_n) \right) \right] \\
= J_{\text{ind}}(\theta) + J_{\text{dep}}(\theta). \tag{4.5}
\]

Here, \( J_n \) is the Fisher information matrix of the \( n^{\text{th}} \) neuron, \( J_{\text{ind}} \) is the Fisher information of the “independent” population and \( J_{\text{dep}} \) represents the contribution of the dependence terms.

By analyzing the \( J_{\text{dep}} \) term we can determine how dependencies affect the performance of the optimal decoder.

### 4.2 Rate Codes

The simplest model of the population response is the multivariate Bernoulli model [26].

Here, the individual neurons’ responses are binary; that is, in a given bin each neuron either fires once or does not fire at all. Letting \( p_n = \Pr[X_n = 1] \) be the probability of neuron \( n \) producing a spike, the parameter of interest is \( \theta = (p_1, \ldots, p_N) \). Applying the Sarmanov-Lancaster expansion to the joint distribution and using the orthogonal polynomials as the bases for the expansion (cf. [38]), we then compute the corresponding Fisher information matrix for an \( N \) variable binary population.

Consider the simple homogeneous two-neuron binary population with \( p_1 = p_2 = p \).

Figure 4.2 shows the inverse Fisher information \( J^{-1}(\theta) \), \( \theta = p \) as a function of the de-
pendence parameter $\rho$, for different values of $p$. The figure shows that, while positive correlations increase the minimum decoding error, negative correlation actually improves decoding accuracy. This result suggests that the optimal two-neuron population rate code is one in which the spike responses are as negatively correlated as possible. This interpretation must be considered carefully. If we bound the bivariate Bernoulli distribution using the Fréchet lower bound, we obtain [28]

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

(4.6)

Thus the admissible range of correlation is directly limited by the marginal probabilities of the neurons. If the marginal probability is low, for example $p = .1$, the strongest possible negative correlation is $\rho \approx -.11$, which corresponds to $\phi^2$ on the order of $10^{-2}$. As the marginal spike probability goes to zero, the lower bound on the correlation in equation 4.6 also goes to zero; thus for very low values of $p$, the optimal population rate code has approximately uncorrelated components.

### 4.3 Dependence Codes

The parameters of the Sarmanov-Lancaster model include both rate parameters (marginal probabilities) and dependence parameters. The previous section considered pure rate coding, in which the population may be independent but the dependencies are not related to the stimulus. A second alternative is dependence coding; in this scheme, the stimulus is encoded in the dependence parameters of the joint response only, while the rate parameters
Figure 4.2: Cramér-Rao bound on mean-square estimation error for the marginal rate parameter of a bivariate binary distribution. The CRB is shown as a function of the correlation $\rho$ for various values of the rate parameter $p$. Negative correlations decrease the minimum variance for estimating $p$. 
remain unrelated to the stimulus. DeCharms and Merzenich [63], Maynard et al. [64] and others have reported evidence for such dependence codes in real neural systems. In this case, the Fisher information consists of only the $J_{dep}$ term, since the marginal distributions are unrelated to the dependence parameters. Evaluating the resulting Fisher information matrix using the Sarmanov-Lancaster model with $\theta = (a_i)$, we obtain

$$J_{ij}(\theta) = [J_{dep}(\theta)]_{ij} = \sum_x \frac{\prod_{n=1}^N p_{X_n}(x_n)}{p_X(x)} \psi_i(x) \psi_j(x).$$

(4.7)

This expression depends on the marginal spike probabilities of the neurons through the basis functions $\{\psi_i(x)\}$.

Consider again the homogeneous two-neuron binary population. The parameter of interest for the dependence code is $\theta = (\rho)$, where $\rho$ is the correlation coefficient for the Sarmanov-Lancaster expansion using polynomial bases. Figure 4.3 shows the inverse Fisher information for the correlation coefficient as a function of the marginal spike probability $p_1 = p_2 = p$. With positive correlation, the mean-square error bound for the dependence parameter is lowest for values of $p$ near 0.5 and increases dramatically when the marginal spike probability is low. For negative correlation, however, best performance is achieved for low marginal probabilities, and the CRB is similar to the positive correlation case when the spike probabilities are high. Thus the optimal correlation code encodes the stimulus in negative values of correlation with the smallest possible marginal probabilities (as dictated by the Fréchet bound).

It is difficult to determine from neural recordings, such as the data presented in Chap-
Figure 4.3: Cramér-Rao bound on mean-square estimation error for the correlation parameter of a bivariate binary distribution. The CRB is shown as a function of the marginal rate $p$ for various values of the correlation $\rho$. Best performance is obtained for high positive and negative values of $\rho$, with the sign depending on the marginal probabilities $p$. 
ter 3, whether stimulus information is encoded in the marginal spike probabilities, in interneuron dependencies, or a combination of the two. However, the analysis presented here suggests that for either scheme, the optimal population code should exhibit negative interneuron dependencies, and the results of the data analysis in Chapter 3 are consistent with this finding. The small dependencies that we found in the crayfish data are on the same order as the negative dependencies that are predicted by the Fisher information analysis and the Fréchet bound. The data analysis, however, does not distinguish between negative and positive dependence, and consequently more sophisticated techniques may be required to determine conclusively whether the data exhibit optimal population coding.
Chapter 5

Conclusion

Neuroscientists have long recognized that, while single neurons can make incremental contributions to encoding and processing sensory information, most significant processing in the nervous system is due to the coordinated actions of populations of neurons. As a result, there are numerous papers dealing with theoretical issues related to population codes. Most of these studies are united by two common themes.

Accuracy: The stochastic nature of the neural response limits the accuracy to which the stimulus parameters can be recovered from a single neuron’s response. Under the right conditions, considering the outputs of multiple neurons responding to the same stimulus increases the estimation accuracy, enabling a more appropriate behavioral response.

Robustness: Involving large numbers of neurons in a single processing task may make the system less sensitive to failures of individual neurons.

To simplify the analysis, such studies often assume that the neurons in the population are statistically independent. However, this assumption can have significant effects on both the accuracy and robustness of the population, and consequently must be borne out by experimental evidence. Even before considering any empirical results, this assumption seems rather odd, since we expect that neurons reacting to the same or related stimulus param-
eters should exhibit some stimulus-induced dependence in their outputs. Until recently, recording from populations of neurons was impossible, so this assumption was never validated. Now, multi-electrode recording techniques promise to shed light on the question of inter-neuron dependencies, and the techniques presented in this thesis provide the statistical tools to analyze dependencies in such population data.

Quantifying statistical dependencies among jointly distributed random variables is no simple task. The correlation coefficient, which only measures linear dependence between pairs of variables, is often misapplied simply because of the lack of any other comparatively simple technique. The $\phi^2$ and $\nu$ dependence measures described in Chapter 2 solve this problem; both quantities are easy to apply to data, and encompass the entire range of dependencies between any number of random variables. Both measures decompose orthogonally into contributions from distinct subsets, providing a greater level of detail about the interactions in the population. Although both measures provide a strong expression of dependence, practical considerations dictate which measure should be used for analyzing neural populations. Decomposing $\nu$ requires solving large systems of non-linear equations, and hence is too computationally expensive to be of practical use. Moreover, the decomposition only distinguishes between different orders of dependence, without indicating which specific variables interact. Decomposing $\phi^2$ is also computationally intensive, requiring $2^N$ function evaluations for a population of $N$ neurons; however, this can be accomplished much more efficiently (for example, with multiple processors). The $\phi^2$ decomposition is also much more detailed, elucidating the specific interactions in the population that con-
tribute to the overall dependence.

The crayfish data analyzed in Chapter 3 exhibited very small but statistically significant dependencies. While this finding does seem to violate the independent response assumption, it is unclear how these small dependencies affect the theoretical results. In Chapter 4 I used the Sarmanov-Lancaster model, whose parameters are directly related to \( \phi^2 \), to determine the optimal conditions for population coding. There, I found that optimal encoding of the stimulus by spike firing rates requires negatively dependent outputs. In addition, I showed that the range of the dependence parameters is directly limited by the rates, with lower rates admitting a smaller range for each dependence parameter. It is perhaps not surprising, then, that the observed dependencies are so low. Indeed, in all three experiments analyzed, the observed dependencies are around what is predicted by this result.

The analysis in Chapter 4 relates the optimal code to the problem of estimating the parameters of a statistical model. While this does provide some insight into the observations in Chapter 3, it is by no means the whole picture. We require further analysis to understand how a stimulus parameter gets mapped onto a population response. Taking the stimulus into account changes the expression for Fisher information given in equation 4.2. Letting \( \alpha \) represent the stimulus parameter being encoded by the parameter vector \( \theta \), the Fisher information becomes

\[
F(\alpha) = \frac{d\theta^T}{d\alpha} F(\theta(\alpha)) \frac{d\theta}{d\alpha}.
\]

Johnson and Ray [26] evaluate this expression numerically for the parameters of the multivariate Bernoulli model. In their analysis, they seek the optimal coding function that
provides constant Fisher information over the entire range of stimulus values. As a result, they find that only positive correlations can improve performance of the optimal code, but that the performance gain is small; they conclude that the optimal population code (for their coding criterion) has effectively independent outputs. The small dependencies in the crayfish neurons could also support this view of population coding. In addition, Johnson and Ray show that stimulus-induced dependence can only result in positively correlated outputs. Ultimately, their findings and the results presented in this thesis are equally startling.

In both cases, we found that dependencies, if they exist, are very small; this result is surprising, since neurons responding to the same stimulus should certainly exhibit stimulus-induced dependence. Johnson and Ray offer an intriguing solution: neurons in a population could cooperate to decorrelate the stimulus-induced dependence, thereby approaching the optimal code. This hypothesis has its own difficulties, however, including consequences on the size of the population required to implement the decorrelating function, as well as the robustness of the resulting population. However these results are interpreted, we clearly require more sophisticated analyses to fully understand population codes.

One important issue for population codes that this thesis did not deal with at all is the problem of temporal dependencies. For example, the output of one neuron could be dependent on the output of a second neuron 100 ms in the past, or even on its own past output. Such effects could easily be approximated, for example, by an appropriate Markov model; whatever approach is used, the consequence of considering temporal dependencies is simply to add more parameters to the multivariate model. In that case, the amount of data
required to estimate the dependence parameters grows as $D^{BN}$, where $D$ is the alphabet size, $B$ is the Markov order, and $N$ is the number of neurons. For any reasonably sized population, the amount of data thus required becomes prohibitively high; indeed, the four-neuron population analyzed in section 3.3.3 is about the largest population we can reliably analyze with the amount of data available, without even considering temporal dependence. While improved probability estimators and bootstrap methods compensate somewhat for the lack of data, it is still difficult (if not impossible) to obtain reliable estimates for larger populations. To deal with this problem, Johnson and Uppuluri [65] take a different approach: rather than estimating a single parameter value, they search for a range of model parameters that could reliably have produced the data. For example, using their method with the Sarmanov-Lancaster family, we would obtain a "volume" representing all possible Sarmanov-Lancaster models that could have produced a given response. While certainly not ideal, this method would give us a way to gain some insight into dependencies in large populations given the limited data available to us.
Bibliography


Appendix A

Derivation of the Sarmanov-Lancaster Model

Let \((\Omega, \beta, P)\) be a probability space, \(\Omega \subseteq \mathbb{R}\), and consider the space \(L^2(\Omega)\) of square integrable functions on \(\Omega\), with the usual inner product

\[
\langle f, g \rangle = \int \overline{f(x)}g(x)dP(x) = \mathbb{E}[f(x)g(x)],
\]

(A.1)

for any \(f, g \in L^2(\Omega)\). More formally,

\[
L^2(\Omega) = \{ f : \Omega \to \mathbb{R} \text{ s.t. } \|f\| < \infty \},
\]

(A.2)

where \(\|f\|^2 = \sqrt{\langle f^2, f^2 \rangle}\). This defines a Hilbert space, and consequently any function \(f \in L^2(\Omega)\) has an orthonormal series expansion [66]. Hence, letting \((\psi_i)\) be a complete orthonormal sequence in the Hilbert space \(L^2(\Omega)\),

\[
f = \sum_{i=0}^{\infty} \langle f, \psi_i \rangle \psi_i.
\]

(A.3)

and \(\|f\|^2 = \sum_{i=0}^{\infty} |\langle f, \psi_i \rangle|^2\) by Parseval’s theorem.

Next, consider a collection of \(N\) random variables, \(X = (X_1, \ldots, X_N)\), where each variable \(X_n\) takes values \(x \in \Omega\) with probability measure \(P_n\). As before, each variable has a corresponding Hilbert space \(L^2_n(\Omega)\), the space of functions on \(\Omega\) that are square integrable with respect to the inner product induced by the probability measure \(P_n\) (as defined by
equation A.1). Similarly, we can define a complete orthonormal sequence \( (\psi_i^{(n)}) \) in the Hilbert space \( L^2_n(\Omega) \).

We define the probability vector space \( (\Omega^N, \beta, P^{[N]}) \), where \( P^{[N]} = P_1 \times \cdots \times P_N \) is the product measure on \( \Omega^N \). Now, let \( L^2(\Omega^N) \) be the space of square integrable functions on the vector space \( \Omega^N \):

\[
L^2(\Omega^N) = \{ f : \Omega^N \to \mathbb{R} \text{ s.t. } \|f\|^2 < \infty \}.
\]

(A.4)

The set \( (\psi_{i_1 \cdots i_N}) = \left( (\psi_i^{(1)}) \times \cdots \times (\psi_i^{(N)}) \right) \) is a complete orthonormal system in \( L^2(\Omega^N) \), and consequently any function \( f \in L^2(\Omega^N) \) has the series expansion

\[
f = \sum_{i_1, \ldots, i_N} \langle f, \psi_{i_1 \cdots i_N} \rangle \psi_{i_1 \cdots i_N}.
\]

(A.5)

In particular, consider the function

\[
f = \frac{dP_X}{dP^{[N]}},
\]

(A.6)

where \( P_X \) is any probability measure on \( \Omega^N \). This function belongs to the space \( L^2(\Omega^N) \) provided that it is square integrable, i.e. if \( \int_X |dP_X(x)|^2 < \infty \). In that case, \( f \) has the following series expansion:

\[
\frac{dP_X}{dP^{[N]}} = \sum_{i_1, \ldots, i_N} \left( \int_X \frac{dP_X}{dP^{[N]}} \psi_{i_1 \cdots i_N} dP^{[N]} \right) \psi_{i_1 \cdots i_N}
\]

\[
= \sum_{i_1, \ldots, i_N} \alpha_{i_1 \cdots i_N} \psi_{i_1 \cdots i_N}.
\]

(A.7)
Here, $a_{i_1 \ldots i_N} = \mathbb{E}[\psi_{i_1 \ldots i_N}]$, and the expectation is with respect to the probability measure $P_X$. Finally, adopting the convention $\psi_0^{(n)} = 1$ for all $n$, we obtain the expansion

$$dP_X = dP^{[N]} \left[ 1 + \sum_{i_1, \ldots, i_N} a_{i_1 \ldots i_N} \psi_{i_1 \ldots i_N} \right], \quad (A.8)$$

where it is understood that the summation is only over those indices that include at least one non-zero element. Additionally, from Parseval's theorem,

$$\phi^2 = \left\| \frac{dP_X}{dP^{[N]}} \right\|^2 - 1 = \sum_{i_1, \ldots, i_N} a_{i_1 \ldots i_N}^2. \quad (A.9)$$