Encoding natural priors in neural populations

Benjamin Poole
<ben@cmu.edu>

Undergraduate Thesis
Advisor: Tai Sing Lee

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

April 29, 2011

Abstract

Bayesian theories of the brain have provided insights into perception, but the underlying neural mechanisms which could implement these computations remains unknown. To perform Bayesian inference, sensory information must be combined with prior information about the natural world. We investigated how these natural priors could be learned and encoded in populations of neurons in primary visual cortex. We found that the distribution of neuronal tuning properties for depth-tuned neurons was very similar to the distribution of depths occurring in natural scenes. This finding is consistent with the hypothesis that neurons are performing optimal sampling of the natural environment based on the information maximization principle. By using the priors encoded in the tuning properties of neuronal populations, we were able to develop a framework for performing Bayesian inference in the brain.

Contents

Chapter 1

Introduction

1.1 Overview

Numerous psychological studies have shown that human observers are able to optimally integrate noisy sensory stimuli with prior information and resolve ambiguity [14, 8]. These studies show that humans can perform optimal Bayesian inference in many tasks, and contexts. However, the neural mechanisms which underly these computations remains unknown. To perform Bayesian inference, the brain must store and utilize priors, which encode our prior beliefs, knowledge, and experience about the world. Here we evaluate the encoding of prior information in primary visual cortex, and the role that this prior plays on information encoding.

Before we can adress how priors could be encoded, we must have a better grasp on the representations used by neural populations in sensory systems such as primary visual cortex. Sensory systems in the brain play the crucial role of connecting an individual to the outside world. These systems have the job of encoding and representing external stimuli by converting analog input (such as brightness of light on the retina) into binary spiking responses. The set of spiking responses of a population of neurons is thought of as the neural representation, or neural code for that population. Understanding what code is used by populations of neurons, and what information they encode remains an open question in neuroscience. Until we understand how the activity of neurons encodes information, we cannot reliably understand the types of computations being performed by the brain.

One of the tremendous difficulties in deciphering the neural code is the tremendous number and diversity of neurons in the human brain. The numerous biological cell types, structures, and activity patterns create an incredibly complex system. Modern recording techniques can only capture a small fraction of this activity, but these samples of the population can still yield insights into the properties of the brain. When looking at a small (1mm cube) patch of primary visual cortex, there still exists a tremendous diversity in neural properties. Some neurons will respond more to colors, or textures, whereas others will respond to edges, curves, or binocular signals combined from both eyes. The diversity in properties makes the neural code much more complicated to decipher, but also allows for a more robust representation of the external world[1]. Understanding how this diversity comes about and how it aids in information coding has proved to be a difficult task.

Through evolution and learning, the brain has developed diverse neural populations which operate to encode information about the environment. But what principle has guided this learning? In other words, what is the information that this population is trying to encode, and how does its structure and diversity aid in achieving that goal? If we can learn what information a population of neurons is encoding, we will have a better idea of the function the population serves.

Many prior studies have postulated that the function of sensory systems is to provide a maximally informative representation of the world while limiting energy usage [14, 11, 9]. The classic study by Olshausen and Field demonstrated that receptive fields in primary visual cortex (V1) match the bases learned from independent component analysis of natural images [11]. These learned bases indicate that the receptive fields of brightness-tuned neurons in primary visual cortex are encoding a sparse representation of the natural images that maximizes the information content.

However, very little work has gone into relating how other properties of the natural world may influence the properties of the brain. Here we have attempted to understand how statistics of depth in the natural would are related to properties of neurons encoding depth in primary visual cortex. However depth is not explicitly encoded in visual cortex, but instead is derived from disparity, which measures the discrepancy between where an object is projected onto your left eye and where it is projected onto your right eye. In this thesis, we address the relationship between disparity in the natural world and the brain, and how priors for disparity could be encoded in neural populations.

We found that the distribution Fisher information in disparity-tuned neurons was nearly identical to the distribution of disparities in natural scenes. This result indicates that disparity-tuned neurons in primary visual cortex utilize a representation which maximizes the information about the stimulus in the external world. Furthermore, this distribution of Fisher information represents an encoding of prior information in the neural population. This bias in Fisher information may be the brain's way of using this prior for Bayesian inference. Further work needs to address what types of algorithms could utilize this bias in Fisher information to aid in Bayesian decoding.

We also evaluated the role of temporal dynamics in information encoding, and found that disparity-tuned neurons increase the information encoded about the stimulus over time. This finding indicates that even in primary visual cortex, populations of neurons are integrating information about the stimulus over time. Thus low-level cortical areas are not just passively providing feed-forward input to higher layers, but instead may be playing a larger role in our active perception.

When looking at higher-order statistics, we found no correspondence between the co-occurrence statistics in natural rangemaps and correlations between neurons. Thus disparity-tuned neurons in primary visual cortex may not encode these more complex statistical structures.

1.2 Learning priors from nature

Deriving natural statistics from the environment requires a dataset which captures the natural world. Here we analyze 50 rangemaps collected using LIDAR. These images provide a representative example of the types of visual input present in the natural world. We selected 50 rangemaps that had resolutions near 22.5 pixels per degree, and masked out people, cars, and other unnatural objects. Typical images consisted of trees, shrubs, grass, and other natural objects.

To compare with neural data, we converted the depth from the rangemaps into disparity. This conversion requires knowledge of where humans fixate in a particular scene. Here we make the simplifying assumption that a human observer would randomly fixate on any particular pixel in the rangemap with equal probability. This greatly oversimplifies the human fixation distribution, but resembles the empirical fixation depth distribution [3]. For each fixation, we can convert the given depth map into a disparity map. These disparity maps are then a representative example of the types of inputs that the human visual system would experience.

Using these disparity maps, we can compute basic statistics such as the general distribution of disparities found in natural scenes (histogram), or look at the pairwise co-occurrence statistics between certain pixels to learn higher-order structures in disparity.

1.3 Information content of neural populations

Information from the left and right eyes is first combined in the primary visual cortex (V1) to compute disparity. In V1, large populations of neurons are found that respond preferentially to certain disparity stimuli [5]. We performed recordings from V1 of an awake behaving monkey while presenting dynamic random dot stereograms. Each trial contained a binocular movie which showed a disc at a certain disparity for 1 second. These stimuli are essentially constant disparity stimulus, and allow us to measure the firing rate of neurons as a function of disparity. Using a multielectrode array, we were able to simultaneously record from up to 50 disparity-tuned neurons. These simultaneous recordings allow us to analyze the temporal interactions and dynamics of a large group of neurons.

Here we were interested in estimating the information content of this disparity-tuned population of neurons. If the distribution of information contained in the neural population matches the distribution of disparities contained in natural scenes, then this population of neurons may be following the information maximization principle.

Directly computing the information contained in a population of neurons is intractable. To reduce the complexity of the problem, we first assumed that information is contained only in the firing rate of a neuron over the entire trial, and not in the temporal pattern of spiking activity within the trial. With this assumption, the mutual information between the stimulus and the neural response can be computed for small populations. However, we are working with hundreds of neurons and must approximate this information metric. Instead of directly estimating mutual information, we instead estimate Fisher information, which can be used to lower bound mutual information. Fisher information can be computed analytically from the means and standard deviations of neural activity.

However, this approach makes many assumptions about the variability of neural responses and provides extremely noisy results. Neurons with very low firing rates but steep slopes would provide peaks in the information, even though their tuning was extremely weak. These "bad" cells would tend to have predicted Fisher information orders of magnitude higher than the rest of the population, and would tend to have extremely low firing rates. In typical analyses these types of cells were thrown out, but we wanted to keep them in the analysis to estimate the information contained in the *entire* population.

To cope with this difficulty, we approximated the Fisher information through the Cramer-Rao bound. The Cramer-Rao bound is an inequality which states that the Fisher information is greater than the inverse of the variance of any estimator. Thus if we can come up with an estimator for the stimulus (disparity), then we can lower bound the Fisher information contained in the neural population. Any estimator can provide a lower bound, but we seek to saturate the bound to get the most accurate measure of Fisher information. In hopes of achieving a more accurate approximation, we tried using a variety of estimators including: Support vector machines, logistic regression, Bayesian decoding, locally optimal linear estimators, and k-Nearest Neighbors. We found that SVMs achieved the smallest variance, and used the variance of its estimates to approximate the Fisher information of the neural population.

We found that the distribution Fisher information in disparity-tuned neurons was nearly identical to the distribution of disparities in natural scenes. This result indicates that disparity-tuned neurons in primary visual cortex utilize a representation which maximizes the information about the stimulus in the external world. Furthermore, this distribution of Fisher information represents an encoding of prior information in the neural population. This bias in Fisher information may be the brain's way of using this prior for Bayesian inference. Further work needs to address what types of algorithms could utilize this bias in Fisher information to optimally perform Bayesian inference.

1.4 Neural dynamics and correlations

One major flaw in the previous analysis is that spike count data is aggregated over the course of an entire second. Our perceptual capabilities operate at a much faster speed, and thus must be supported by a neural code which can perform inference at short timescales [6]. In chapter 3, we analyzed the information content of our neural population as a function of time. Instead of using spike counts from entire trials, we analyzed 5 to 50ms bins of neural activity over the full 1000ms trial. To model this time-varying neural response we used Generalized linear models (GLMs), which can be used to represent the response of an individual neuron as a linear combination of other factors. We used this model to predict the spiking activity of a neuron over time using the local field potential as well as the spiking history of other neurons in the population [7]. This model was able to capture a large deal of the variability in a neuron's spiking response, and provided more accurate estimates of the information content of a neural population over time. We also used the spike-count based techniques by sliding a window over time and normalizing the firing rate. However, these techniques performed poorly in the temporal domain due to the complex dynamics of neurons. In particular, these models fail to capture the refractory period which occurs after a neuron spikes, and inhibits it from firing again immediately.

In chapter 4, we extend the work on relating natural scene statistics to neural properties by looking at the second-order statistics of pixel co-occurrence and pairwise correlations. Based on our findings in chapter 2, we anticipated that neural populations may be optimally encoding higher-order structures such as surfaces and planes in the 3D world. For example, if surfaces are generally tilted away from the viewer (i.e the ground plane), then neurons which encode the top and bottom of the visual field may be negatively correlated. To evaluate this hypothesis, we first computed co-occurrence statistics from the rangemap data. This allowed us to create a prior over pairs of disparities, instead of just a single disparity as in chapter 2. To compare this prior with the neural data, we must estimate the connectivity between each pair of neurons. An estimate of the connectivity, or coupling strength, between pairs of neurons can be found using the GLM used in chapter 3. The coupling strength provides a measurement of the influence of one neuron on another, but we cannot know whether these neurons are anatomically connected. We found that there was no correspondence between the predicted correlation derived from the co-occurrence statistics, and the measured coupling strength determined by the GLM. This finding indicates that disparity-tuned neurons in primary visual cortex may not encode these statistical structures. Instead, it may be the job of neurons in higher-level cortical areas to capture these more complex relationships and encode the associated priors.

Chapter 2

Natural and neural priors

2.1 Datasets

2.1.1 Neural Recordings

Information from the left and right eyes is first combined in the primary visual cortex (V1) to compute disparity. In V1, large populations of neurons are found that respond preferentially to certain disparity stimuli [5]. We performed recordings from V1 of an awake behaving monkey while presenting dynamic random dot stereograms. Each trial contained a binocular movie which showed a disc at a certain disparity for 1 second. These stimuli are essentially constant disparity stimulus, and allow us to measure the firing rate of neurons as a function of disparity. Using a multielectrode array, we were able to simultaneously record from up to 50 disparity-tuned neurons. These simultaneous recordings allow us to analyze the temporal interactions and dynamics of a large group of neurons.

We recorded from a total of 958 neurons over 12 days. During each day, we presented 11 different disparities at least 30 times each. For each neuron, we computed the mean firing rate as a function of the stimulus (disparity) to get tuning curves, $f_i(s)$, and we also computed the standard deviation as a function of the stimulus, $\sigma_i(s)$. We then used a 1-way ANOVA with $p < 0.05$ to select cells with significant disparity tuning.

We found 202 neurons that were significantly tuned to disparity. For each of these neurons, we fit Gabors to the tuning curve shape, and looked at the distribution of Gabor parameters for the 202 cells. The distribution of preferred disparities (the center of the Gaussian envelope for the Gabor) exhibits a strong bias towards 0 disparities with very few neurons outside of ± 1 degree disparity.

We also found that the variability of these disparity-tuned neurons was higher than typically found in cortex. In general neurons in cortex are thought to exhibit Poisson-like variability, with a Fano factor (ratio of the variance over the mean) of around 1. In our dataset we found that the majority of cells had Fano factors greater than 1, indicating tremendous variability.

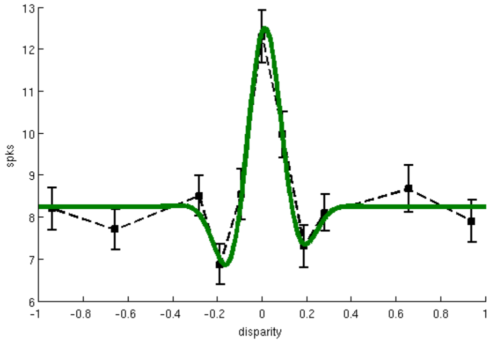

Figure 2.1: Example tuning curve. The x-axis is the stimulus (disparity in degrees) and the y-axis is the average firing rate of the cell for that particular stimulus. The black points correspond to measured disparities (with error bars representing two standard deviations), and the green line is the Gabor fit to this tuning curve.

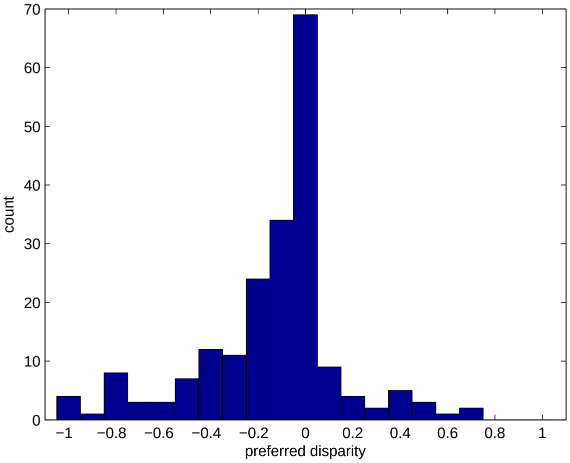

Figure 2.2: Distribution of preferred disparities for 202 neurons

We believe this variability is actually due to the dynamic stimuli we are presenting. Over the course of a trial we are actually presenting 11 slightly different random dot stereograms to elicit a greater response. The refreshes for each frame of the stimulus movie causes an increase in the firing rate of the cell, an introduces more variability into the spike count. For some cells, the refresh of these frames greatly modulated their firing rate.

2.1.2 Rangemap Images

The database consists of 50 optical images and associated range maps. They have been selected to have close to 22.5 pixels per degree. People and cars moving through the scene were masked out, and multiple scans were averaged together for greater accuracy and fewer invalid pixels. These

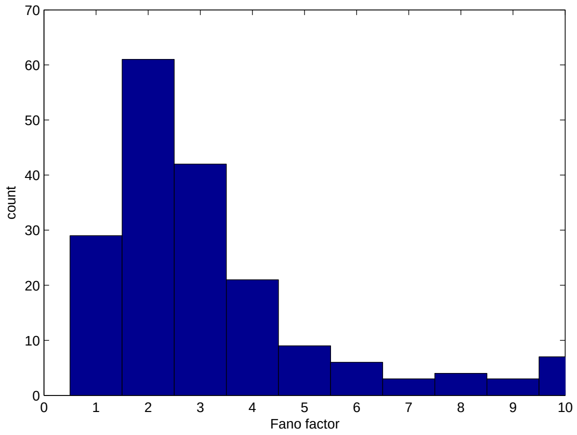

Figure 2.3: Distribution of Fano factors for disparity-tuned cells

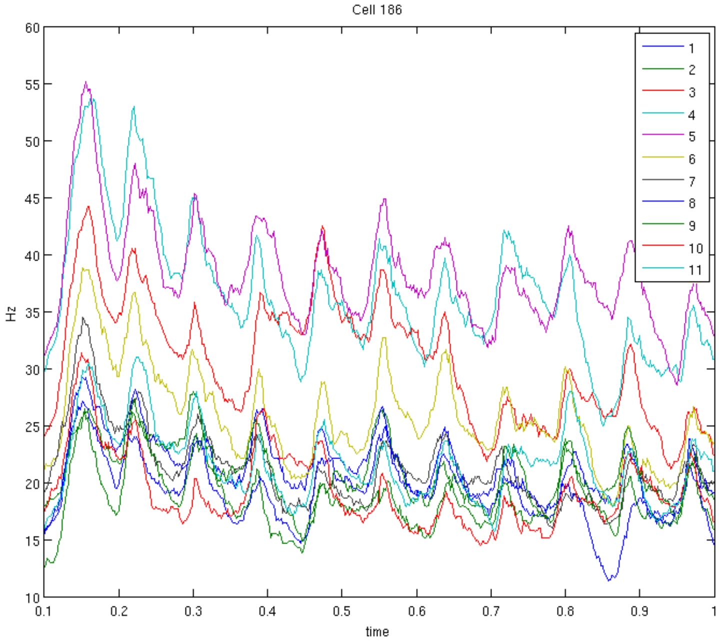

Figure 2.4: An example neuron who's mean firing rate over time is greatly modulated by the refreshes of the dynamic random dot stereogram stimulus. Each line represents the mean firing rate over time for one of the 11 different disparity stimuli.

images were taken of "natural" scenes, primarily containing fields, trees, and shrubs. An example image pair is shown below.

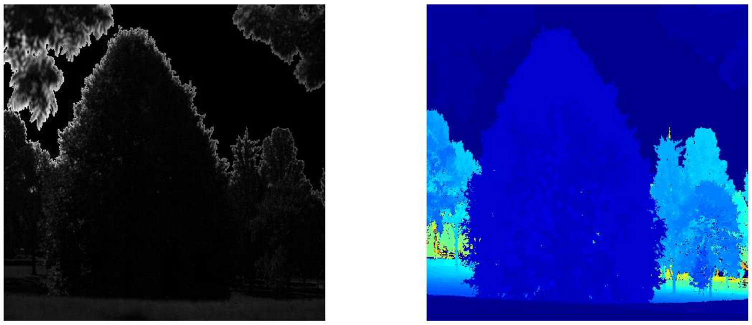

Figure 2.5: Example image pair from natural image database. The left image shows the optical intensity image, and the right image contains the depthmap. Cool colors represent nearer points, and warmer colors represent further points. Regions with no signal, such as the sky, were masked out from the analyses.

2.1.3 Converting Depth to Disparity

To convert depth from the rangemaps to disparity, we must have an optical model of the human eye. Using this model, for any fixation point in an image, we can compute the horizontal disparity for any point in the horizontal plane.

We used the same model as (Liu et al., 2008) which approximates the human eye as a perfect sphere with its center at its nodal point. The interpupillary distance is assumed to be 0.065m, with nodal points at $(-0.0325, 0, 0)$ and $(0.0325, 0, 0)$. The observer/camera is assumed to be pointing in the negative z direction. See Figure 1 in Liu et al., 2008 for a picture (reproduced without permission below).

Consider some fixation point, $F = (x_f, y_f, z_f)$. Let O_c be the midpoint between the two eyes $(0, 0, 0)$. We assume mid-sagittal fixation, that is $x_f = y_f = 0$. The distance from O_c to F is then just z_f. We can find the disparity,d, of an arbitrary point $P = (x_p, y_p, z_p)$, using the following equations:

$$\begin{aligned} d &= \beta_r - \beta_l = \alpha - \phi \\ \alpha &= 2\text{atan}(-0.0325/z_f) \\ \phi &= \text{atan}\left(\frac{-x_p - 0.0325}{z_p}\right) - \text{atan}\left(\frac{-x_p + 0.0325}{z_p}\right) \end{aligned}$$

We need to determine a fixation point in an image before we can compute horizontal disparity. Thus to determine the distribution over disparity, we first need to decide on which fixation points to use. Here we make the simplifying assumption that a view is ranodmly fixating at any point in the image with uniform probability. This assumption is not correct, as certain visual features such as edges are more salient. Future work will need to address this shortcoming.

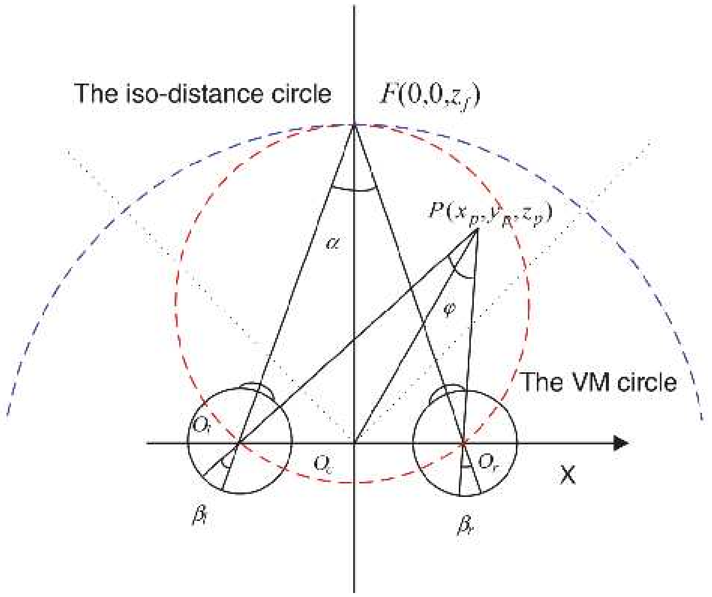

Figure 2.6: Optical model of the human eye. Taken from Bovik 2008

Given a fixation point, we can use the equations stated above to compute the horizontal disparity at any point along the row in which that point belongs in the range image. We can repeat this process over and over again, randomly sampling fixation points and computing the disparity. We can then compute a histogram of the disparities to come up with our natural prior over disparities.

The resulting distribution is roughly Laplacian-shaped, with most of its mass concentrated at 0 and fat tails.

2.2 Connecting Natural Priors to the Brain

Through evolutionary pressures and adaptation, the brain has become optimized to encode its environment. The statistics and properties of the environment can shape the distribution of neural responses, but what exactly is this mapping?

In the case of one neuron, Laughlin proposed a simple equal response criteria, where the goal of a single neuron is to distribute equal regions of its response to equal probability regions in stimulus space. In the case of a population, one might expect that the population response should contain as much informatoin as possible about the stimulus. Furthermore, we may want to allocate more resources to stimuli which have a higher probability of occuring, and less resources to stimuli which have a lower probability. This intuition of allocating resources proportional to a prior distribution has showed up in a variety of fields, from vector quantization in electrical engineering to importance sampling in statistics. What this means for a population of neurons is that they would choose to have more neurons encoding regions of high probability. If we think of the preferred disparity

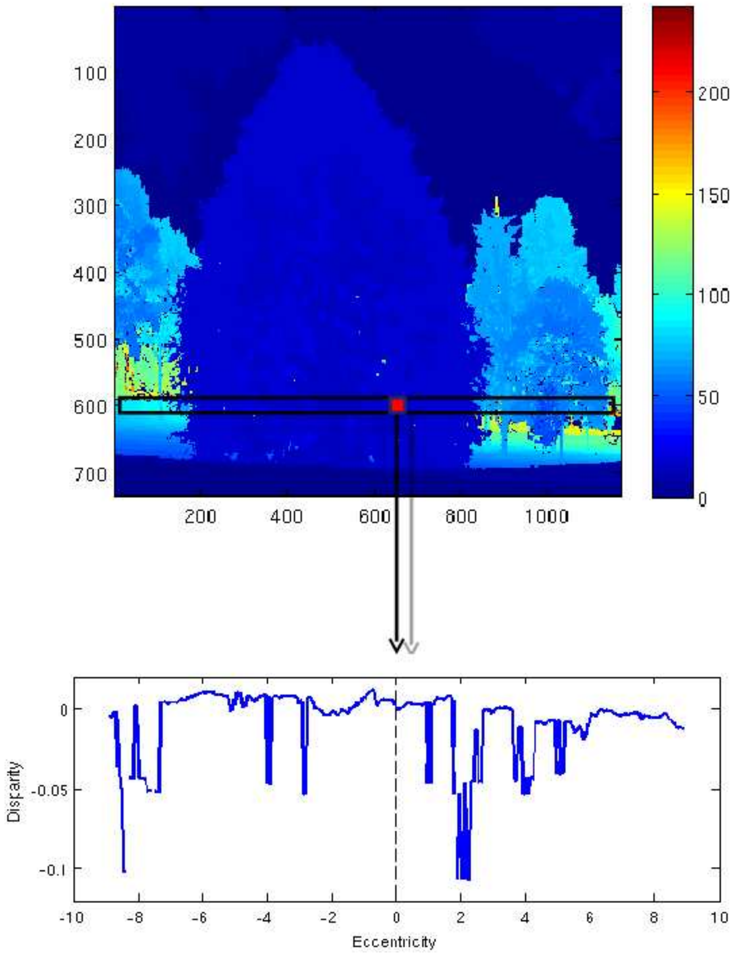

Figure 2.7: Example of computing disparity from a fixation point. The red square in the range image represents the current fixation point. We can then select the depth data to the left and right of the fixation point and use it to compute the distribution over disparity as a function of the eccentricity (horizontal distance in degrees) from the fixation point.

(the point of maximum slope or maximum firing rate) for a particular neuron as the resource which it encodes, then we may expect the distribution of preferred disparities to match the distribution of disparities in natural scenes. When we superimpose these distributions, we see that they are somewhat consistent but do not match well.

The problem with this approach is that it assumes that a neuron only encodes information at its preferred disparity. However, the flanks of a neuron's tuning curve can also encode a great deal of information! Knowing that a neuron is not firing tells you that the stimulus is not at its peak. Thus if we want to measure information in a neural population we will need to use a more global metric.

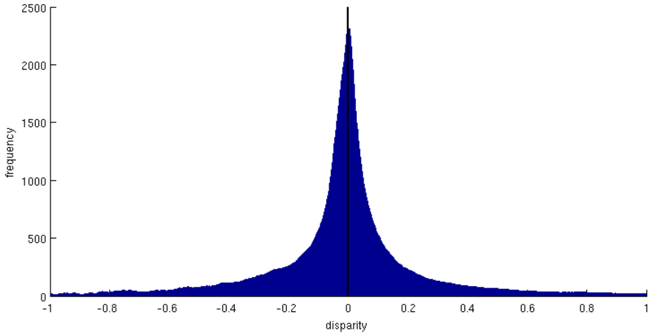

Figure 2.8: Distribution of disparities in natural scenes. Most disparities are near 0, with less mass concentrated in the tails.

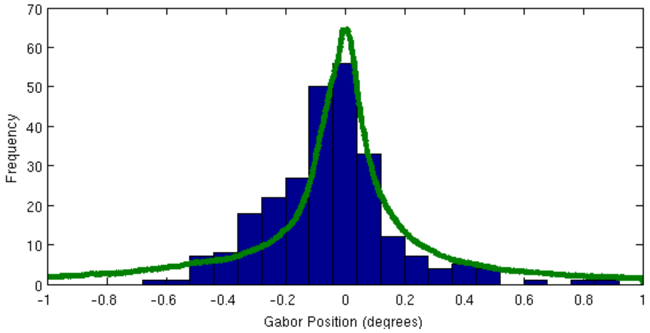

Figure 2.9: Scene statistics distribution superimposed on the distribution of preferred disparities from the neural population. We see that the yare similar in their over-representation of 0 but do not match very well

2.3 Information content of neural populations

To assess the performance of a decoding algorithm, we would like to know how close it is to the optimal decoder. If a decoder can recover all the information contained in a neural population about the stimulus, then it is said to be optimal. The information stored in a neural population thus provides a bound on how accurately we can decode. Similarly, we can use an optimal decoder to estimate the information content of a neural population [10, 12].

Due to noise in the stimulus as well as the encoding process in the brain, the spike counts for a neuron on any particular trial will vary. These noisy spike counts will in turn lead to noisy estimates of the stimulus. To quantify the performance of a decoding algorithm, we can look at the mean square error (MSE), which measures the squared difference between the estimated stimulus value and the true stimulus value. This MSE term can be decomposed as the sum of two quantities: bias and variance. The bias term measures the difference between the mean estimate of the stimulus and the true stimulus, and reflects systematic errors in the decoder which always over or under-estimate the stimulus value. The variance term measures how noisy the estimated stimulus value is between trials. Ideally the bias should be 0 and the variance should be as small as possible. This would imply that the stimulus estimate was always close to the true stimulus value.

The variance term can be related to Fisher information via the Cramer-Rao bound. Fisher information is defined as the expected value of the derivative of the log-likelihood function squared:

$$J(s) = \mathbb{E}\left(\left(\frac{\partial}{\partial s}\log P(\mathbf{r}|s)\right)^2 \mid s\right)$$

where $\mathbf{r}$ is the random vector of spike counts, and s is the stimulus value. The Fisher information, $J(s)$, is a function of the stimulus and measures the amount of information that $\mathbf{r}$ encodes about s. Fisher information is related to discriminability, mutual information, and the variance of estimators [4]. Here we are interested in the Cramer-Rao bound, which specifies that the smallest variance that any unbiased estimator,$\hat{s} = T(\mathbf{r})$, can achieve is inversely proportional to the Fisher information:

$$\mathrm{Var}(\hat{s}) \geq \frac{1}{J(s)}$$

If $\hat{s}$ is biased, where $b(s) = \mathbb{E}\left(\hat{s}\right) - s$, then the bound becomes:

$$\mathrm{Var}(\hat{s}) \geq \frac{\left(1 + b'(s)\right)^2}{J(s)}$$

Given the Fisher information we can compute the smallest variance of any estimator. However, we can also use this bound to compute the Fisher information. Given any unbiased estimator, $\hat{s}$, we know that:

$$J(s) \geq \frac{1}{\mathrm{Var}(\hat{s})}$$

Thus the variance of any esimator provides a lower bound on the FIsher information. If we have a non-optimal estimator, then it's variance will be greater than the minimal variance, and the predicted Fisher information will be smaller. One can thus try using a multitude of estimators, and use the one with the lowest variance to bound Fisher information. This technique for estimating information has been used in a variety of papers and allows one to avoid analytically computing the Fisher information [2, 13].

2.3.1 Analytical Computation of Fisher information

If we assume that the neural response variability for a particular stimulus, $\mathbf{r_s}$, is drawn from a simple distribution then the Fisher information can be computed analytically.

For independent Gaussian variability, the Fisher information can be computed from the tuning curves (f_i) as:

$$J(s) = \sum_{i=1}^{N} \frac{f_i'(s)^2}{\sigma_i(s)^2}$$

For independent Poisson variability, this reduces to:

$$J(s) = \sum_{i=1}^{N} \frac{f_i'(s)^2}{f_i(s)}$$

Additionally, for the case of correlated Gaussian variability (population response has a multivariate Gaussian distribution) we have:

$$J(s) = f'(s)^T \Sigma^{-1}(s) f'(s) + \frac{1}{2}\text{Trace}\left(\Sigma'(s)\Sigma^{-1}(s)\Sigma'(s)\Sigma^{-1}(s)\right)$$

where $f(s)$ is a column vector of tuning curve means, and Σ is the covariance matrix.

These equations for the Fisher information all require the computation of the derivative of the tuning curves. As we only have discrete points, we need to somehow estimate the form of the tuning curve to get the derivative. I've tried three different approaches:

1. Linear interpolation between points, $f'(s) = \frac{f(s+\delta)-f(s-\delta)}{2\delta}$
2. Cubic interpolation between points
3. Gabor fit of tuning curve, which has a closed form solution for the derivative

The first two approaches, linear and cubic interpolation, would often lead to values of the Fisher information which were far too high. The derivative would be much greater than the variance, and the Fisher information for that particular cell would be orders of magnitude higher than the rest of the population. These bad cells tended to be ones with low mean firing rates, but having a "noisy" peak at some disparity, so the variance is low but the derivative is reasonably high.

Using the Gabor fits led to a more stable result, but many of our cells are not Gabor-like. So the mean firing rate and the derivative were fairly inaccurate compared to the data.

For the multivariate Gaussian case, we have to compute both the covariance matrix and the derivative of the covariance matrix. This requires fitting a tremendous number of paramaters with a relatively small amount of data. The covariance matrices are almost certainly overfitting, and we need to interpolate these covariance matrices to get estimates of derivatives (as well as the covariance matrix at points which were not tested). The resulting Fisher information was sharper compared to the independent case.

Thus assuming response variability and estimating Fisher information analytically led to very unstable results. This is most lkely due to the incorrect assumptions of the form of the response variability.

2.3.2 Estimating Fisher information

Due to the instability of the analytical approach, we decided to estimate the Fisher information using the Cramer-Rao bound described earlier. I used a variety of regression and classification techniques to predict the stimulus value from the neural response, and computed the estimated Fisher information as the inverse of the variance of the estimates. The final estimated Fisher information was the technique which

The first technique I tried was simple maximum likelihood decoding. If we assume some form of neural response variability, such as $\mathbf{r} \sim N(\mathbf{f}(s), \sigma(s))$, then we can find the stimulus value which maximizes the likelihood. For computing the Fisher information at one particular disparity, I first computed the mean and variance of the tuning curve for each neuron from 29 out of 30 trials. The mean and variance of the tuning curves were then used to do maximum likelihood decoding on the remaining trial. If we repeat this for each of the 30 trials, we get 30 different estimates of the true stimulus, one for each trial. The inverse of the variance of these estimates is then the approximate Fisher information. The actual computations are:

$$\begin{aligned} \hat{s} &= \arg\min_s P(\mathbf{r}|s) \\ &= \arg\min_s \prod_i P(r_i | f_i(s), \sigma_i(s)) \\ J(s) &\approx \frac{1}{\mathrm{Var}(\hat{s})} \end{aligned}$$

In order for this to be an accurate estimate of Fisher information, the estimator must be close to efficient (lowest variance unbiased estimator). Our estimate of Fisher information gives a lower bound on the true Fisher information, but I'm not sure how close we are to saturating that bound. If we are close, then our estimate is good, if not, then our estimate could be awful. To verify that the MLE was close to optimal, we tried a variety of other classifiers including k-NN, logistic regression, ridge regression, and SVMs. We found that SVMs provided the best results, but only increased Fisher information by about 10%,thus we used the MLE results for our estimated Fisher information as the distribution of information was similar across multiple recording days.

The final estimated Fisher information matches the scene statistics disparity distribution extremely well. It has fat tails, unlike the distribution of preferred disparities, and has a similar shape near 0 disparity as well. Thus the information encoded in disparity-tuned neurons in V1 is proportional to the prior distribution over disparity in the natural world. This finding is consistent with the information maximizatoin principle proposed by Barlow and the recent theoretical work by Simoncelli's group.

The correspondence between Fisher information and the natural prior is much better than between the preferred disparity distribution and the natural prior. Incorporating more global properties of the tuning curve instead of just the peak location was the key reason for these results. We believe that more neural data analysis needs to take advantage of global metrics of information instead of simply looking at the distribution of properties of individual tuning curves.

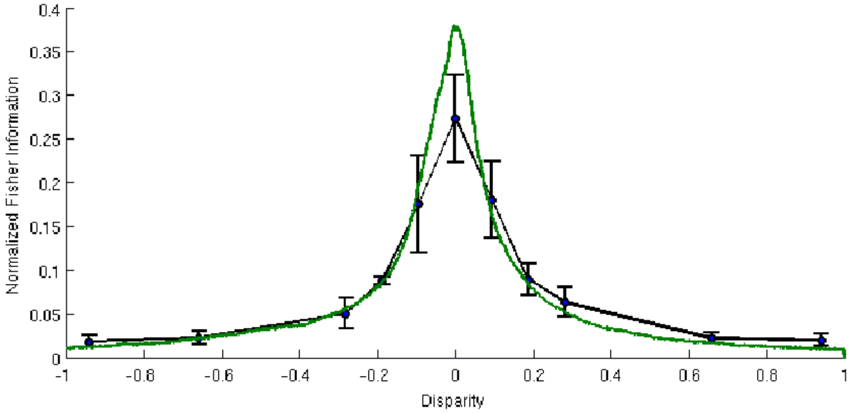

Figure 2.10: Fisher information as a function of disparity. The green line is the disparity distribution derived from the rangemaps. The black line is the normalized Fisher information derived from 11 days of neural recordings. Standard error bars computed using bootstrapping.

2.4 Effect on Neural Representations

Although we have identified that the distriubtion information in the neural population is proportional to the scene statistics distribution, we have not yet determiend how this bias effects neural representations. To evaluate the role of this information bias on neural coding, we performed maximum likelihood decoding using the probabilistic population code framework. This framework argues that populations of neurons are not estimating a single estimate of the stimulus, but are instead representing a probability distribution over stimulus space. Thus neural activity is able to encode estimates of the stimulus as well as the certainty associated with that estimate. Being able to encode uncertainty allows one to optimally combine noisy stimuli and perform Bayesian inference. Recent research has shown that under certain assumptions, inference using probabilistic population code only requires adding together the responses from different neural populations [10].

For each trial, we computed the posterior distribution over the stimulus given the neural response we observed during that trial. We then computed the geometric mean of these posterior distributions to identify the typical shape of the posterior distribution for that particular stimulus. We found that near 0 disparity, the posterior distributions were sharp, indicating a high degree of certainty in the estimate. At more distal disparities, the posterior distributions were generally wider, indicating less certainty. Thus it seems as though the distributions represented by these neural populations are able to incorporate the bias in Fisher information to represent uncertainty.

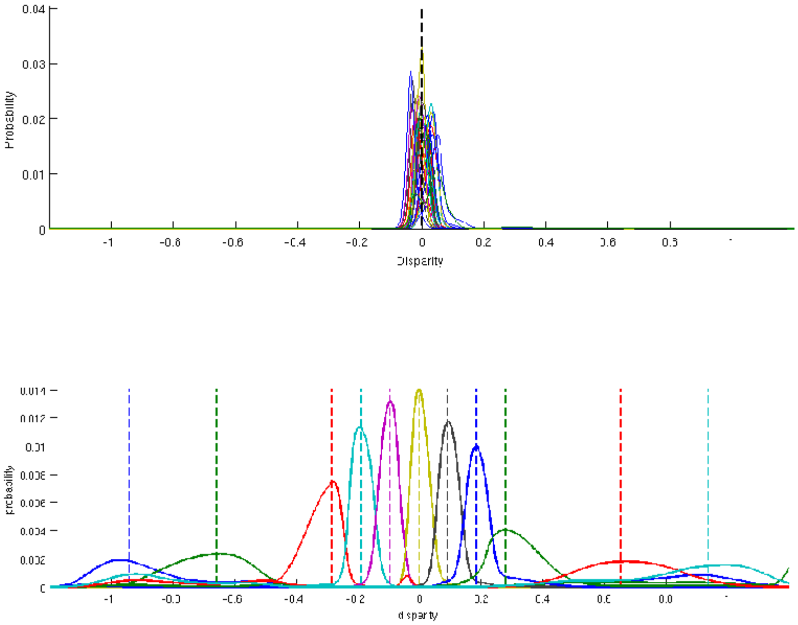

Figure 2.11: Top: Posterior distribution for 30 individual trials from a disparity stimulus of 0. Bottom: Geometric mean of posterior distributions over 30 trials for each of the 11 disparities. Note that the posteriors near 0 are sharper, while more distal disparities have a broader posterior.

Chapter 3

Second-order statistics

3.1 Overview

Given that the first-order statistics of disparity are reflected in the Fisher information of the neural population, we were interested in evaluating whether the second-order statistics from natural scenes were also encoded neural populations. One potential way for these interactions to be encoded is in the correlated activity of neurons. If certain disparities are more likely to co-occur than others, then the neurons encoding these disparities might have stronger connectivities. This connectivity could be learned through a Hebbian-like learning rule where neurons firing at a similar time would grow stronger connections. To evaluate this hypothess, we first needed a framework to estimate connectivity between pairs of neurons. We applied a L_1-regularized pairwise MRF to model the neural data and estimate these connectivities. [1]

3.2 Markov Random Fields

One way to understand the dependencies between action potential activity, or firing, of one neuron with those of its neighbors is to treat each neuron as a node in a graphical model and model the interactions between nodes using a Markov Random Field (Schneidman et al. 2006; Truccolo, Hochberg, and Donoghue 2010). The activity of each of the neurons is collapsed across time into discrete time bins (Figure 1). If a neuron fires in a particular bin, it is registered as a '1' and '0' otherwise; so the firing of all the neurons within a time bin constitutes a binary word.

We modeled the probability distribution over these binary strings using a MRF. The MRF framework presents a way to represent these complex patterns in a compact form. These models can capture a variety of complex interactions, such as correlations due to common input. An example of a learned MRF from a group of neurons with common input should have edges between neurons

[1] The majority of this chapter was completed as a final project for 10-708 with Shreejoy Tripathy.

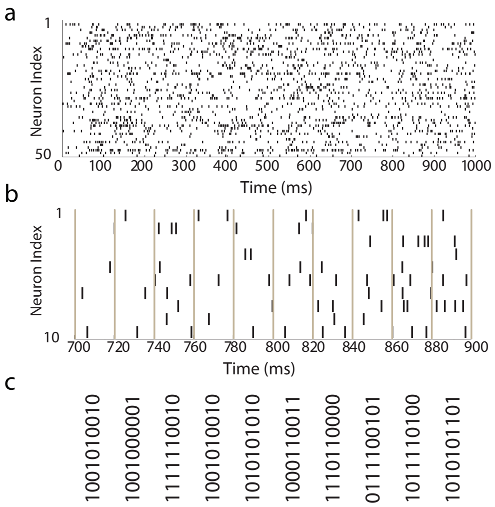

Figure 3.1: Example neural data and discretization into bins. **a)** Spiking data for 50 neurons. Each tick mark represents the time of an action potential, and each row represents spiking data for one neuron over time. **b)** Discretization of neural data into binary strings. The grey lines represent the extent of the time bins (20ms). **c)** The activity of each neuron is converted to a binary string which is 1 if the corresponding neuron has at least one action potential within the bin and 0 otherwise.

sharing common input (Figure 2). The presence of these edges and connections could be used to better understand the relationships and structures among neurons in large populations.

Prior studies have fit MRFs using exact methods on groups of neurons of size ≈ 10. They found that 2nd but not 1st order MRFs were required to fit neuronal data collected from more simple brain regions, like the eye (Schneidman et al. 2006). More recently, other groups have found that 3rd order MRFs were required to represent the distributions collected from data from monkey visual cortex (Ohiorhenuan et al. 2010). However, these results were based solely on the results on training data. To our knowledge, no group has compared test set accuracy using this technique. It is possible that these higher order models are just overfitting the data and that 3rd order models are not required for modeling neural activity. In this project, we attempted to answer this question as well as provide insights into scaling these models up to larger network structures.

3.3 Complexity of Neural Connectivity

To evaluate the complexity of the factors we needed to model neural activity, we looked at a large number of small subgroups of neurons. From the 50 neurons that we recorded, we randomly formed 500 groups of 10 neurons each and fit MRFs for each group. We used the maximum entropy toolbox provided with the SciPy Python distribution to perform the parameter estimation. Specifically we fit fully-connected 1st, 2nd and 3rd order models using 80% of the data, and then

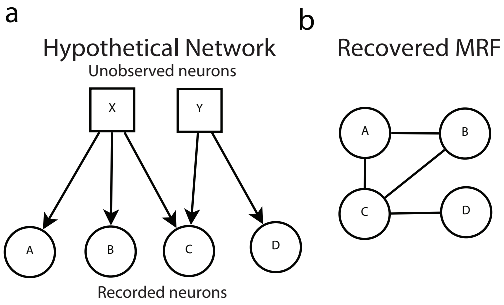

Figure 3.2: Example network and MRF structure. **a)** Hypothetical neural network with unobserved common input influencing activity in recorded neurons. These common inputs cause correlated activity in the recorded data. **b)** Possible recovered MRF from neural data which has learned edges between nodes that share common input.

tested these models on the remaining 20% of the data. We evaluated these models by comparing the probability distribution derived from the data with the probability distribution encoded by the trained model. We used the Jensen-Shannon Divergence to measure the dissimilarity between these two distributes. JS-divergence is a normalized measure of the difference between two distributions:

$$D_{\text{JS}}(P,Q) = \frac{D_{\text{KL}}(P||M) + D_{\text{KL}}(Q||M)}{2} \tag{3.1}$$

$$\text{where } M = (P+Q)/2$$

When the JS-divergence is small, the probability distributions are similar, and we say that error is small. When the JS-divergence is large, the probability distributions are more distinct and we say that the error is large.

We first looked at training set error, which has been the primary metric in evaluating models in neuroscience. Our results were qualitatively similar to those from prior studies. Third order models had the lowest training set error followed by 2nd and 1st order models (Figure 3a, 4a). These results are consistent with the idea that higher order models are able to represent a larger number of probability distributions than lower order models.

When considering testing set error (JS-divergence), we found that while the 3rd order model had the lowest error, the relative difference between the 3rd and 2nd order models was small (Figure 3b, 4b). Furthermore, for 22% of the 500 subsamplings of the 10 neuron groups, the test error was lower for the 2nd order model than the 3rd order model (Figure 5) despite the fact that the 3rd order models had lower training error. This is one indication that the third-order models may be overfitting the training data and may not be required. Another indication of overfitting is that the space of all second order models is a subset of the space of all third order models. Thus any third order model can represent any second order model by just setting third order cliques to be 1. So when the second order model is better, it is likely that the third order model is overfitting.

Our findings indicate that a first order model is not sufficient for modeling the activity of groups of neurons in primary visual cortex. This result makes sense as these neurons share common input

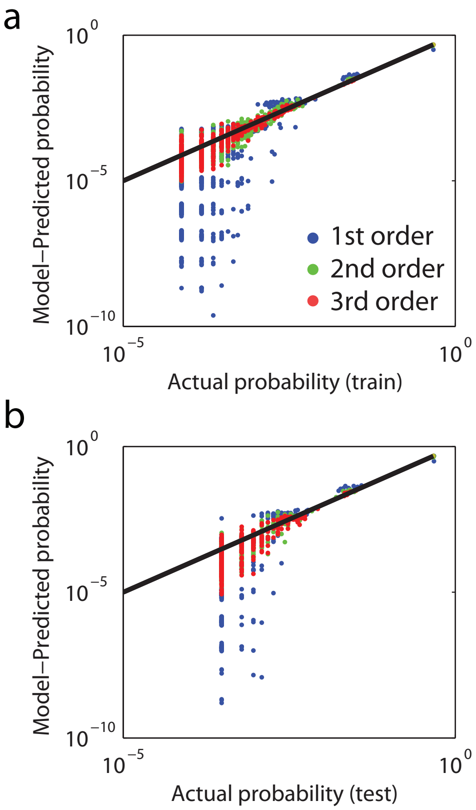

Figure 3.3: Scatter plots of true vs. predicted probability distributions. Each point represents the probability of one of 2^{10} binary patterns in the samplespace. The black line indicates the unity line. Points which deviate from the unity line indicate a difference between the predicted and true probability of a pattern. Colors indicate the order of model: first, second, third (blue, red, green respectively). **a)** true probability based on training data. **b)** true probability based on testing data.

from other groups of neurons which would correlate them. The second and third order models performed similarly, although the third order model was slightly better. Despite this slight increase in performance, we decided to stick with the second order models for two reasons. The first reason for using the second order model was computational tractability. Including all third order factors in our networks tremendously slowed down computation and we did not believe it would be feasible to use them in large networks. Our second reason for not using third order models is that they are not biologically plausible. The types of connections and interactions between neurons in the brain are thought to only include pairs of neurons communicating, and not triplets. Additionally, prior work in our research lab using iterative proportional scaling has shown that third order interactions are not present in this data (unpublished). For these reasons we decided to exclusively utilize second order models for fitting larger networks.

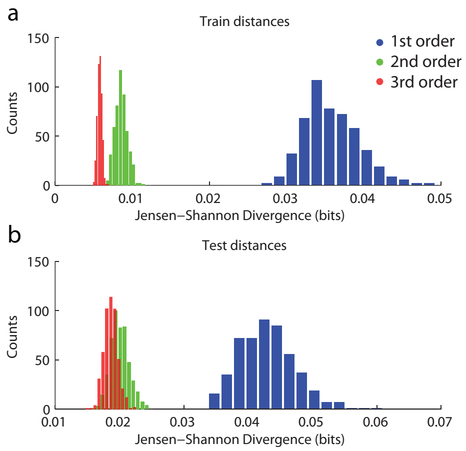

Figure 3.4: Histogram of JS-divergences for 500 10-neuron subsamplings. **a)** JS-divergences between probability distribution derived from training data and probability distributions from model. **b)** same as in **a** but for testing data.

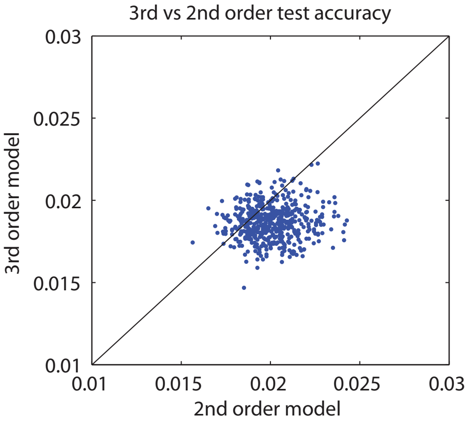

Figure 3.5: Scatter plot of the JS-divergence between the testing distribution and the model distribution for 2nd and 3rd order models. Each of the 500 points represents one 10-neuron subsamplings.

3.4 Using regularization to learn sparse MRFS for large networks

We were interested in extending the framework we used on the 10-neuron networks to larger groups of neurons. However, these larger models would have a much greater number of parameters ($O(n^2)$ for the pairwise models) which would be difficult to fit with our limited dataset. To help prevent overfitting we tried to add an L_1 penalty on the values of the factors. We encountered a number of problems with extending our previous framework in Python. We were not able to achieve convergence using a number of different gradient-based techniques which led to fitting the models being extremely slow.

Due to time constraints and the slowness of our Python implementation, we decided to investigate other frameworks for fitting L_1-regularized MRFs. We found that the UGM Matlab toolbox (`www.cs.ubc.ca/ schmidtm/Software/UGM.html`) provided most of the tools we needed to learn these models. After porting our Python code to Matlab, We were finally able to learn the parameters of L_1-regularized MRFs using loopy belief propagation and the UGM toolbox's projected quasi-Newton algorithm for constrained optimization. This framework also allowed us to place a group L_1 penalty on the edge parameters instead of on all the parameters individually. The regularized objective function we optimized was:

$$\min_{w,v} - \log P(X|w,v) + \lambda_1 \sum_{g} ||v_g||_1$$

Here w are the parameters associated with nodes, v are the parameters associated with edges, g represents the groups (1 for each edge in the model) and v_g is the set of edge parameters associated with group g.

We learned MRFs for 3 different conditions: 1. a model with a first order structure (no edges); 2. a model with a fully connected second order structure; 3. a second order model with sparse structure learned through a group L_1 regularization among factors associated with an edge. We hoped that by enforcing an L_1 penalty, the edges which remained after regularization would be those which were most essential and reflect the conditional independence assumptions inherent in the data. For each model, we divided the data into 80% training data and 20% testing data, and performed 5-fold cross validation on the training data to learn the models.

To assess the performance of each model we used a conditional decoding scheme where we computed the conditional spiking probability of a single neuron given the spiking of all other neurons as evidence. For each testing example we computed the conditional spiking probability for every node in the graph and averaged errors over nodes. We used two error metrics: 1. a simple MAP based decoding accuracy; 2. the predictive power computed as the area under the ROC curve of the conditional spiking probability (Figure 6).

3.5 Large network results

We found that with both decoding metrics the fully connected 2nd-order and L_1 models outperformed the 1st order model (Table 1). Interestingly, while the full and L_1 models had similar levels of decoding accuracy the penalized model had substantially fewer edges. These findings indicate that many of the edges which are retained in the fully connected model are unnecessary and can be removed by regularization with little to no detriment in decoding accuracy. This is also indicated by the L_1 regularization path (Figure 7).

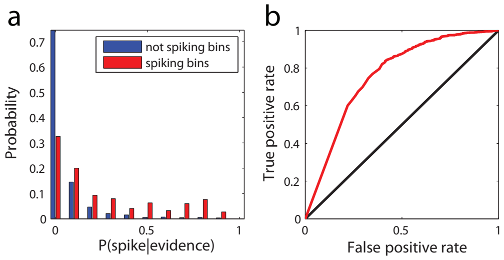

Figure 3.6: Example evaluation for one neuron. **a)** Conditional probability of a neuron spiking given evidence for a single neuron. Histogram computed over all testing examples and sorted into examples where the neuron spiked or did not spike. **b)** ROC curve for the probability distribution in (a).

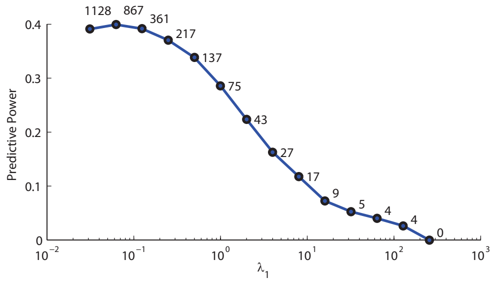

Figure 3.7: Regularization path for cross-validated dataset. Numbers indicate the number of edges in each regularized graph.

We further studied which edges between neurons remain in the graph following regularization. We found that with small L_1 penalties, the graph was almost fully connected with neurons connected to others large physical distances away. By increasing the penalty, the edges which remained were between neurons in close proximity with one another (Figure 8). This result is precisely in line with published anatomical studies (Ohiorhenuan et. al 2010). Thus our methodology is exciting in that it may serve as a novel way to recover the underlying biological circuitry using data collected in this fashion.

3.6 Discussion

Our work has shown that MRFs provide a viable framework for analyzing neural activity. In small networks, we were able to use MRFs to evaluate the required complexity needed to model our the activity of groups of neurons. We found that while first order models were not sufficient, second and third order models provided a good match to the data. However, the third order models required a much greater computational cost, and could not plausibly be fit into biological models

Table 3.1: Final model results on a 49-neuron network for decoding accuracy, predictive power, and number of edges.

MODEL	ACCURACY	POWER	EDGES
1ST ORDER	48.70%	0.00	0
2ND ORDER	64.41%	0.37	1176
L_1	64.34%	0.39	867

of the brain.

By using L_1 regularization we were able to apply MRFs to larger networks of 50 neurons. Although using L_1 regularization did not improve the accuracy of our models when compared to full models, it did greatly reduce the number of edges in the learned networks. This reduction in edges tremendously reduced the computational complexity required in fitting and performing inference on these models. In the future, we hope to use these sparser networks to perform more interesting inference queries in groups of neurons.

Using the L_1 penalty to increase the sparsity of our networks, we found that the edges which remained in the network were those with shorter lengths. This finding indicates that the important connections in our neural networks are between neurons in close proximity. Biological studies have also found that connections between neurons are primarily local, with very few long-range connections (Ohiorhenuan et al. 2010). We believe that using L_1 regularization not only to select the best model, but also to select the best edges could help in understanding the underlying connectivity among neurons in the brain.

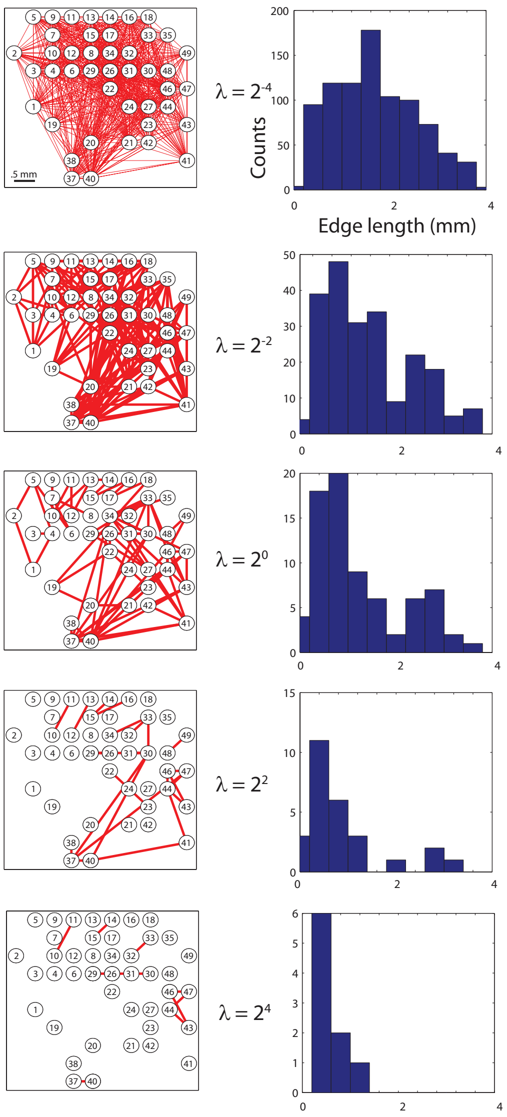

Figure 3.8: Graph structures learned using regularization. The left column indicates the graph structure for each L_1 penalty given in the center. The right column is a histogram of edge lengths between neurons in the graph to its left. As the L_1 penalty is increased, the edges that remain are between neurons in close proximity.

Chapter 4

Conclusions

See: Introduction.

Bibliography

[1] Bruno B Averbeck, Peter E Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature reviews. Neuroscience*, 7(5):358–66, May 2006.

[2] Jeffrey M Beck, Wei Ji Ma, Roozbeh Kiani, Tim Hanks, Anne K Churchland, Jamie Roitman, Michael N Shadlen, Peter E Latham, and Alexandre Pouget. Probabilistic population codes for Bayesian decision making. *Neuron*, 60(6):1142–52, December 2008.

[3] Alan C Bovik and Lawrence K Cormack. Disparity statistics in natural scenes. *Journal of Vision*, 8:1–14, 2008.

[4] N Brunel and J P Nadal. Mutual information, Fisher information, and population coding. *Neural computation*, 10(7):1731–57, October 1998.

[5] B G Cumming. An unexpected specialization for horizontal disparity in primate primary visual cortex. *Nature*, 418(6898):633–6, August 2002.

[6] Adam L Jacobs, Gene Fridman, Robert M Douglas, Nazia M Alam, Peter E Latham, Glen T Prusky, and Sheila Nirenberg. Ruling out and ruling in neural codes. *Proceedings of the National Academy of Sciences of the United States of America*, 106(14):5936–41, April 2009.

[7] Ryan C Kelly, Matthew a Smith, Robert E Kass, and Tai Sing Lee. Local field potentials indicate network state and account for neuronal response variability. *Journal of computational neuroscience*, January 2010.

[8] David C Knill and Alexandre Pouget. The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends in neurosciences*, 27(12):712–9, 2004.

[9] M S Lewicki and T J Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2):337–65, February 2000.

[10] W J Ma and A Pouget. Population Codes : Theoretic Aspects. *Neuroscience*, 7:749–755, 2009.

[11] Bruno a Olshausen and David J Field. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–7, August 2004.

[12] a Pouget, S Deneve, J C Ducom, and P E Latham. Narrow versus wide tuning curves: What's best for a population code? *Neural computation*, 11(1):85–90, January 1999.

[13] Peggy Seriès, Peter E Latham, and Alexandre Pouget. Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations. *Nature neuroscience*, 7(10):1129–35, October 2004.

[14] Eero P Simoncelli. Optimal Estimation in Sensory Systems. *New York*, 2009.