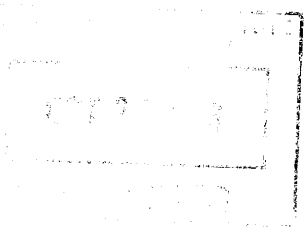
Effects of Neuronal Correlations on Population Decoding and Encoding Models



by

Ami Patel

S.B., Computer Science and Engineering, MIT (2012)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013

Signature redacted

Author .....
Department of Electrical Engineering and Computer Science
August 9, 2013

Signature redacted

Certified by.....
Tomaso Poggio
Eugene McDermott Professor
Thesis Supervisor

Signature redacted

Accepted by .....
Professor Albert R. Meyer
Chairman, Masters of Engineering Thesis Committee

Effects of Neuronal Correlations on Population Decoding and Encoding Models

by

Ami Patel

Submitted to the Department of Electrical Engineering and Computer Science
on August 9, 2013, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In this thesis, we analyze the effect of the correlations in neural activity on the information that is encoded in and can be decoded from a population of neurons. Various noise models describing these correlations are considered - in particular, we use models that take into account the pairwise correlations and other, simpler models that assume shared global additive and/or multiplicative noise factors. The performance of these models on firing rate prediction (encoding) and population decoding are studied. Our analyses show a significant beneficial effect of pairwise correlations on encoding models, with much of this benefit being explained by the global noise models. However, the effects of correlations on decoding vary among our datasets, providing an empirical justification to the theoretical results suggesting correlations can be either helpful or harmful to decoding.

Thesis Supervisor: Tomaso Poggio
Title: Eugene McDermott Professor

Acknowledgments

I would first and foremost like to thank Ethan Meyers for his guidance, both with his ideas and feedback on this specific project and with advice on doing research in general. I'm also grateful to Tomaso Poggio and the other members of CBCL for their technical feedback and help, and also for making CBCL feel like such a welcoming place.

Much of the data used in this thesis was as a result of a collaboration with the research group of Adam Kohn and that of Jim DiCarlo; I would therefore like to thank Adam Kohn and Ha Hong of the DiCarlo Lab for generously sharing their data. The thanks extends to all those in different research groups including CBCL who were involved in the collection and preprocessing of the data, including Nicole Rust, Joel Leibo, Ethan Meyers, Andy Liang, and many others with whom I did not collaborate directly but to whom I am still very grateful.

Finally, I would like to express my gratitude toward my family and friends for their support throughout my MIT career; I would not be where I am without them.

Contents

List of Figures

List of Tables

help address whether decoding strategies implemented within the brain have to take correlations into account, or whether factors that affect correlations - such as training or attention - would have a large effect on decoding accuracy.

From the encoding perspective, we can ask whether we can improve the prediction accuracy by using the firing rates of other neurons in the population as features, and how doing so compares to the standard method of predicting neural behavior from stimulus properties.

The remainder of this chapter summarizes the properties of the datasets we use in our analyses and describes the brain regions from which they were recorded. In Chapter 2, we define and quantify the correlations within the recorded populations. In Chapter 3, we examine the effect of correlations on population decoding, and in Chapter 4, we use correlations within encoding models.

1.1 The Ventral Stream

In our analysis, we use multielectrode recordings from three different areas of the visual cortex: V1, V4, and the inferior temporal cortex (IT). These three regions are part of the ventral stream, which is thought to play an important role in visual object recognition.

V1 Area V1 represents the first stage of visual processing within the cortex. Standard models for neurons in V1 are based on the early work of Hubel and Wiesel (1962). They documented simple cells in V1, many of which were tuned to respond to bars at particular orientations and at particular locations within the visual field. They also discovered some cells, which they labeled as complex, which also responded to bars at particular orientations, but displayed tolerance to the precise location of the bar. A proposed model of this is that complex cells achieve invariance by pooling inputs from simple cells. Simple and complex cells in V1 have been modeled computationally, e.g. by the feedforward, hierarchical HMAX model (Riesenhuber and Poggio, 1999). This model contains a layer of computational units that have the orientation and location

Chapter 1

Introduction

Multielectrode arrays can record the spiking activity of hundreds or thousands of neurons within a localized brain region simultaneously. Techniques for analyzing such data include the creating of encoding models, which predict a neuron's firing rate from external variables, and decoding models, which do the opposite - using the population of neural responses to predict the external variable (Dayan and Abbott, 2005). In the study of neurons in the visual cortex, the external variable would typically correspond to which image was being presented to the animal at that time. Apart from understanding the computations performed by various brain regions, multielectrode array technology along with decoding analysis also has applications in neural prosthetics, for example allowing humans to gain a degree of control over robotic limbs.

The firing rates measured on a given electrode show significant variability from trial to trial; a widespread finding on multielectrode data is that this trial by trial variability is actually correlated across different electrodes. In this thesis, we examine the effects of correlations on the neural analysis methods described above.

The analysis of correlations has implications for population decoding from a practical perspective in order to choose decoding algorithms that optimize for decoding accuracy as well as efficiency, since algorithms that ignore correlations might be more computationally efficient. It can also be used to examine whether the correlations' effects are significant enough that decoding on non-simultaneous recordings could produce misleading results. From the perspective of understanding the brain, it can

IT is based more on shape similarity than semantic similarity.

1.2 Datasets

Table 1.1 summarizes the data used in our analyses, which were from a total of four different microelectrode arrays - with two from different regions of the same animal's visual cortex. For all datasets, the features used for analysis are the multiunit firing rates on each electrode, or recording site, during a 150ms time bin. The time bin is fixed for all data recorded from a particular array, and was chosen to maximize population decoding accuracy. Sample images from the different stimulus sets used with the arrays are shown in Figure 1-1.

Table 1.1: Summary of Multi-electrode Array Recordings

Stimulus Set	Num. Stimuli	Date	Num. Stimulus Repetitions
Array 1 - V1, 131 recording sites			
Drifting Gratings	8		50
Array 2 - V4, 32 recording sites			
		Jan 27	25
Color	48	Feb 04	48
		Feb 05	48
		Feb 13	34
Objects	105	Mar 10	55
		Jan 23	37
		Jan 26	36
Natural Images	100	Jan 28	48
		Feb 03	36
		Feb 06	50
Array 3 - V4, 96 recording sites, and			
Array 4 - IT, 110 recording sites			
		Aug 21	29
Natural Images	300	Sep 01	49
		Sep 02	49

selective properties of simple V1 cells, referred to as the S1 layer. Units in the next layer, known as C1, correspond to complex V1 cells, and have responses equal to the maximum of its input S1 units. The S1 units that are inputs to a given C1 unit are tuned to similar orientations but at slightly shifted locations, thus allowing the C1 unit to capture the invariance properties of complex V1 cells.

V4 V4 was early on characterized as a color processing area; however, further studies have demonstrated the shape-selective properties of V4 neurons, which are more complex than those of V1 neurons and might aid in figure-ground segmentation (Roe et al, 2012). V4 neurons can be receptive to differences in the curvature of object boundaries (Pasupathy and Connor, 2001). In terms of computational modeling, a spectral receptive field that computes responses as a linear function of the orientation and spatial frequency power spectrum of the stimulus has been used to replicate the properties of V4 neurons (David et al, 2006). The HMAX model is also thought to have layers that have similar properties to V4 neurons. In addition to the S1 and C1 layers, there is also a S2 layer which uses inputs from C1 units to exhibit selectivity for features that are more complex that just oriented bars, and a C2 layer whose units perform max operations over sets of S2 units to achieve invariance in a way analogous to C1 units. Cadieu et al (2006) fit parameters of a C2 unit to different V4 neurons to model the shape selectivity properties of the neurons.

IT The inferotemporal cortex is the highest purely visual area in the ventral stream. Studies have found neurons in this area that respond to specific objects, such as faces, with specificity as well as invariance - i.e., the face-tuned neuron might respond similarly to faces of various sizes and positions. Representational similarity analysis (Kriegeskorte et al, 2008) has also been used to show that IT neurons' responses are based on object category - for example, responses to animate and inanimate objects are highly dissimilar. These properties were not explained by computational features such as the C1 and C2 layers of the HMAX model. However, more recent computational analysis by Baldassi et al (2013) suggests that the representation in

For the V4 data, the firing rate between 50 and 200 ms after the stimulus onset was used; for IT, we used the firing rate between 100 and 250 ms after stimulus onset. The optimal time windows for decoding - and therefore the time windows used in our analyses - are different for areas V4 and IT because IT is further along the ventral stream than V4.

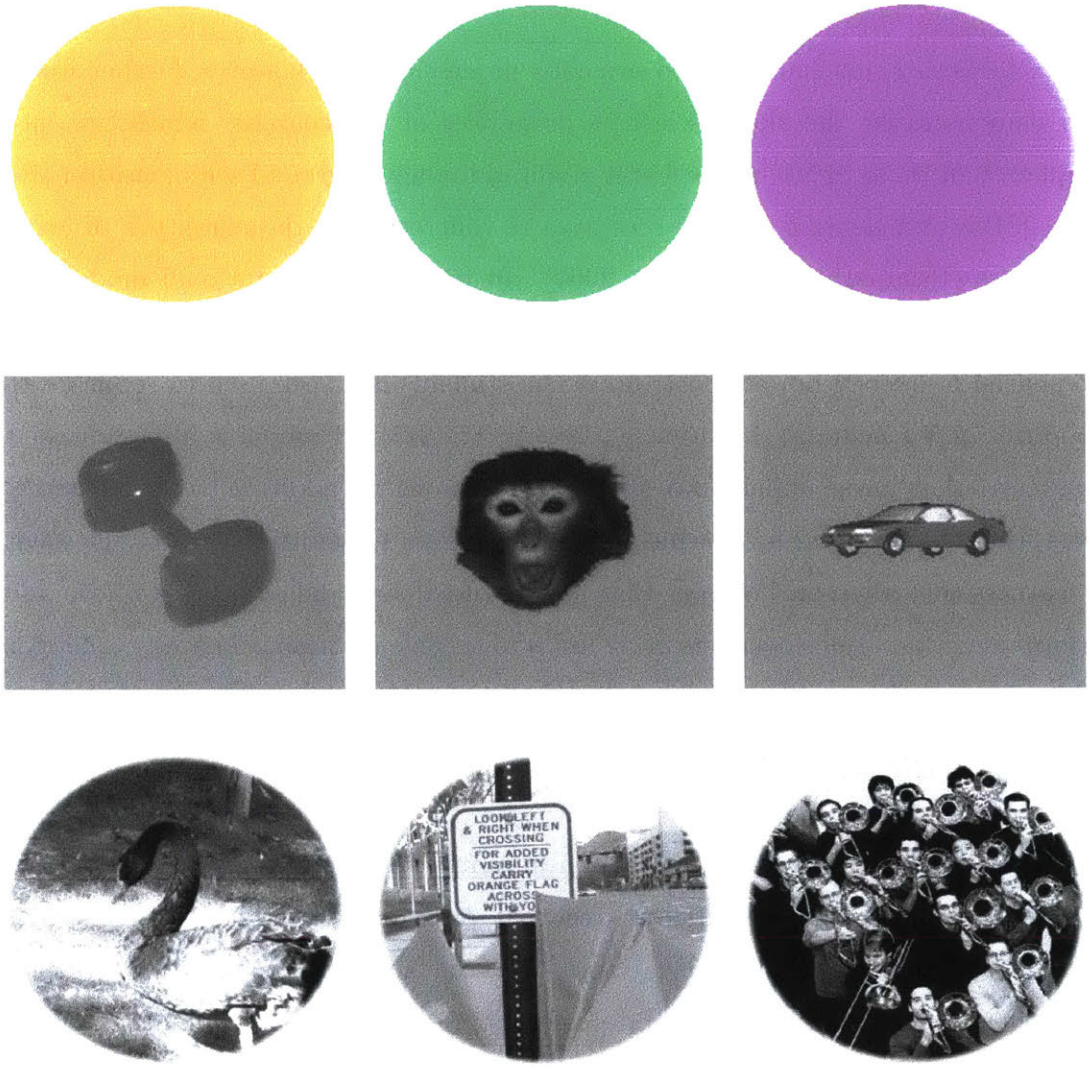


Figure 1-1: Row 1: Sample stimuli from the Color stimulus set used with Array 2. Row 2: Sample stimuli from the Object stimulus set used with Array 2. Row 3: Sample stimuli from the Natural Image stimulus sets used with Arrays 2, 3, and 4.

Chapter 2

Response Stability and Correlations

In this chapter, we first examine the similarity of the multielectrode recordings across different recording sessions, which we refer to as stability. Next, we describe and quantify the various types of correlations between neuron pairs that are present in the data before using them for decoding and firing rate prediction in the next chapters. We quantify correlations in terms of both the signal correlation, the similarity of two sites' stimulus tuning, and noise correlation, the similarity of deviations of average stimulus responses, and also assess how stable the correlations are between different recording sessions.

For results in this and future chapters, recording session data is referred to using the array number (either A1, A2, A3, or A4), and the stimulus set. For example, the string "A2Color1" represents the first recording session for which the Color stimulus set was used with Array 2. For a list of all recording sessions and their corresponding arrays and stimulus sets, see Table 1.1.

2.1 Stability

One concern when using a multi-electrode array for multiple days of recording is whether the responses recorded by a unit remain stable over time. A lack of stability

This is a somewhat simplified approach to quantifying stability in that it treats each neuron independently; correlations between neurons and the stability thereof is the focus of the following section.

2.2 Correlations

Signal correlation Signal correlation is a measure of the correlation between two neurons' average responses to a set of stimuli, and is given by the correlation coefficient between the mean responses to each stimulus:

$$sc(i,j) = \frac{\sum_k (r_{ik} - r_i)(r_{jk} - r_j)}{\sqrt{\sum_k (r_{ik} - r_i)^2} \sqrt{\sum_k (r_{jk} - r_j)^2}} \tag{2.1}$$

where r_{ik} is the mean response of site i to stimulus k, and r_i is the mean response of site i to all stimuli (i.e. $r_i = \frac{1}{numstimuli} \sum_k r_{ik}$).

Figure 2-2 shows a histogram of signal correlations computed for each pair of sites. For all recording sessions, the mean signal correlation is positive. Array 1 has the highest mean signal correlation, and it also has many pairs for which the signal correlation is close to 1. This may be because there are only 8 unique stimuli, and these stimuli - oriented gratings - represent the features to which neurons in that area (V1) are tuned. Arrays 3 and 4 have the next highest mean signal correlations, but the signal correlations for Arrays 2, 3, and 4 have similarly shaped distributions.

We can examine the stability of signal correlations across the different days of recording by computing the correlation coefficient of the signal correlations values of all pairs of neurons on those two days.

Figure 2-4 shows the results, which indicate that while neuron pairs with a high signal correlation on one recording session with a particular stimulus set are likely to have a high signal correlation on another recording session with the same stimulus set, they are not likely to have a high signal correlation with another stimulus set. Signal correlations with the Color stimulus set, in particular, have a very low correlation

1.2.1 Array 1: V1

Recordings from an electrode array with 131 recording sites were taken from V1. The stimuli shown were drifting grating patterns, with 8 possible grating orientations.

1.2.2 Array 2: V4

Data was collected from area V4 of the primate visual cortex. The stimuli were shown as a rapid serial presentation, with each stimulus being shown for 200 ms each and no time in between subsequent stimulus presentations. Multiple stimulus sets were shown, and included a set of colors of varying hue and luminance, a set of objects on a plain background, and natural images.

Noise Rejection A preprocessing step we took on this dataset was to reject detected waveforms that appeared to be noise rather than an actual spike. To do this, we used a straightforward outlier rejection procedure in which any waveform with a value in any dimension that was greater than $p_{75} + 2.5IQR$ or less than $p_{25} - 2.5IQR$, where p_i is the i^{th} percentile and IQR is the interquartile range for the dimension, was classified as noise. However, because the spurious waveforms could influence the computed percentiles and interquartile ranges, we computed those values only using waveforms which conformed to a standard spike shape - with the global minimum occurring before the global maximum and no other local extrema in between. After the percentile and IQR statistics were computed from this subset, all waveforms were taken as possible waveforms when performing the outlier rejection procedure.

1.2.3 Arrays 3 and 4: V4 and IT

For these datasets, a set of natural images was presented in a rapid serial presentation format, with stimuli being presented for 100ms each with a 100ms gap in between, while separate arrays recorded from areas V4 and IT. Three hundred natural images were presented as stimuli while Arrays 3 and 4 were recording; these included the one hundred natural images used as stimuli for Array 2, and two hundred others.

could indicate that the array has drifted and is recording from a different neuron or set of neurons. Assessing stability can help determine whether data from a multi-electrode array can be pooled across days, or whether model parameters learned on one day can be applied to data on another day. This has implications for the use of chronically implanted multi-electrode arrays in brain-machine interfaces, and determining how long we can rely on a trained brain-machine interface to produce valid results.

Multiple metrics have been used to quantify stability. Dickey et al (2009) computed stability scores both based on the correlation between average spike waveforms recorded by a single unit on consecutive days and the similarity of the interspike interval histogram of a unit's recordings on consecutive days. These scores were used to identify a stable subset of neurons that were also more likely to be stable in the future. Jackson and Fetz (2007) also use the correlation between waveforms recorded by the same unit on different days as a measure of stability, but note that different neurons can have similar waveforms, and thus this method may not always pick up on instability that results from an electrode recording from a different neuron.

Since the decoding and encoding models we employ in Chapters 3 and 4 us firing rates within a certain time interval as features, we also compute stability on the level of firing rates. We quantify stability by comparing each recording site's mean firing rates to each stimulus, computing the distance between a site's vector of stimulus means on two different days. To compute the distance between two stimulus mean vector, we use two different distance metrics:

- Euclidean distance, which will be high if the mean responses to each stimulus remained the same on two different days
- Correlation distance (1 minus the correlation coefficient between the two vectors) which would remain high even if the stimulus means didn't stay exactly the same, but were all subjected to the same additive and/or multiplicative noise constants.

Our method takes advantage of the fact that stimulus modulates firing rate in different

ways for different neurons, which can also help avoid the problem that waveform-based methods face of different neurons having similar waveforms.

The plots in Figure 2-1 indicates, for each recording site, the distances between the stimulus response vectors on the two different days, using the two different distance metrics.

Both Arrays 2 and Arrays 4 show many sites with high correlation distance stability. There are also clusters of sites on these arrays with relatively high correlation distance stability, but low Euclidean distance stability. In these cases, it is possible that the location of the recording site remained relatively stable, but that there was a change in the overall means or variances of the neurons' firing rates. Many of Array 3's recording sites, on the other hand, show low stability according to both distance metrics.

Figure 2-1: Stability of mean stimulus responses. We plot stabilities for each pair of subsequent recording sessions with the same stimulus set. To quantify stability, we compute both the Euclidean distance and correlation distance between each recording site's mean firing rates to each stimulus on the two different days. Each point represents a recording site plotted against these two distance metrics.

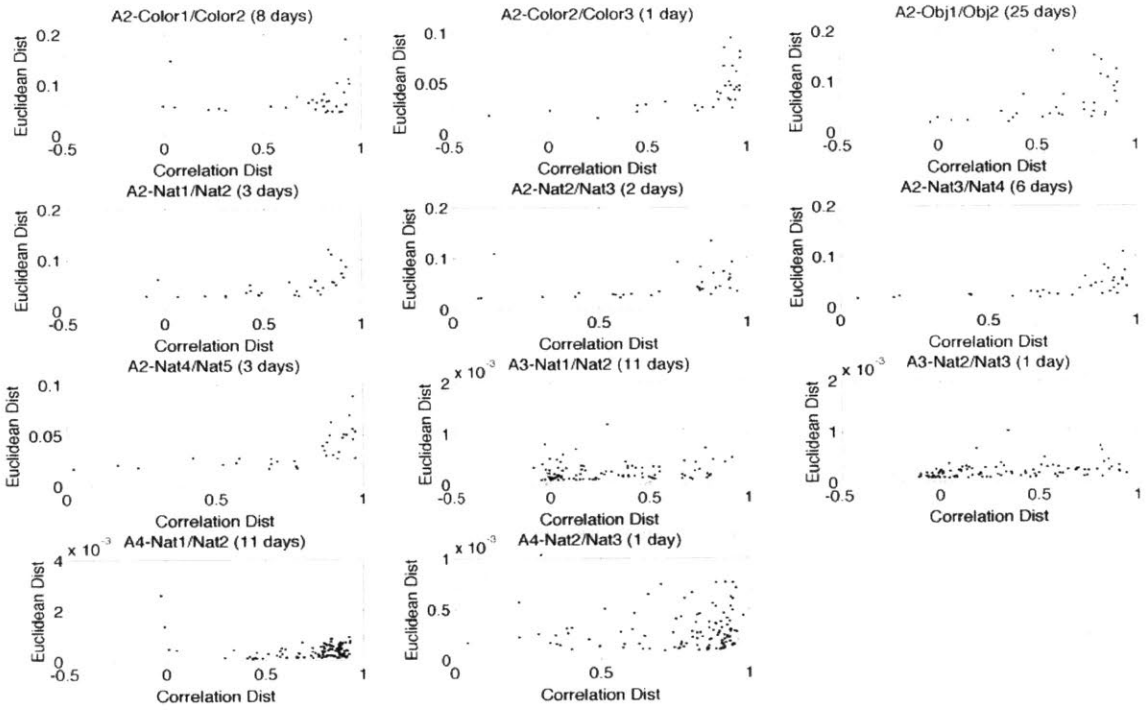
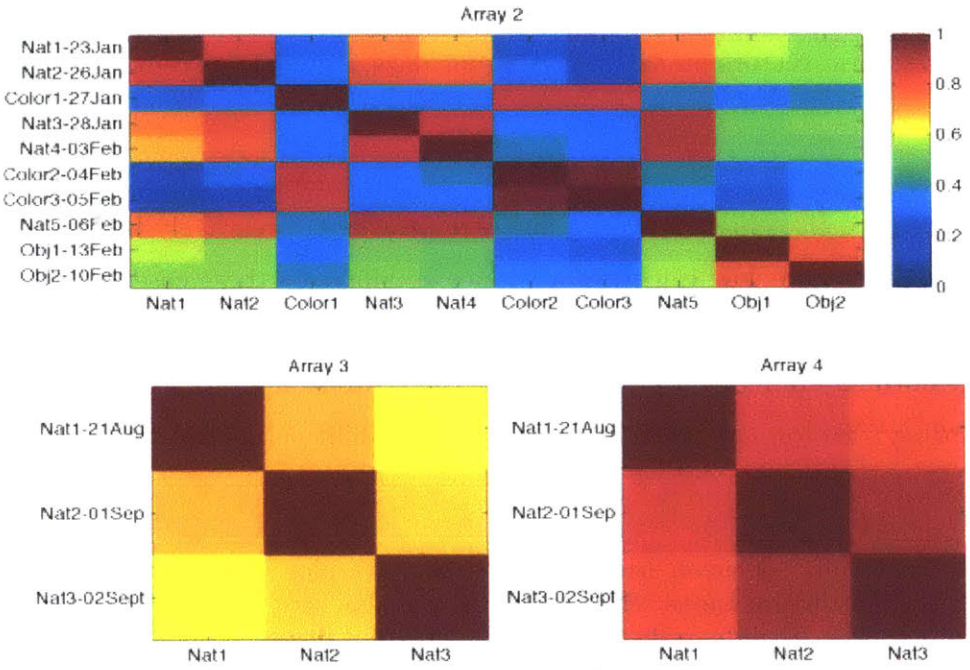


Figure 2-4: Signal correlation similarity across different image sets. The similarity between two days is computed as the correlation coefficient between the sets of signal correlations.



Our data in Figure 2-5 shows positive noise correlations on average, with most recording sessions showing similar distributions except for Array 1, whose units exhibited very low noise correlations. The mean noise correlations for the other arrays generally agree with the noise correlation values of .1-.2 found in the literature (Cohen and Kohn, 2011)

We can look at the stability of noise correlations in an analogous manner to signal correlations. We create a vector of noise correlations of all neuron pairs for two recording sessions, and then compute the correlation coefficient between these two vectors. From Figure 2-7, we can see that, for Array 2, the noise correlation stability is high, even for different stimulus sets. There does appear to be a mild effect of stimulus set, however. As one example, recording Nat4 and Nat5 have a higher noise correlation stability that either Nat4 and Color2 or Nat4 and Color3, even though the Color2 and Color3 recording sessions occurred in between Nat4 and Nat5. For Arrays 3 and 4, on the other hand, noise correlation stability is relatively low.

Figure 2-2: Signal correlations

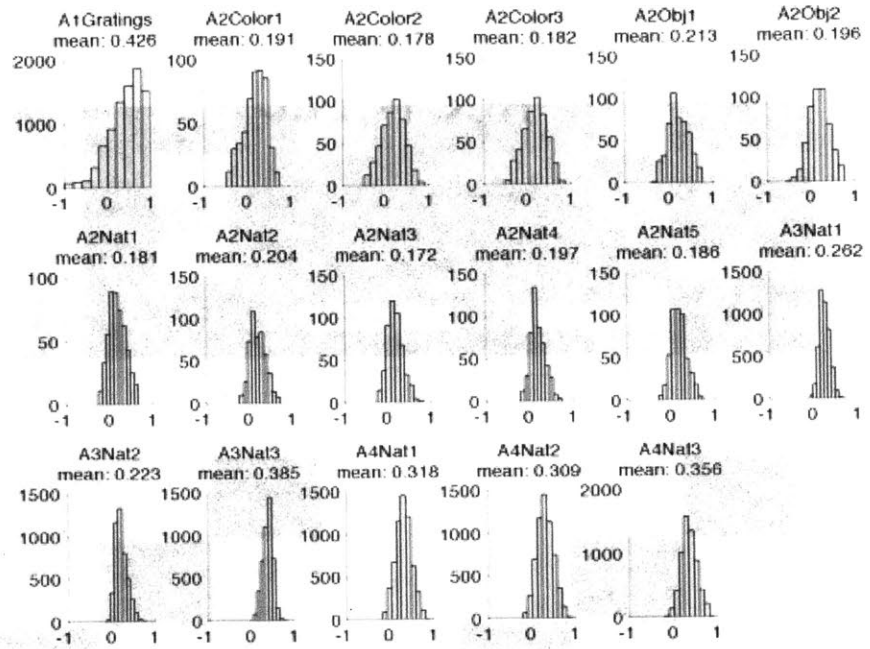


Figure 2-3: Distribution of signal correlations for each recording session

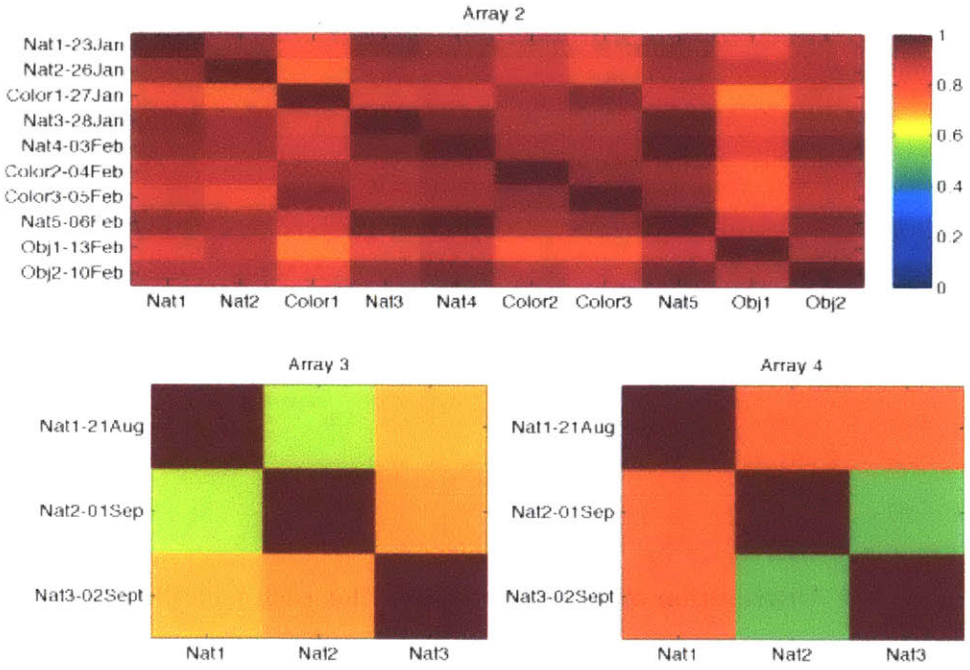
with signal correlations in either the Object or Natural Image stimulus sets. However, signal correlations with the Natural Image set have a moderate correlation with signal correlations in the Object set. For Arrays 3, the signal correlation similarity between different days with the same stimulus set is relatively low compared to the other arrays; however, this is not as surprising given the low response stability demonstrated in Figure 2-1.

Noise correlation Noise correlation describes the relationship between two neurons' deviations from their average stimulus responses.

$$nc(i,j) = \frac{\sum_k \sum_m (r_{ikm} - r_{ik})(r_{jkm} - r_{jk})}{\sqrt{\sum_k \sum_m (r_{ikm} - r_{ik})^2} \sqrt{\sum_k \sum_m (r_{jkm} - r_{jk})^2}} \tag{2.2}$$

Here, r_{ikm} is the response of neuron i to stimulus k on its mth presentation, and r_{ik} is the mean response of neuron i to stimulus k.

Figure 2-7: Noise correlation similarity across different image sets. The similarity between two days is computed as the correlation coefficient between the sets of noise correlations.



decoding and prediction of the neural code (e.g. Maynard et al (2010)). To analyze the extent to which correlations are dependent on stimulus, we can compare stimulus-specific noise correlations on different days, where the stimulus-specific noise correlation to stimulus k is given by the correlation coefficient between all responses to that stimulus:

$$ssnc_k(i,j) = \frac{\sum_m (r_{ikm} - r_{ik})(r_{jkm} - r_{jk})}{\sqrt{\sum_m (r_{ikm} - r_{ik})^2} \sqrt{\sum_m (r_{jkm} - r_{jk})^2}} \tag{2.3}$$

We can then examine whether there are any patterns in a neuron pair's stimulus-specific noise correlations that are stable across days by creating, for each pair of neurons, a vector of all stimulus-specific noise correlations for two different recording days, and then correlating these two vectors. We then obtain a set of stimulus specific noise correlation stabilities.

Figure 2-5: Noise Correlations

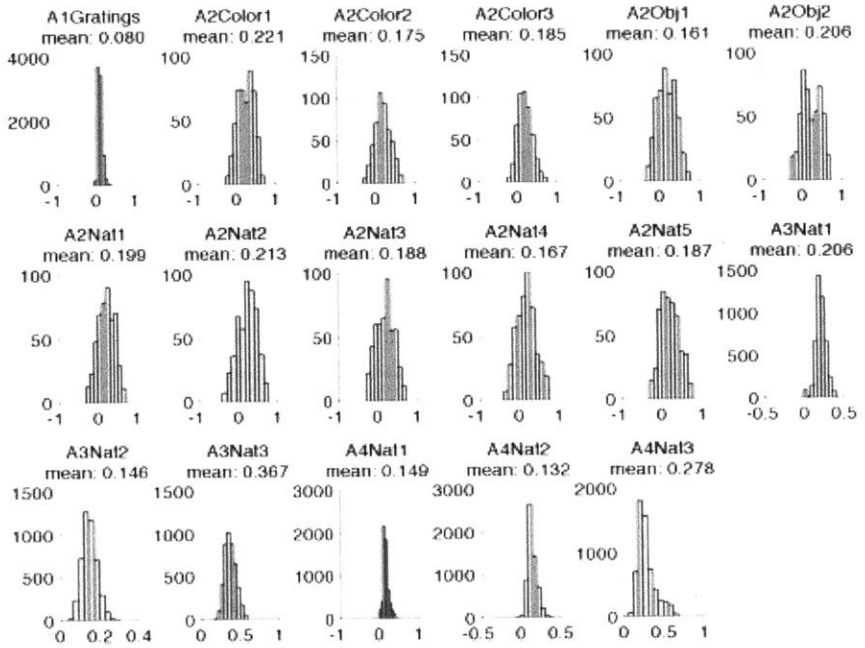


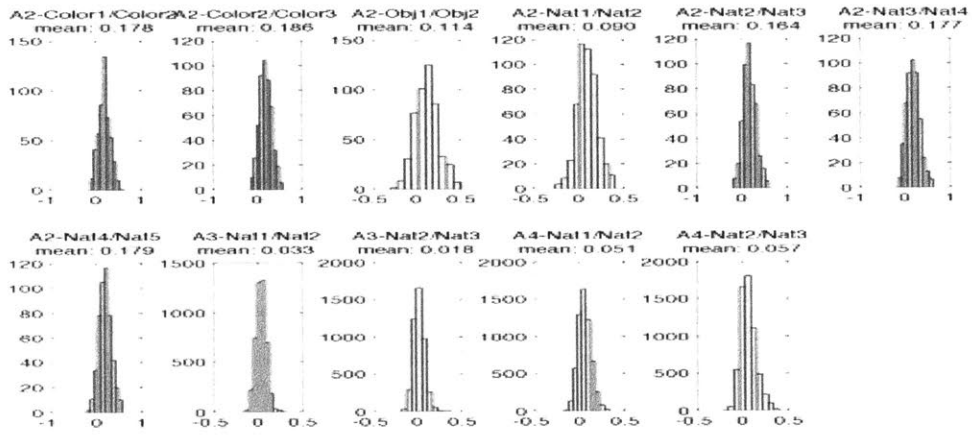
Figure 2-6: Distribution of noise correlations for each recording session

Relationship between signal and noise correlation A positive relationship between signal correlation and noise correlation has been well documented in the literature (Gu et al, 2011; Cohen and Maunsell; 2009). We find a similar trend in our datasets. Plots of signal correlation vs. noise correlation and their corresponding line-of-best-fit slopes are shown in Figure 2-8. Again, the exception is Array 1, which had low noise correlations and also a very low slope. Among the recording session with Array 2, the Color stimulus sets tend to have a lower slope than either the Object or Natural Image sets.

A positive relationship between signal and noise correlation is thought to have a deleterious effect on population coding (Averbeck et al, 2006); this will be discussed more in Chapter 3.

Stimulus-modulated correlations There is some evidence that noise correlations might in fact depend on the stimulus - thus making them stimulus-modulated correlations. If correlations are stimulus-modulated, it would have implications for both

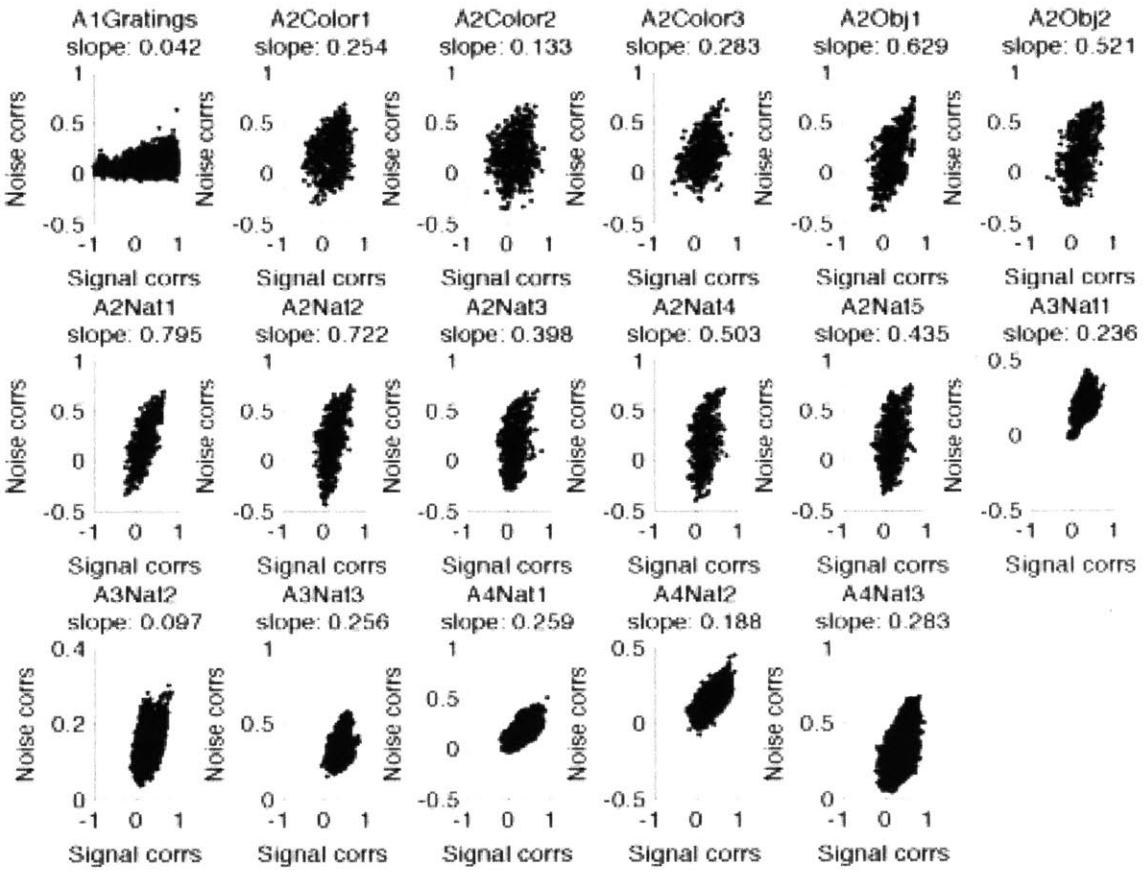
Figure 2-9: Correlation of Stimulus-Specific Noise Correlations. For each subsequent pair of recording sessions with the same stimulus set, a histogram of all sites' correlations between stimulus-specific noise correlations on the two different days is shown. All pairs of days show a higher average correlation value than in a trial-shuffled population.



these distributions are all greater than 0 ($p < .05$) indicating that there may be some stimulus-modulated correlations. The means are also all significantly greater ($p < .05$) than the means obtained when performing the same analysis on a pseudopopulation.

Another way of assessing the dependence of correlations on stimulus is by examining the second-order interaction terms - firing rates from two sites multiplied together. Because the information about the stimulus in the firing rates of single neurons might contribute to the eta squared value of the interaction terms, we compute interaction terms by multiplying together the z-score of the two sites' firing rates (after subtracting out the stimulus mean and dividing by the stimulus standard deviation). We can compare the average eta squared values of the interaction terms to the values when we use a pseudopopulation. Eta squared is a measure of effect size - it measures the proportion of variance in the interaction terms that is explained by the stimulus. We find that the interaction terms show a small effect of stimulus according to Cohen's criteria (where .01 represents a small effect); furthermore, the eta squared values are significantly greater without shuffling that with shuffling ($p < .05$), indicating that the interaction terms do carry information about the stimulus (Figure 2-10).

Figure 2-8: Relationship between signal and noise correlations. For each recording session, the slope of the line of best fit is indicated.



If correlations are not stimulus-modulated, we would expect the stimulus-specific noise correlations to be randomly distributed around the standard (non-stimulus-specific) noise correlation for a given pair of neurons, so there should be no relationship between the order of stimulus-specific noise correlations across different days. If there is a positive correlation between the stimulus-specific noise correlations, it would indicate that the stimulus modulates the noise correlation between a pair of neurons in predictable ways that possible encode information.

For each pair of sites, we compute the correlation of all stimulus-specific noise correlations between two subsequent days of recording the same stimulus set. Figure 2-9 shows a histogram of these correlations for all pairs of neurons. The means of

everywhere else:

$$C^{reg} = \lambda C + (1 - \lambda)S \tag{3.2}$$

where the regularization parameter λ ranges from 0 to 1 (0 meaning no regularization, and 1 being equivalent to the diagonal covariance classifier).

To regularize the variable covariance classifier, we interpolate the class-specific covariance matrix C_k with the overall full covariance matrix C.

$$C_k^{reg} = \lambda C_k + (1 - \lambda)C \tag{3.3}$$

Because regularization introduces another free parameter, λ, to the classifier, we must ensure that λ is estimated from data separate from the data on which it is tested. We do this by dividing the training set itself up into cross-validation folds; the regularization parameter that maximizes the accuracy on cross-validation within the training set is then used when evaluating the test set.

Global noise model classifiers These classifiers also compute $v_{template}$ for each stimulus from the training set. However, instead of computing a covariance matrix to model correlations between pairs of neurons, they assume that all neurons in the population are subject to the same noise constant(s).

Max correlation coefficient (MCC)

Maximizes the Pearson product-moment correlation coefficient between v_{test} and $v_{template}^k$. Because the correlation coefficient between v_{test} and $v_{template}$ is equivalent to the correlation coefficient between $av_{test} + b$ and $v_{template}^k$, this classifier assumes linear noise, and is invariant to changes in location and scale.

Max uncentered correlation coefficient (MUCC)

Maximizes the correlation coefficient between v_{test} and $v_{template}^k$ without subtracting off mean. This is equivalent to the cosine of the angle between the two vectors. Because the angle between v_{test} and $v_{template}^k$ is equivalent to the angle between av_{test} and $v_{template}^k$, this classifier assumes multiplicative noise, and is

Chapter 3

Noise Correlations and Population Decoding

The amount of information carried by noise correlations, and the information lost by ignoring them, is the topic of this chapter. Our study is an empirical one, in which information is quantified by decoding accuracy. As outlined by Averbeck et al (2006), the effect of correlations on the decoding accuracy can be examined from two different perspectives. The first asks whether a population with no correlations - that is, not simultaneously recorded - has more or less information than a simultaneously recorded population. This helps address the validity of decoding analyses done on non-simultaneously recorded populations. The second perspective looks at how much information is lost when using a classifier that ignores correlations, and has more to do with potential decoding strategies that can used by downstream neurons in the brain to read out information from neural populations. For example, if a classifier that ignores correlations loses little accuracy, it is plausible that the brain's readout mechanism also does not have to take correlations into account.

3.1 Population Decoding

In neural population decoding, we use the responses of a population of neurons to predict which stimulus was being presented at a given time. We define a neural

response to be the firing rate within a time interval. We concatenate the firing rates of all sites into a feature vector, with a length equal to the number of recording sites. The feature vectors and stimulus labels are used to train a classifier, which can then be used to predict the stimulus of a new feature vector.

To assess accuracy, we run cross validation - dividing the data into folds, then repeatedly holding one fold out while training a classifier on the remaining folds and testing the classification accuracy on the held-out fold. The number of folds is typically set equal to the number of presentations of each stimulus, and each fold contains one repetition of each stimulus. The entire cross validation procedure is repeated with different fold partitions 2-10 times, depending on the dataset, and results are averaged together to ensure the accuracies converge.

To run decoding, we used the Neural Decoding Toolbox (Meyers, 2013).

3.1.1 Classifiers

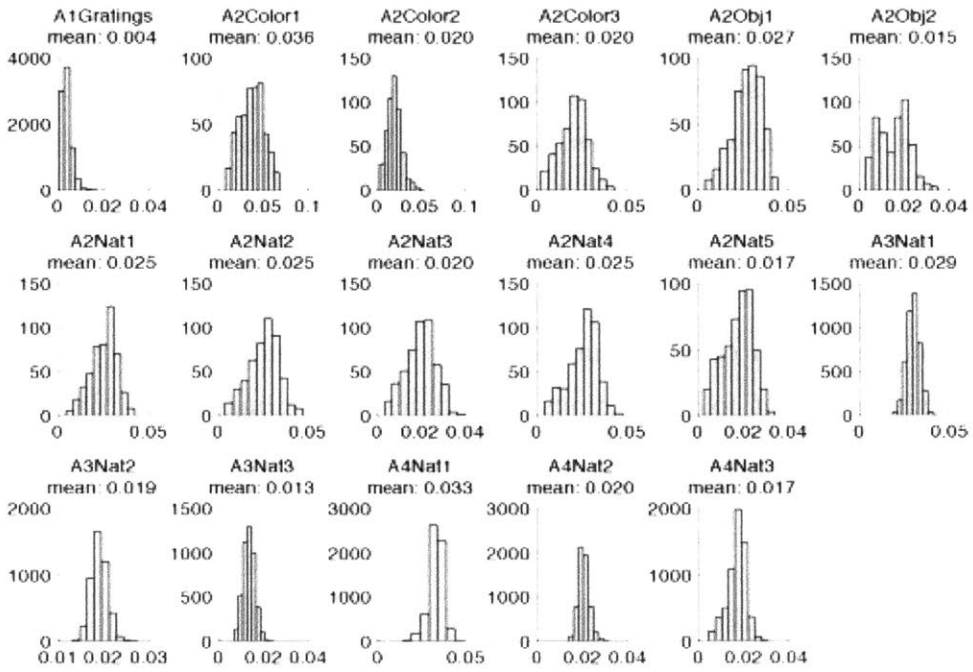
In each of our classifiers, the training set responses to a given stimulus k are summarized by a vector containing the means of each site's responses to that stimulus, which we refer to as the template vector $v^k_{template}$. For some classifiers, the variance in each dimension or covariances between all dimensions are also used. To classify a new feature vector v_{test}, we compute the distance between it and each of the template vectors. The predicted label is the stimulus whose template is closest according to the given distance metric.

Gaussian maximum-likelihood classifiers These classifiers assume the neural responses for a particular stimulus are given by a multivariate Gaussian distribution. The distance between $v^k_{template}$ and v_{test} is then quantified by the equation

$$(v_{test} - v^k_{template})^T C^{-1} (v_{test} - v^k_{template}) \tag{3.1}$$

which appears in the exponent of the equation for the Gaussian probability density function. By altering the nature of the covariance matrix C, different distance metrics,

Figure 2-10: Eta-Squared values of interaction terms. Each histogram shows the distribution of eta-squared values for all neuron pairs' interaction terms.



2.3 Conclusion

Our finding on signal and noise correlations match the general trends found in the literature of positive correlations on average, and a positive relationship between signal and noise correlation. We also investigate the stability of recorded responses and correlations across different days of recording. The finding of dependence of signal correlation stability on stimulus set is a new one; explaining this phenomenon, possibly in terms of the tuning curves of neurons, is left for future work. We find some evidence for noise correlations that are modulated by stimulus; they will be investigated further in the context of population decoding in the following chapter.

and thus different classifiers, are obtained.

Full covariance classifier

Minimizes the Mahalanobis distance between v_{test} and $v^k_{template}$. Uses a full covariance matrix, which is the same for each stimulus, for C. The full covariance matrix is the average of the stimulus-specific covariance matrices, and therefore only reflects noise correlations.

Diagonal covariance classifier

Minimizes standardized Euclidean distance between v_{test} and $v^k_{template}$ (Euclidean distance with each dimension normalized by its standard deviation). This is equivalent to computing a diagonal covariance matrix with the average stimulus-specific variance of each dimension as the diagonal elements, which is the same for each stimulus (or, alternatively, computing the full covariance matrix as described above and then setting all non-diagonal elements to zero).

Variable covariance classifier

Minimizes the Mahalanobis distance between v_{test} and $v^k_{template}$. Uses a different full covariance matrix C_k for each stimulus, which just takes into account the noise correlations specific to that stimulus. This classifier is used to understand whether stimulus-modulated correlations are helpful for population decoding.

Regularization While previous work on population decoding with Gaussian maximum likelihood classifiers typically employs cross-validation to ensure that results are not due to overfitting the training set, no studies to our knowledge employ regularization in these classifiers to actually help prevent overfitting. Classifiers with many parameters are most susceptible to overfitting - modeling noise rather than meaningful information in the training set in a way that does not generalize to the test set - so we use regularization on the full covariance and variable covariance classifiers.

To regularize the full covariance classifier, we do not use the raw covariance matrix estimate C in the full covariance classifier, but instead interpolate it with a simpler covariance matrix - a diagonal matrix S with variances on the diagonal and zeros

1. A - the decoding accuracy using a full covariance matrix decoder on simultaneous data

2. A_{diag} - the decoding accuracy using a diagonal covariance matrix decoder on simultaneous data

An alternative way to compute ΔA_{diag} is to compute A the same way, but to compute A_{diag} as the decoding accuracy using a full covariance matrix decoder that is trained on shuffled data, but evaluated on the original unshuffled data (Averbeck et al, 2006). This is because the full covariance classifier, when trained on shuffled data, should end up with a diagonal covariance matrix because of the lack of noise correlations in the data. We will refer to this as Method 2 for computing A_{diag}, and the diagonal classifier as Method 1.

Theoretically, ΔA_{diag} should always be positive. The full covariance classifier has the same parameters as the diagonal covariance classifier does for modeling the data and many more; therefore, it should not do worse than the diagonal covariance classifier. In practice, however, we may see negative values of ΔA_{diag} when performing cross validation or testing accuracy on data that was not used to train the classifier. A negative value would indicate that the full covariance classifier is overfitting its training set, modeling noise that does not generalize to the test set.

3.1.4 $\Delta A_{shuffled}$ and ΔA_{diag}

$A_{shuffled}$ and A_{diag} are computed in strikingly similar ways. Since using a full covariance classifier on shuffled training data is theoretically equivalent to using a diagonal covariance classifier, the difference between $A_{shuffled}$ and A_{diag} lies only in the testing data. The classifier for computing $A_{shuffled}$ is tested on shuffled data; the classifier for A_{diag} is tested on unshuffled data.

Despite these similarities, Averbeck et al (2006) show that depending on the data, there may not be a strong relationship between the two values. A particular dataset may have a high value of A_{diag}, but a negative value of $A_{shuffled}$. This is also explored empirically by Averbeck and Lee (2005), who, when performing decoding with small

On the other hand, other studies show that positive noise correlations result in negative values of $\Delta I_{shuffled}$ only when the stimulus correlations are positive, but positive values of $\Delta I_{shuffled}$ when stimulus correlations are negative. Poort and Roesfelma (2009) show that these opposing effects essentially cancel each other out as population size increases, resulting in an overall negligible effect of noise correlations. Similarly, Gu et al (2012) found that training animals to do a task decreased noise correlations, but that since the change in noise correlation was not related to the signal correlation, these changes did not significantly affect decoding accuracy. Averbeck and Lee (2006) found that populations of size 3-8 show larger effects of correlations than pairs of neurons, with $\Delta A_{shuffled}$ values ranging from roughly -2% to 2.5% on populations where decoding accuracies typically ranged from 70-100%. However, the mean of the $\Delta A_{shuffled}$ values was close to zero.

Finally, a theoretical study of Fisher Information by Abbott and Dayan (1999) found that for some noise models, positive correlations actually increase the information in the population. Noise models examined in this study included additive, multiplicative, and a limited-range model in which noise correlation is higher for pairs of neurons with more similar tuning curves. For the additive and multiplicative models, correlations increase information as long as the neurons are diversely tuned, meaning that they do not all respond most strongly to the same few stimuli. However, for limited range correlations, higher noise correlations tend to decrease information. Since neurons with more similar tuning curves have a higher signal correlation, this agrees with the finding that positive noise correlations are harmful for information coding when there are also positive signal correlations.

Previous work on ΔI_{diag} tends to show that this value is small and often insignificant for pairs of neurons, but can have a significant effect at larger population sizes. For example, Nirenberg et al (2001) claimed, based on computations of ΔI_{diag} with neuron pairs, that retinal ganglion cells are largely independent encoders. However, Averbeck and Lee (2006) and Stevenson et al (2010) both demonstrated empirically that the values of ΔI_{diag} increase with population size. Stevenson (2010) studied the effect of correlations by using Gibbs sampling to estimate the joint probabil-

invariant to changes in scale, but not location.

Min parallel distance (MPD)

Minimizes the distance between the parallel lines $x(t) = v^k_{template} + t$ and $y(t) = v_{test} + t$. This classifier explicitly assumes additive noise; it is invariant to changes in location, but not scale.

3.1.2 Decoding with pseudopopulations: $\Delta A_{shuffled}$

We can use neural population decoding as a tool to discover how much information is encoded by noise correlations. Specifically, we can compare the decoding accuracies when using a dataset of simultaneously recorded neurons to a dataset of non-simultaneously recorded neurons. In practice, to control for the neurons and stimulus set, we can artificially create a population of non-simultaneously recorded neurons by performing the following shuffling procedure: for all trials on which a given stimulus was presented, randomly and independently shuffle each neuron's responses on those trials, and repeat for each stimulus. This so-called pseudopopulation theoretically has no noise correlations.

If we define A to be the decoding accuracy with the simultaneous population and $A_{shuffled}$ to be the decoding accuracy with the shuffled population, then the difference $A - A_{shuffled}$, or $\Delta A_{shuffled}$, is representative of the amount of information that noise correlations encode. $\Delta A_{shuffled}$ can either be negative or positive (Averbeck et al, 2006). Note also that when computing A and $A_{shuffled}$, a classifier that actually takes correlations into account (e.g., the full covariance decoder) should be used.

3.1.3 Decoding with a suboptimal classifier: ΔA_{diag}

A complementary approach to studying correlations in neural population decoding is to consider, on population data the loss in decoding accuracy when a suboptimal decoder (i.e., one that ignores any correlations between neurons) is used. This difference is termed ΔA_{diag}, named as such because it can be thought of as the difference between:

provement in absolute decoding accuracy for many of the recording sessions. The Array 3 data's optimal λ value, however, shows an average 12.5% improvement over the standard full covariance classifier. Moreover, the optimal value of λ is relatively stable across days. Thus for some datasets its seems that regularization can allow us to gain benefits from correlation information without overfitting the training set, potentially preventing theoretically unsound results such as a negative value of ΔA_{diag}.

Based on the results, we find that even with regularization, estimating a different covariance matrix for each stimulus set does not improve decoding accuracy, despite some evidence suggesting the existence of stimulus-modulated correlations we saw in Chapter 2. This result is different from that of Maynard et al (2010), who found that when decoding arm movement direction from the primary motor cortex, the variable covariance model improved 11 percent over the model that didn't take into account correlations (diagonal covariance), compared to a 5 percent improvement for a regular full covariance classifier. However, we have relatively few repetitions of each stimulus with which to estimate a stimulus-specific covariance matrix; it is possible that with more stimulus repetitions, the results reported here may change.

$\Delta A_{shuffled}$ In Figure 3-2, for each day of recording, we plot A and $A_{shuffled}$ when evaluated using cross validation. We find that for all arrays recording from V4, $A_{shuffled}$ is less than A. This means that those results agree with the theoretical findings of Abbott and Dayan (2005) claiming that noise correlations can be beneficial for some noise models. However, with Array 4, which recorded from IT, there are mixed results for the sign of $\Delta A_{shuffled}$.

We also include results with the regularized versions of the full covariance classifiers. However, unlike in Figure 3-1, we now use cross validation within the training set to find the optimal value of λ to be applied to the test set. Though there are generally slight increases in both A and $A_{shuffled}$ when using regularization, $\Delta A_{shuffled}$ remains approximately the same.

populations of neurons, point out groups with seemingly unrelated $A_{shuffled}$ and A_{diag} values.

The assumption that shuffled populations exhibit no noise correlations is important for the computation of $A_{shuffled}$ and A_{diag} (Method 2). However, given a limited amount of stimulus presentations, this assumption may not be valid. To address this question, we compare both methods for computing A_{diag} as well as the results when using regularization, which should only use correlations to the extent that they are useful.

3.2 Related Work

While some previous studies also examine the effect of correlations on decoding accuracy, many use information-theoretic measures to compute measures analogous to A, $A_{shuffled}$, and A_{diag}, which we will refer to in general as I, $I_{shuffled}$, and I_{diag}. In cases where only two different stimuli were used, measures such as *d-prime*, a measure of the separability of Gaussian distributions, have been used to compute ΔI_{diag} and $\Delta I_{shuffled}$. However, as shown by Averbeck and Lee (2006), these values are strongly predictive of the values computed when using classification accuracy to measure information.

There have been mixed results on $\Delta I_{shuffled}$ in previous work. Averbeck, Latham, and Pouget (2006) approach the study of noise correlations from a theoretical perspective, and suggest that positive values of correlations will result in negative values of $\Delta I_{shuffled}$. They show how even small effects of correlations can get magnified as the population size grows. Cohen and Maunsell (2009) also found that attention reduces noise correlations, and this noise correlation reduction is primarily responsible for better discriminability in neural populations in the attended condition. Stevenson et al (2010) study a variety of datasets and perform decoding with a simulated population of independent neurons with the same tuning properties, finding that it results in a 25% improvement over the actual data with a decoder that takes correlations into account.

ity distribution, instead of using a Gaussian classifier, and compare it to decoding when treating neurons as independent encoders. Compared to the decoder that treats neurons as independent, the one that takes correlations into account results in improvements between 4.8% and 10.3% at population sizes of 65-100 neurons. Another empirical study made at larger population sizes (12-16 neurons) was done by Maynard et al (1999), who found that taking interactions into account improved accuracy of decoding of movement direction in the primary motor cortex by about 5%. They also found that some of the correlations were stimulus-modulated, so using a variable covariance classifier provided additional significant improvements over the standard full covariance classifier.

3.3 Results

In this section, we show the results of decoding analyses for the computation of A, $A_{shuffled}$, and A_{diag}. We also perform other analyses including examine the effect of population size on these values. For Array 1, to which only 8 stimuli were displayed, decoding results are near 100% with the full population; thus, we generally only show results as a function of population size for this dataset.

As in other chapters, the different recording sessions are denoted by both the array number and stimulus set, as presented in Table 1.1. For example, the recording session "A2Color1" represents the first recording session using the Color stimulus set with Array 2.

Regularization Figure 3-1 shows the results of incorporating regularization into the full covariance and diagonal covariance classifiers on simultaneous data. Decoding accuracies when using the regularized form of the covariance matrix are shown for varying values of λ For some values of λ regularized full covariance classifier, this decoding accuracy is better than either the full or diagonal covariance classifiers. However, the improvement is small on average - only about 4% relative improvement over the standard full covariance classifier. This amounts to less than a percent im-

Effect of Population Size As with $\Delta A_{shuffled}$, we also look at the effect of population size on ΔA_{diag}. For recording sessions that have larger values of ΔA_{diag} (in particular, the recording sessions with Array 2), the values of ΔA_{diag} at very small population sizes are often still fairly close to zero, consistent with others who studied the use of suboptimal decoders using small population sizes and found no significant effect (e.g. Averbeck and Lee, 2006; Nirenberg et al, 2001).

In the Array 1 data, both A and A_{diag} level off as the population size increases, A levels off at a near perfect decoding accuracy (approximately 99.9%), while A_{diag} levels off at only 99%. Though this is still a good decoding accuracy, it is interesting that it doesn't increase even as the population size does; it seems that in order to gain the final one percent of decoding accuracy on this dataset, correlations need to be taken into account.

We can also note that the phenomenon found with A_{diag} in Array 3 and Array 4, where shuffling the training data does not produce the same results as using a diagonal covariance matrix for computing A_{diag}, is less pronounced at smaller population sizes. This supports the idea that the ratio between the number of presentations of each stimulus and the number of recording sites might play a role.

Relationship between $\Delta A_{shuffled}$, ΔA_{diag}, and Signal and Noise Correlations Though $\Delta A_{shuffled}$ and ΔA_{diag} are computed in similar ways, they need not be related. In Figure 3-6, we plot the values of $\Delta A_{shuffled}$ against ΔA_{diag}, with both values normalized by A. In the plot, different colors and point shapes represent different arrays and stimulus sets. Though the only difference between $A_{shuffled}$ and A_{diag} is the testing data (it is shuffled for $A_{shuffled}$ and unshuffled for A_{diag}), there is only a weak relationship between $\Delta A_{shuffled}/A$ and $\Delta A_{diag}/A$. With Arrays 3 and 4, where $\Delta A_{diag}/A$ is close to zero over a wide range of values of $\Delta A_{shuffled}/A$, including both positive and negative values. For most points, $\Delta A_{shuffled}/A$ is greater than $\Delta A_{diag}/A$. Since A is the same for a given datapoint's values of $\Delta A_{shuffled}/A$ and $\Delta A_{diag}/A$, this is equivalent to saying that for most points $A_{shuffled}$ is less than A_{diag}.

Additionally, some have claimed possible relationships between $\Delta A_{shuffled}$ and

Effect of Population Size We also consider the effect of population size on $\Delta A_{shuffled}$, to address the differing claims found in the literature. While some claim that small values of ΔI in small populations can get magnified as the population size grows (Averbeck et al, 2006), others have claimed that differing values of ΔI essentially get canceled out when adding more neurons to the population (Poort and Roesfelma, 2009).

For many of the recording sessions, $\Delta A_{shuffled}$ is near 0 at small population sizes, which agrees with the small effects of correlations found by studies which only looked at smaller populations (e.g. Averbeck and Lee, 2006). In the Array 1 data, which we did not look at in Figure 3-2, the values of $\Delta A_{shuffled}$ remain fairly small at all population sizes, with a possible decrease in $\Delta A_{shuffled}$ at a population size of around 10 before both A and $A_{shuffled}$ converged to nearly perfect decoding accuracy.

ΔA_{diag} In Figure 3-4, we compare the use classifiers that ignore correlations to classifiers that take them into account. When comparing the full covariance and diagonal covariance classifiers, Array 2's data shows an average relative loss of 20% in decoding accuracy. For Arrays 3 and 4 on the other hand, the relative differences are less than 2%, though this increases if A, reg is used in place of A. In particular, the relative losses for Array 3 when comparing A_{diag} to A, reg range from 8-15%, for Array 4 they are between 1-4%, and for Array 2 they increase only slightly to 20.8%.

We also examine both methods of computing ΔA_{diag}, and find that they can lead to significantly different results, in particular in Arrays 3 and 4. For both these arrays, the number of repetitions of each stimulus is significantly less than the number of neurons, meaning that the artificial trial-shuffled pseudopopulation will have trials in which multiple firing rates may have come from the same trial in the original simultaneous population. These residual correlations that are present in the pseudopopulation might have affected decoding results when the classifier is trained on the pseudopopulation. In any case, using a diagonal covariance classifier to compute A_{diag} appears to be a more consistent option.

rately classify data from a different recording session. For each stimulus set, we train a classifier on data from the each recording session with that stimulus set, and evaluate the accuracy on data from the other recording sessions. The results are shown in Figure 3-8. The accuracies stay fairly stable over many days, even though we might expect that a classifier which uses many parameters such as the full covariance classifier would have a greater risk of overfitting one day's data. The stability provides more evidence that the structure of noise correlations stay relatively constant across days and weeks of recording.

Alternative classifiers While comparing the full and diagonal covariance classifiers can illustrate the effect of a decoder that ignores correlations altogether, we can ask similar questions about other classifiers which are often used for population decoding and which assume a particular global model of correlated noise - specifically, the MPD classifier, which assumes global additive noise, the MUCC classifier, which assumes global multiplicative noise, and the MCC classifier, which assumes global linear noise. The use of such classifiers is validated by the average positive correlations seen between neurons. By assuming a simpler noise model, they perform an implicit sort of regularization that may provide accuracy improvements without requiring a search over values of regularization parameter.

In effect, when using these classifiers, we are replacing A_{diag} with $A_{noiseModel}$, where $noiseModel$ is the particular type of noise assumed by the classifier. $\Delta A_{noiseModel}$ is then the amount of information that is gained or lost by assuming this noise model when decoding. Because the MCC, MUCC, and MPD classifiers are not a strict subset of the full covariance classifier (as is the diagonal covariance classifier), $\Delta A_{noiseModel}$ is not constrained to be positive.

The results in Figure 3-9 show that with Array 2, the full covariance matrix usually significantly outperforms any of the global noise model classifiers. Of the global noise model classifiers, the MCC, which assumes linear noise, usually performs best. The data in Array 4 shows similar results, although the difference between the full covariance classifier and the alternative classifiers is not as drastic. Furthermore,

the magnitude of noise correlations or the slope of the line that best fits the relationship between signal and noise correlations (with signal correlations on the x axis). Here, we examine the correlation between these different metrics. For this analysis, instead of using the raw values for $\Delta A_{shuffled}$ and ΔA_{diag}, we normalize them by dividing both by A. Still, comparing values on disparate values and stimulus sets may produce misleading results. Therefore, instead of computing the correlations for all datasets together, we compute them for all datasets with a given array/stimulus set combination, and then average the correlations for all array/stimulus set combinations to obtain the final correlation value. The correlations for specific array/stimulus set combinations are computed with Spearman's rank correlation coefficient, which assesses how well the relationship between two variables can be fit with a monotonic function. By using Spearman's rank correlation coefficient instead of the standard Pearson's r that measure linear correlation, our analysis only depends on the relative values of these different metrics, and not their absolute magnitudes.

Using this method, we find that there are moderate to strong positive relationships between $\Delta A_{shuffled}/A$, the mean signal correlation, the mean noise correlation, and the signal-noise slope (all correlations greater than .45). The strongest relationship is between the signal-noise slope and $\Delta A_{shuffled}/A$, which is shown in Figure 3-7. Qualitatively speaking from these results, only the A2 Color datasets show a non-monotonic relationship between signal-noise slope and $\Delta A_{shuffled}/A$. Other theoretical findings suggest that a higher signal-noise slope should cause the correlations to be more harmful - that is, $\Delta A_{shuffled}/A$ should decrease (Gu et al, 2011; Averbeck et al, 2006); our results here appear to contradict this.

On the other hand, $\Delta A_{diag}/A$ has only weak correlations with the four other metrics, and the signs of these correlations are mixed (all absolute values of correlations are less than .25).

Stability of classifiers In Chapter 2, we examined the stability of responses, signal correlations, and noise correlations across days of recording. We can extend that analysis by looking at whether a classifier learned on one day's data can use to accu-

which does not use any correlations, the classifier that computes A should be able to ignore correlations to an extent in order to prevent overfitting.

We also compared the Gaussian maximum likelihood classifiers to simpler ones based on global noise models. These have the advantage of still allowing for correlations, but not requiring a full covariance matrix. Computing the inverse of the full covariance matrix, as required by the distance equation, is a costly operation; in the regularized version, the search for an optimal regularization parameter also significantly increases the run time of this classifier. From this practical perspective, using one of the global noise model classifiers might provide the benefits of correlations without the computational overhead of the (possibly regularized) full covariance classifier.

Figure 3-1: Decoding accuracy of regularized classifiers as a function of regularization parameter. For the regularized full covariance classifier, a parameter value of 0 is equivalent to the standard, unregularized full covariance classifier, 1 to the diagonal covariance classifier. For the regularized variable covariance classifier, a parameter value of 0 is equal to the unregularized variable covariance classifier, 1 to the full covariance classifier.

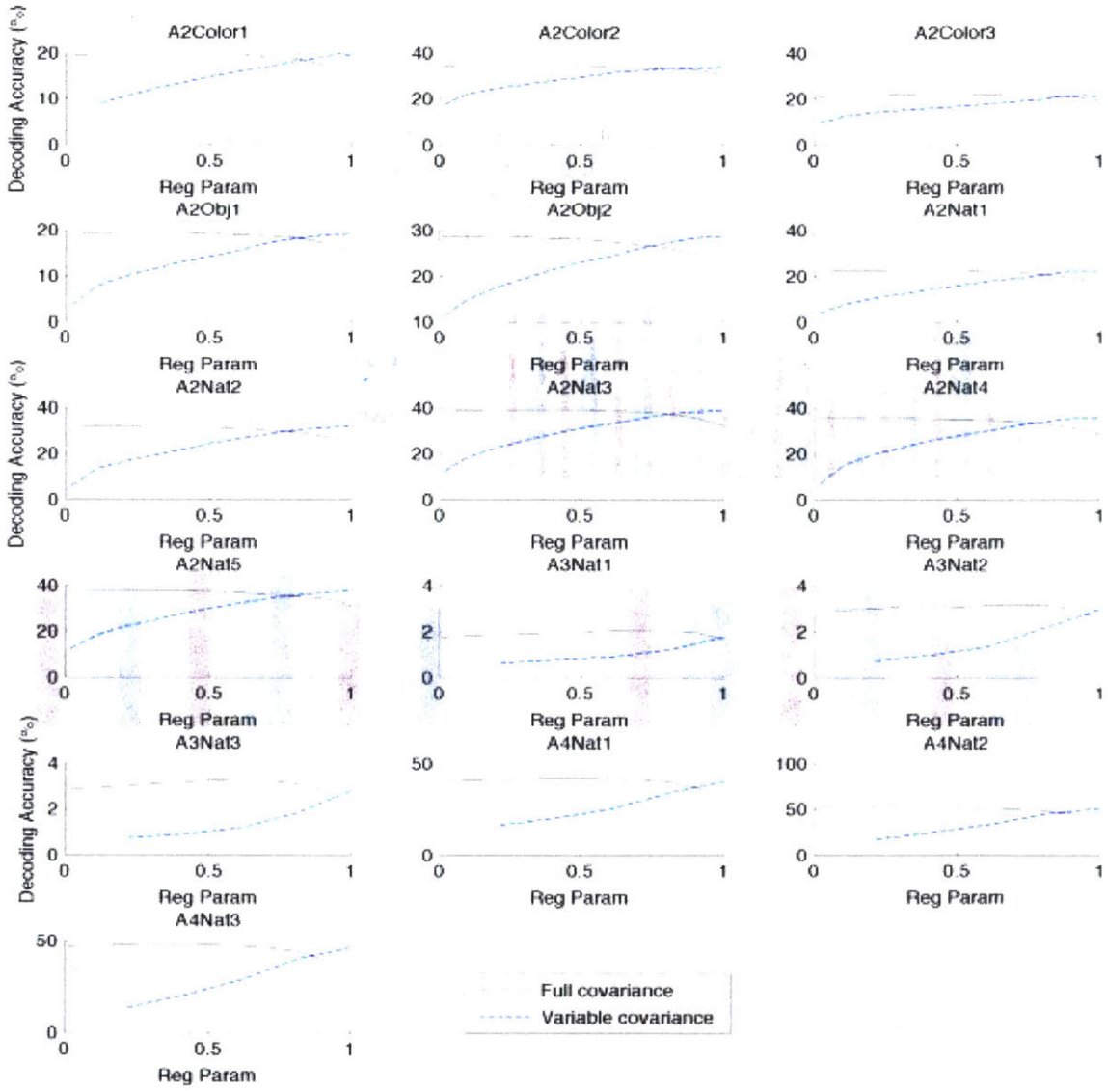
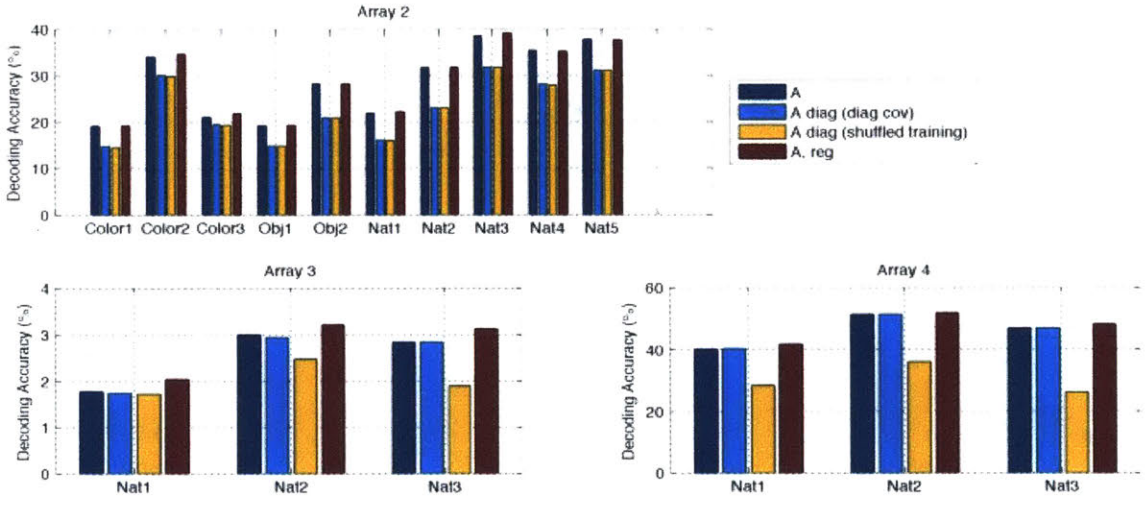


Figure 3-4: A and A_{diag} for each recording session, using the two methods of computing A_{diag} - Method 1, using the diagonal covariance classifier, and Method 2, training the full covariance classifier on shuffled data and testing on unshuffled data. A computed with the regularized full covariance classifier is also shown.



for Array 4, the diagonal covariance classifier that assumes no correlations actually outperforms the global noise model classifiers. Array 3 shows yet another trend, with the global noise model classifiers - in particular the MPD classifier - outperforming the full covariance classifier.

3.4 Discussion

In our study of $\Delta A_{shuffled}$, ΔA_{diag}, and other noise models, we find that results often vary widely between different arrays and stimulus sets. An important future direction would be to try to explain and/or reconcile these differences.

In this chapter, we also compare the two methods of computing A_{diag} and find that on the datasets with a high number of recording sites relative to the number of presentations of each stimulus, there is a significant difference between the two. This is possibly because the standard shuffling procedure does not remove all correlations; when the number of recording sites exceeds the number of presentation of each stimulus, the created pseudopopulation will never be able to have trials such that each firing rate came from a different trial in the original simultaneous recordings. This suggests that when computed ΔA_{diag}, one should compare a full covariance classifier with a diagonal covariance classifier. This also has potential implications for the computation of $\Delta A_{shuffled}$, since $A_{shuffled}$ is also trained on shuffled data which may actually contain residual correlations that affect the value of $A_{shuffled}$.

We also employ a regularized version of the full covariance classifier to prevent overfitting, and find that it results in only very slight improvements except with Array 3. Even so, we can assess the validity of reporting results with regularization. The regularized full covariance is essentially created by interpolating the parameters from a full covariance classifier (used to compute A) and a diagonal covariance classifier (used to compute A_{diag}). Is it fair, then, to instead use the regularized full covariance classifier to compute A? An interpretation which supports the use of the regularized classifier to compute A is that the regularized classifier can be though of as only taking *useful* correlations into account. As long as it is being compared to a classifier

Figure 3-2: A and $A_{shuffled}$ for each recording session are shown, along with A, reg and $A_{shuffled}, reg$ which are analogous to A and $A_{shuffled}$ but are computed using the regularized full covariance classifier.

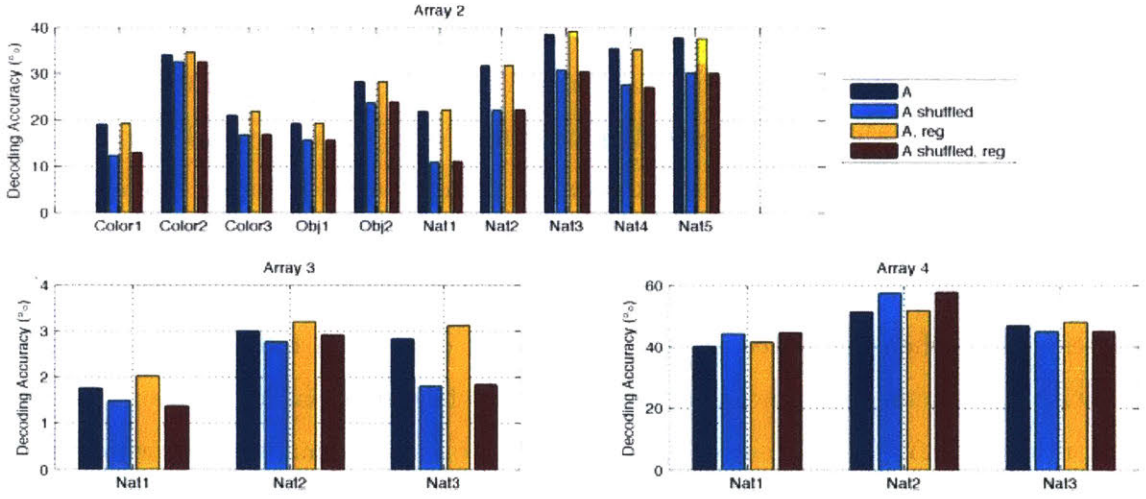


Figure 3-3: Effect of population size on $\Delta A_{shuffled}$

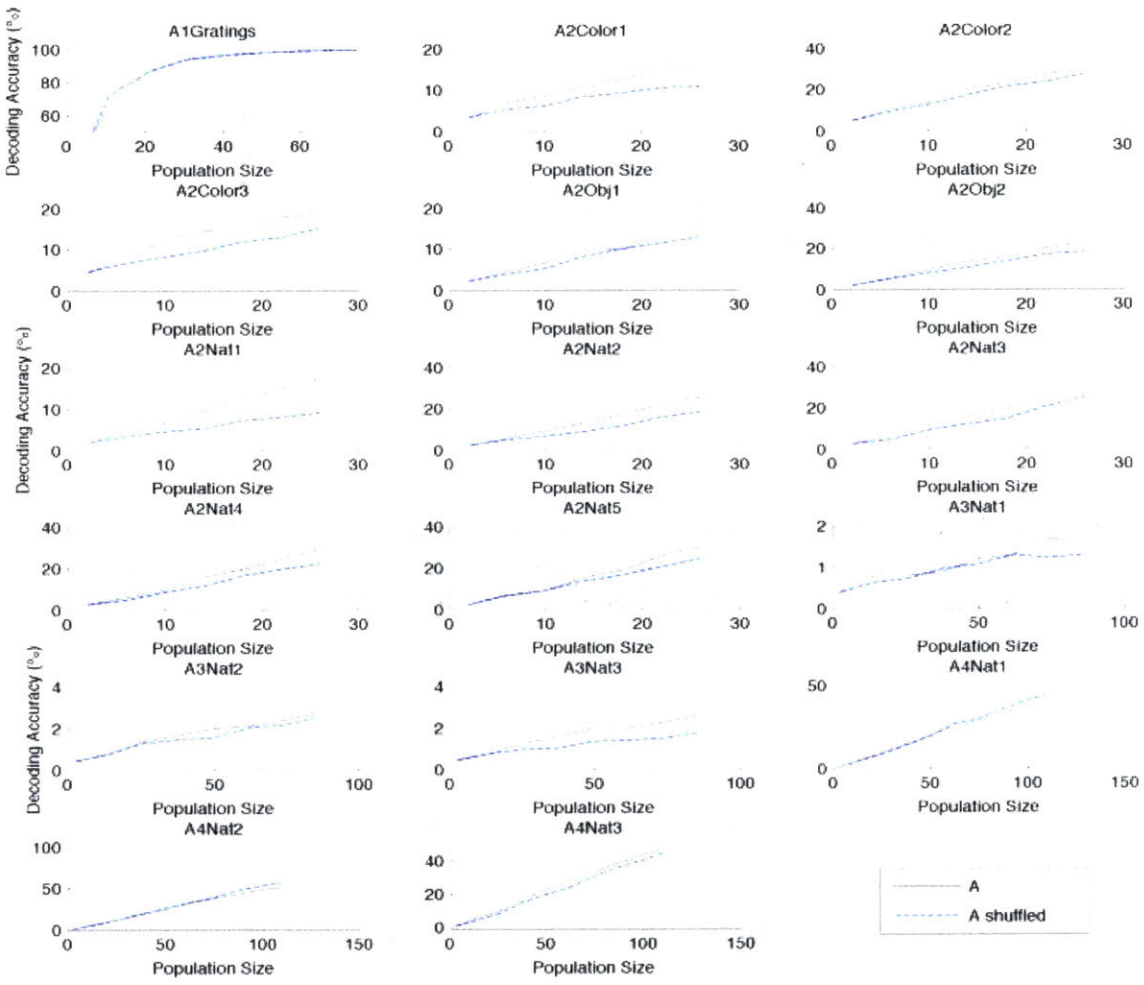


Figure 3-6: Relationship between $\Delta A_{shuffled}/A$ and $\Delta A_{diag}/A$ for each recording session. The different colors represent different arrays.

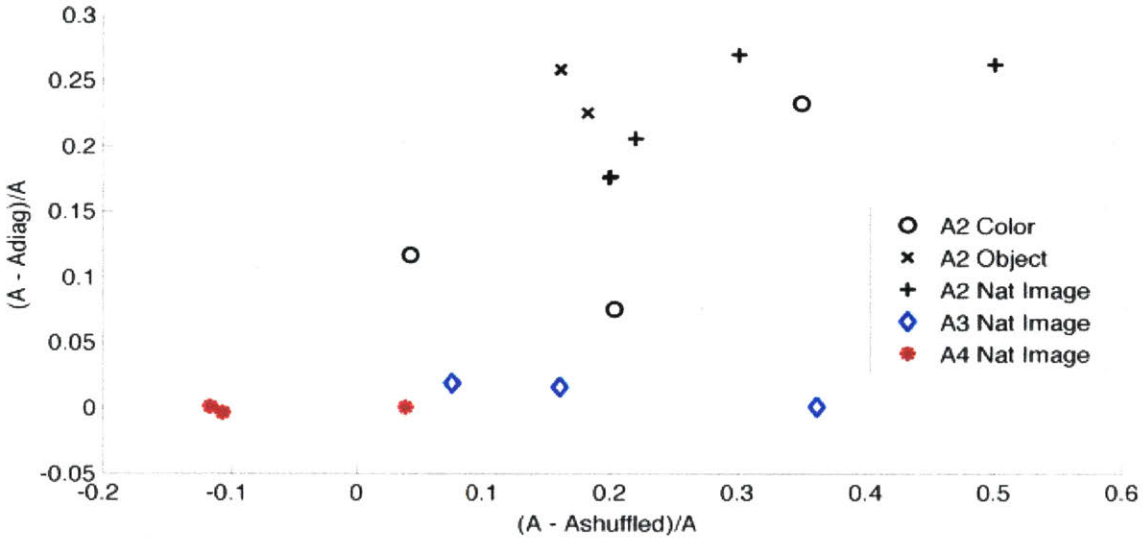


Figure 3-7: Relationship between signal-noise slope and $\Delta A_{shuffled}/A$ for each recording session. The signal-noise slope is computed as the slope of the line of best fit between all signal correlations and all noise correlations (with signal correlations on the x axis).

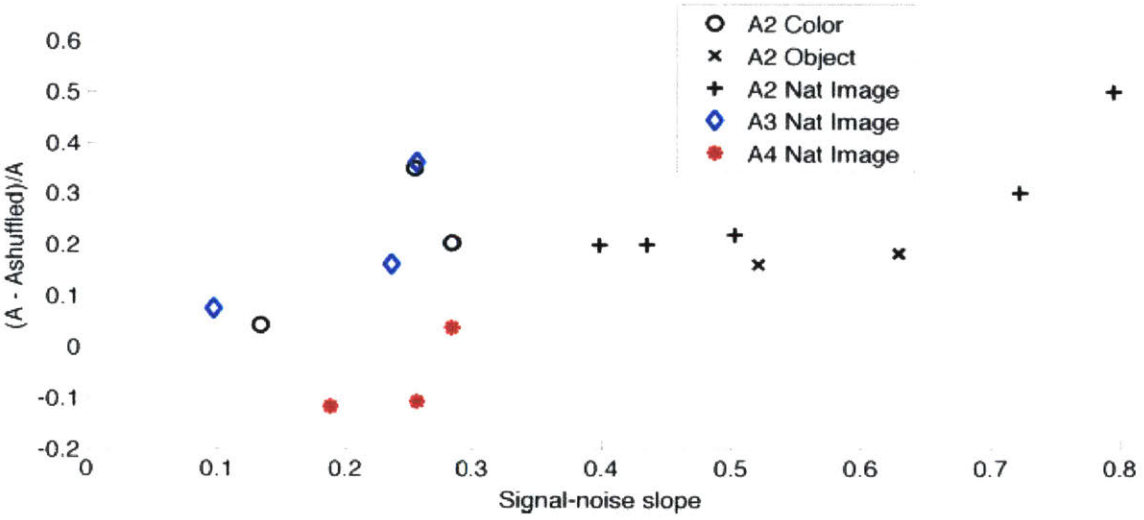


Figure 3-5: Effect of population size on ΔA_{diag}. Both methods of computing A_{diag} are shown: Method 1 - using a diagonal covariance classifier on simultaneous recordings, and Method 2 - using a full covariance classifier on data where the training set has been shuffled to destroy correlations.

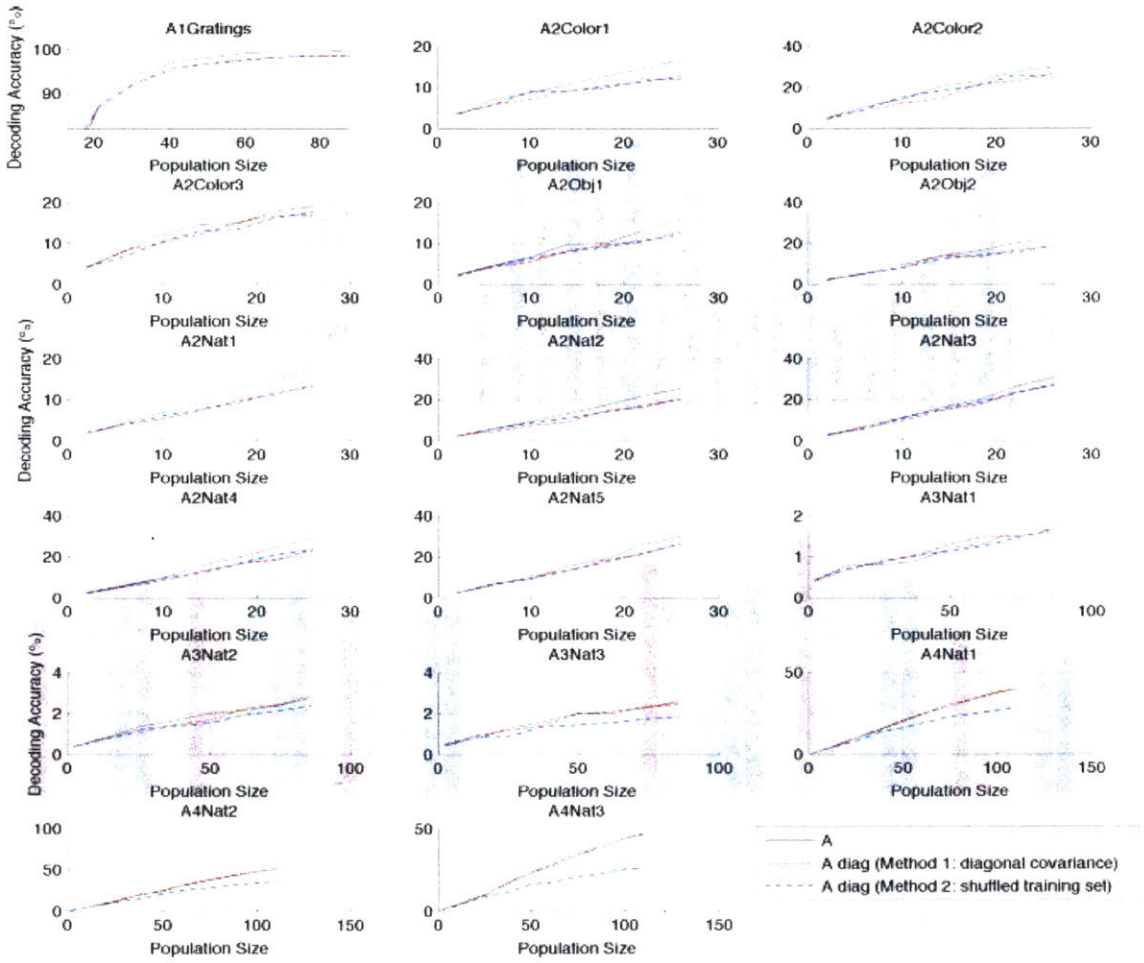
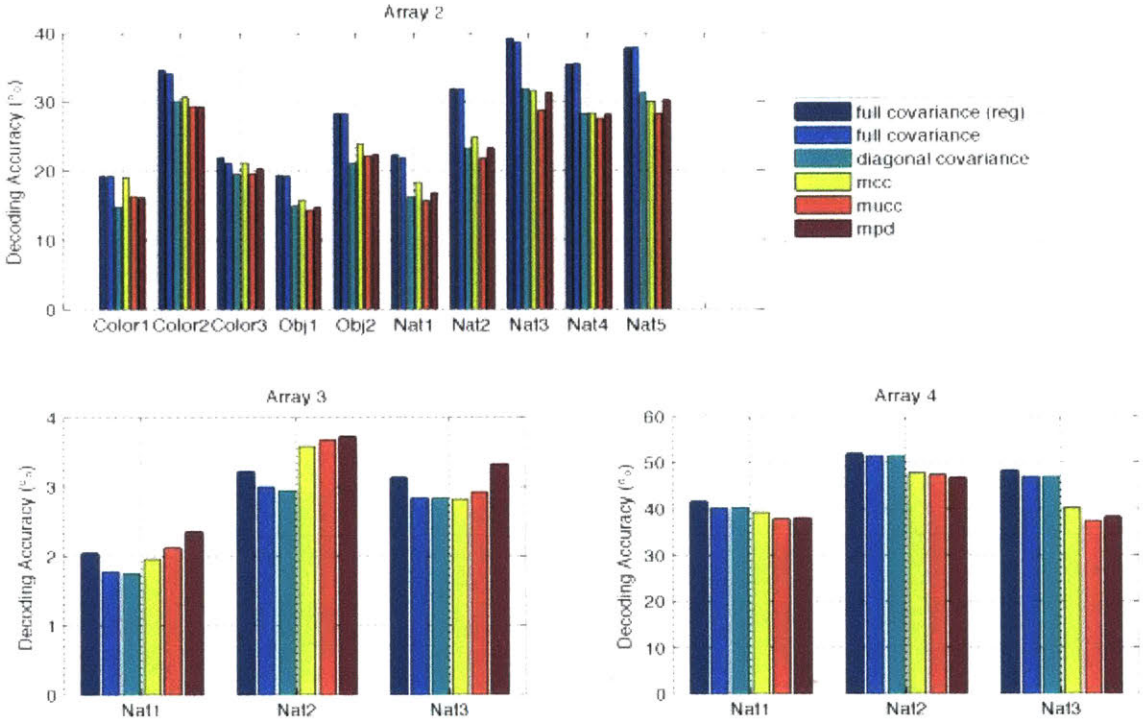


Figure 3-9: Alternative classifiers. Decoding accuracies are shown with classifiers that assume a global noise model: MPD (additive noise), MUCC (multiplicative noise), MCC (additive+multiplicative noise)



Chapter 4

Encoding with Correlation Information

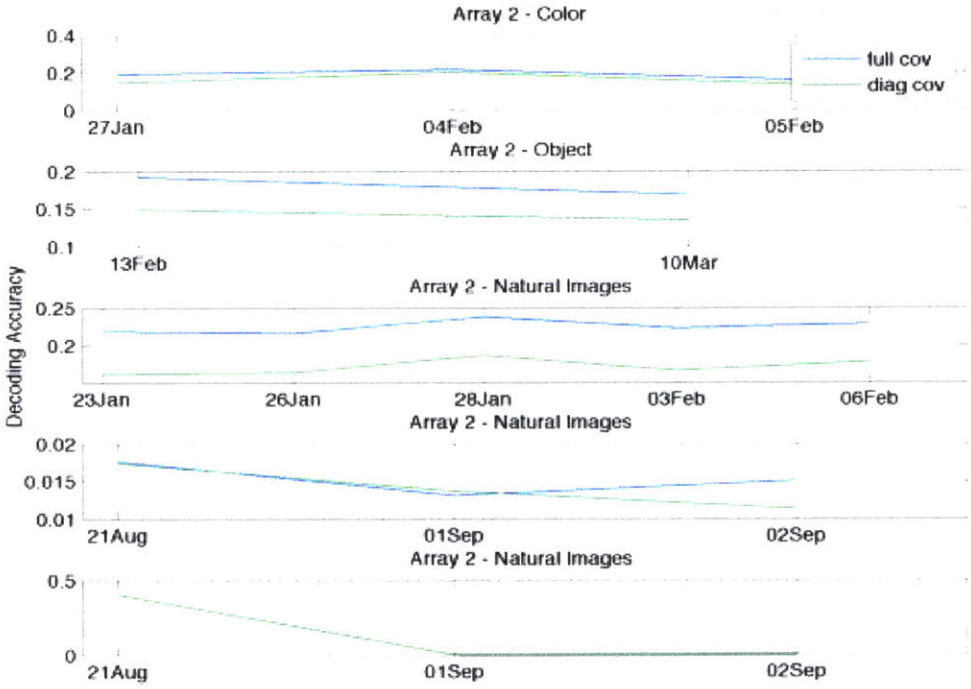
A commonly used method to characterize single neurons is by creating an encoding model - a mapping from stimulus to the neuron's response, in the form of a spatiotemporal receptive field (Abbott and Dayan, 2005). However, with the use of multielectrode recordings, we can also use the simultaneous activity of other neurons as inputs into the model (Paninski, 2004). This can both improve the prediction model, help to assess the relative contributions of both types of data, and give an idea of the functional connectivity of the neural population. It provides a complementary perspective to the decoding perspective in the analysis of the importance of correlations in a neural population.

In this chapter, we perform firing rate prediction with two classes of models:

- Global noise models, which predict firing rate as the mean stimulus response with multiplicative and/or additive noise factors that are shared across neurons for a given trial
- Correlation models, which predict firing rate based on the behavior of other neurons, and optionally the mean stimulus responses

We find that both models improve firing rate predictions, with the much simpler global noise model achieving nearly the same degree of improvement as the correlation

Figure 3-8: Accuracy of classifier trained on one day and tested on others. For each stimulus set, a classifier is trained on data from the first recording session for that array/stimulus set combination and tested on all other recording sessions with the same array and stimulus set. The accuracies shown for the first recording session of each stimulus set are the standard single-session cross-validation accuracies.



model.

4.1 Encoding Model and Related Work

We use linearized discrete reverse correlation for our encoding models. Our predictions are described by the equation:

$$p_{ikm} = x_{km}^T \alpha_i \tag{4.1}$$

where p_{ikm} is the predicted firing rate in presentation m of stimulus k for site i, x_{km} is a column vector of independent variables for presentation m of stimulus k, (e.g. stimulus information or firing rates of other neurons), and α_i is a column vector representing the learned set of coefficients for site i (for example, when stimulus information makes up the independent variables, α_i is the receptive field).

It has been shown that the reverse correlation solution for α_i is equivalent to the linear least squares estimate (Lesica 2009).

$$\alpha_i' = {C_x}^{-1} X r_i \tag{4.2}$$

where r_i is the vector of firing rates for all trials for site i, α_i' is the receptive field estimate, X is the a matrix with columns being different values of x_{km}, and C_x is the independent variable covariance matrix. Xr_i is known as the spike-triggered average.

There are, however, different ways for dealing with the fact that the covariance matrix C_x is often non-invertible. Many of these involve computing a pseudo-inverse, in which some dimensions are forced to zero if the stimulus variance along that dimension is smaller than a noise threshold. David et al (2006) and Theunissen et al (2001), who both use stimulus information to predict firing rates, use cross validation to choose the optimal value for this threshold.

Other variations on this method include linear-nonlinear models (Paninski, 2004;

same additive noise constant:

$$p_{ikm}^{add} = \mu_{ik} + \alpha_{km}^{add} \tag{4.5}$$

The multiplicative model is similar, but assumes a multiplicative constant instead:

$$p_{ikm}^{mult} = \alpha_{km}^{mult} \mu_{ik} \tag{4.6}$$

The linear model contains both the additive and multiplicative constants:

$$p_{ikm}^{linear} = \alpha_{km}^{mult} \mu_{ik} + \alpha_{km}^{add} \tag{4.7}$$

Note that while other work learns a stimulus-response function to model the contribution of the stimulus to firing rate, our use of the mean firing rate in response to a particular stimulus should always preform better than such models in terms of the mean squared prediction error.

We use a cross validation in which a portion of neurons are held out to train and test the global noise models. However, the stimulus mean for each neuron is computed after excluding the particular trial for which the model is being learned.

In Figure 4-1 we see that for Array 2, the more complicated linear model, with two noise parameters instead of 1, outperforms both of the other models. For the other arrays, however, all global noise models perform about equally. For Array 2, there is some correspondence with the results using global noise models for decoding in that the linear model (for decoding, the MCC classifier represented linear noise) tends to perform better than the other global noise models. However, the decoding results also showed the additive noise model consistently performing better for data from Array 3, which is not reflected in the results for encoding.

Though this noise model is simplistic, we can also examine whether the noise carries information about stimulus identity. To do this, we compute the eta-squared values of the multiplicative and additive factors with the stimulus label for each trial. Based on the very low eta-squared values, we do not find find an effect of the stimulus

spherically asymmetric).

4.2 Results

In this section, we report our results for firing rate prediction when using a global noise model and when using other neurons as independent variables.

While other studies report results in terms of the correlation coefficient between the predicted and mean responses, we report normalized root mean squared errors (NRMSE) so as to not ignore multiplicative or additive noise in the predictions. To compute the NRMSE, we divide the root mean squared error of the predicted values by the mean of all observed values.

Like in Chapters 2 and 3, the name of each recording session contains its array number and stimulus set. For example, the name "A2Color1" is used for the first recording session on which the Color stimulus set was used with Array 2.

4.2.1 Global Noise Models

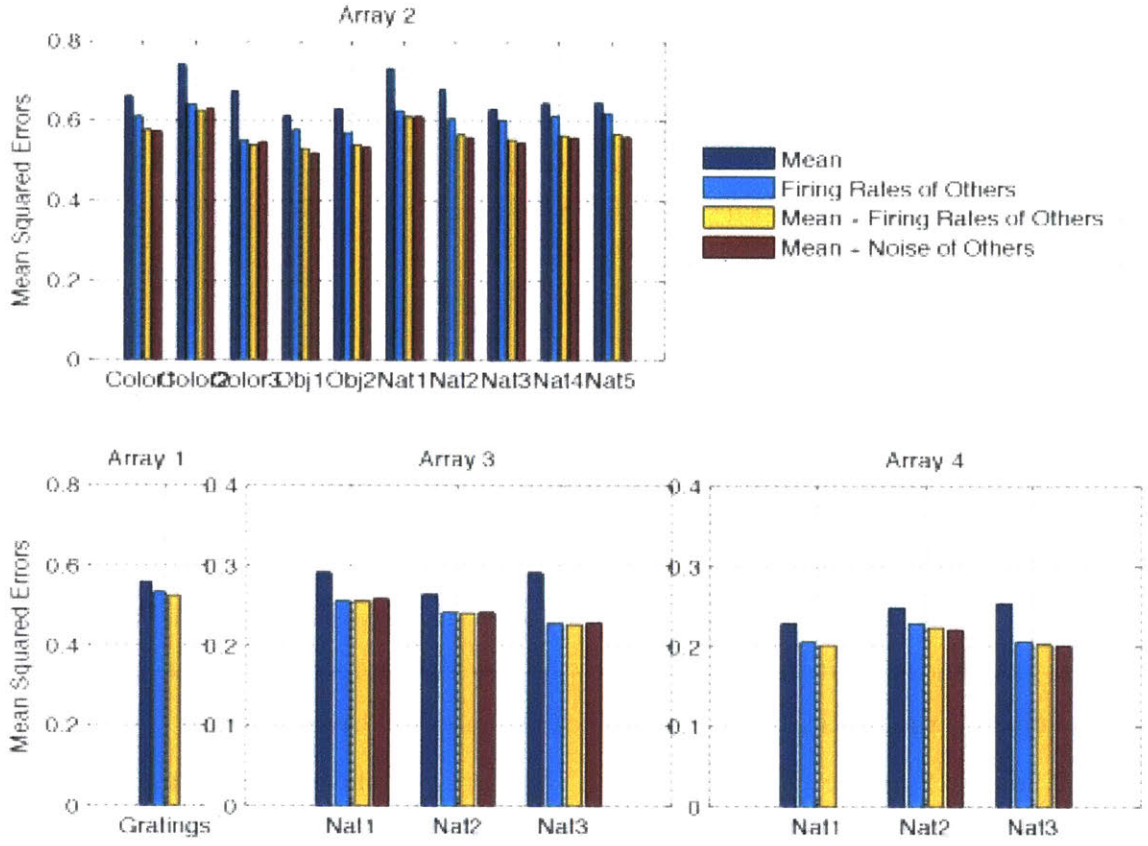
We first examine simple noise models, in which a particular neuron's firing rate is predicted by its mean response to the stimulus plus a constant additive factor (additive model), multiplicative factor (multiplicative model), or both (linear model). These factors are shared across all neurons. While these models are simple, the effects of such models on information decoding were analyzed by Abbott and Dayan (1999). Additionally, as discussed in the previous chapter, some commonly used classifiers assume global noise models. Therefore, we perform encoding based on such noise models as well.

The baseline model simply predicts the firing rate for site i to presentation m of stimulus k as the mean response to a stimulus, μ_{ik}:

$$p_{ikm}^{mean} = \mu_{ik} \tag{4.4}$$

The additive model assumes all neurons' firing rates on presentation m share the

Figure 4-2: Normalized root mean squared errors of instantaneous correlation model. Mean squared errors for each recording day when predicting firing rate with the stimulus mean, the firing rates of others neurons, and both. We also evaluate the model using the noise present in other neurons, where noise is computed by taking the z-score of a firing rate with respect to the stimulus mean and standard deviation.



produces approximately the same results as just using the other firing rates for arrays with larger numbers of recording sites (Arrays 3 and 4) but performs better that just using firing rates for Array 2, which only had 32 recording sites. Using the stimulus mean plus the noise of other neurons performs nearly identically to using the stimulus mean plus firing rates of other neurons, indicating that it is noise rather than signal correlations that help improve the encoding models. Another thing to note is that the model that includes both the mean stimulus response and other neurons' firing rates only improves slightly over the best global noise model we computed, which used both the mean stimulus response and trial-specific additive and multiplicative

Stevenson et al, 2012; Calabrese et al, 2011), governed by the equation

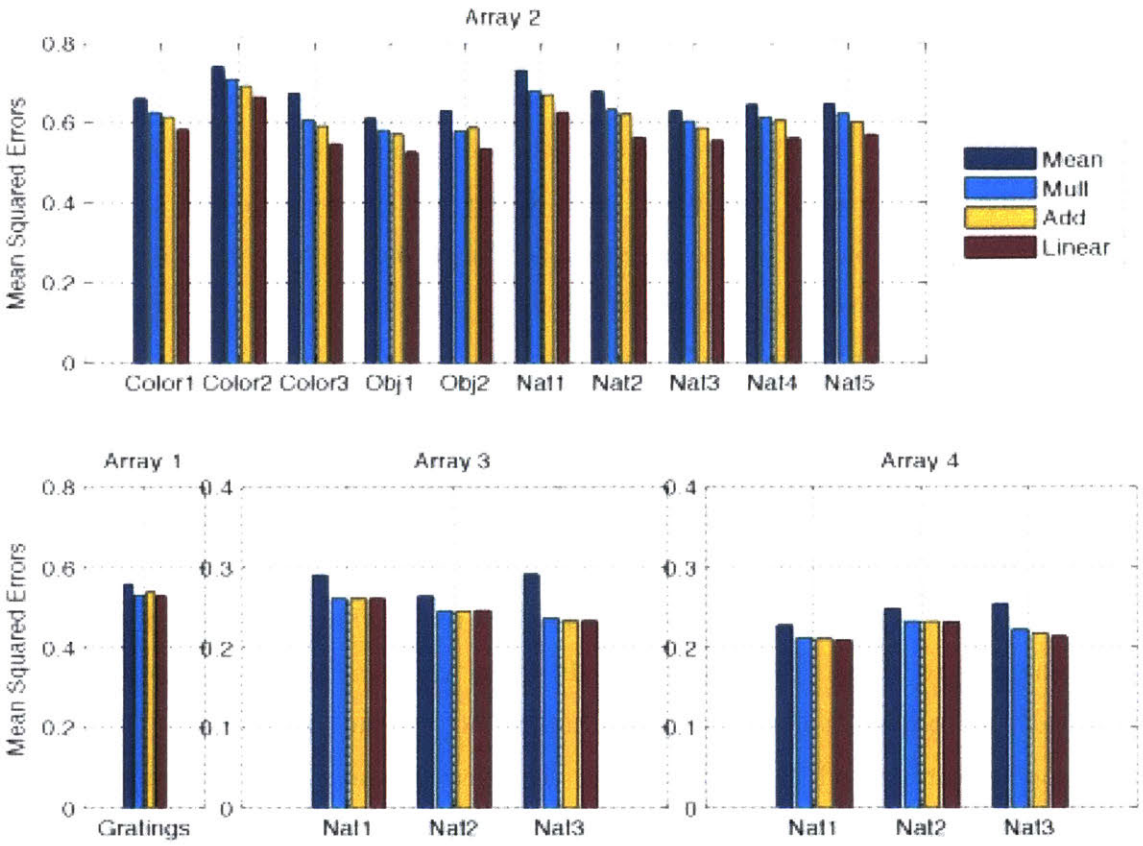
$$p_{ikm} = f({x_{km}}^T \alpha_i) \tag{4.3}$$

where f is what is known as a static non-linearity, usually an exponential function. Paninski (2004) describes this method and conditions on the nonlinearity f for which certain efficient estimation algorithms are possible. Calabrese et al (2011) compare this reverse correlation method to a generalized linear model. Stevenson et al (2012) use firing rate prediction to examine the relative effects of neural connectivity and stimulus tuning in explaining the behavior of neurons. They claim that at population sizes of roughly 10-30 neurons, models using the activity of other neurons begin to outperform the tuning curve models. Vidne et al (2011) also using linear-nonlinear firing rate prediction, with a model that includes the stimulus, the target neuron's firing history, the past behavior of other recorded neurons, and random variables that represent common noise inputs into the set of recorded neurons. When applying their analysis to retinal ganglion cells, they find that the past behavior of other recorded sneurons does not have much importance to the model, while the other independent variables do. In their work, the effect of the stimulus is modeled by a stimulus spatio-temporal filter. As they point out, the physical interpretation of instantaneous correlations such as these are unclear; as a result, their model only includes the past responses of other recorded neurons. The common noise input is meant to represent the instantaneous neural correlations.

Another modification is to use regularization to prevent overfitting. Overfitting can result in low prediction quality on new data due to a model being tuned too well to the data on which it was learned. In this vein, Stevenson et al (2012) employ an L1-penalty in some of their models to penalize high values of the coefficients α, and David et al (2006) compute a shrinkage filter to achieve a similar goal.

Finally, Lesica et al (2009) describe additional modifications of the method to account for the statistical properties of natural stimuli - for example, that the intensities of natural stimuli are not symmetrically distributed around the mean (i.e. they are

Figure 4-1: Normalized root mean squared errors of global noise model. Errors for each recording day when assuming a no noise (mean), additive noise (add), multiplicative noise (mult), or both additive and multiplicative noise (linear)



on the noise values.

4.2.2 Prediction with Correlations

As discussed in Paninski (2004), the independent variables used to predict firing rate can in fact be the firing rates of other neurons - which can be used to examine the functional connectivity of a neural population (Stevenson et al, 2012). The corresponding model (where "firing rates of other neurons" is abbreviated "fron") is:

$$p_{ikm}^{fron} = b_i + \sum_{i \neq j} \alpha_{ij} r_{jkm} \tag{4.8}$$

where b_i is the baseline firing rate for site i.

To evaluate the improvement gained by incorporating the behavior of other neurons compared to the stimulus mean, we use a model (abbreviated "mean, from") that incorporates both:

$$p_{ikm}^{mean,fron} = \mu_{ik} + \sum_{i \neq j} \alpha_{ij} r_{jm} \tag{4.9}$$

Since μ_i represents the stimulus mean, we do not need to include a term for a baseline firing rate. As with the global noise models, we use the stimulus mean as a lower bound on the amount on the mean squared error provided by a receptive field estimate.

Finally, we create a model that predicts firing rate based on the stimulus mean and noise of the other neurons - that is, the z-score of r_{im} computed using the stimulus mean and stimulus standard deviation for neuron j (abbreviated as "mean, non"):

$$p_{ikm}^{mean,non} = \mu_{ik} + \sum_{i \neq j} \alpha_{ij} \frac{r_{jm} - \mu_{jk}}{\sigma_{jk}} \tag{4.10}$$

This essentially removes any stimulus-related information from the firing rates of the other neurons, so it can help assess whether signal or noise correlation is responsible for any improvement in performance with models incorporating both mean and activity of other neurons. If using the noise of other neurons performs comparably to using the actual firing rates, it confirms that noise correlations are responsible for the performance of these models.

For these encoding models, we apply cross validation by dividing the trials into folds, with each fold containing data from one presentation of each stimulus (this differs from the global noise models, in which the recording sites were divided into folds). As is the standard cross validation procedure, each fold gets held out while a model is trained on the remaining trials. The results when testing on the held-out folds are averaged together.

As shown in Figure 4-2, using solely the firing rates of other neurons to predict firing rate yields better prediction results than using just that site's mean response to a stimulus. Using both the stimulus mean as well as other neurons' firing rates

the improvement gained by correlations in encoding is the difference between prediction accuracies in the stimulus mean + firing rates of other neurons model and the stimulus mean model, divided by the prediction accuracy of the stimulus mean model.

In Chapter 3, we noted positive correlations between mean signal correlation, mean noise correlation, signal-noise slope, and $(A - A_{shuffled})/A$, with the strongest relationship between the two latter metrics. We find that the improvement gained by correlations is also positively correlated with each of these values; however, its strongest positive relationship is with the mean noise correlation. This relationship is shown in Figure 4-4. Thus, while the effect of correlations on decoding seems more tied to the relationship between signal and noise correlation, for encoding models it seems to be better explained simply by the average noise correlation of a dataset.

4.3 Discussion

The goal of this chapter was to predict firing rates using the behavior of other neurons. We find that assuming all sites are subject to the same additive and multiplicative noise constants on a given trial improve prediction results by an average of 12.48% over just predicting firing rate using the mean stimulus response. Taking into account all pairwise correlations provides an improvement of 13.92% over the mean stimulus response model. Though these results suggest that the simpler global noise model with only 2 parameters performs nearly as well as the more expensive correlation model, we might find that modifying the correlation model with techniques such as regularization or log-linear modeling might improve its behavior so that there is a larger difference between it and the global noise model. This direction is left for future work.

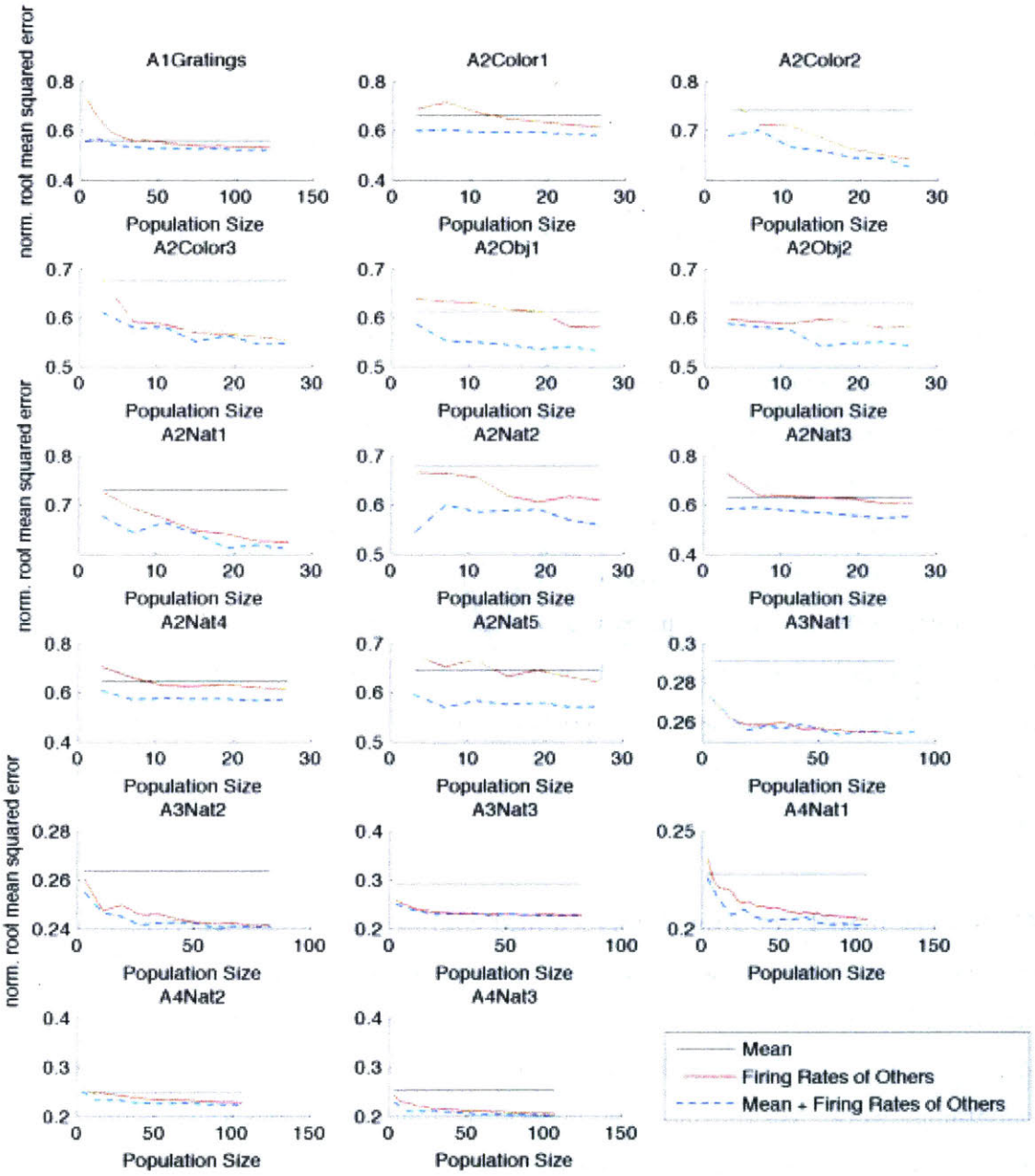
The global noise models and correlation models used for encoding in this chapter roughly correspond to the global noise model classifiers and full covariance classifier, respectively, used in the previous chapter for decoding. However, there wasn't a strict correspondence between the relative performance of different global noise models used

noise factors shared among all sites. On average, using both the stimulus mean plus other neurons' firing rates improves over using just the stimulus mean by 13.92%. Using the linear global noise model improves over just using the stimulus mean by an average of 12.48%, which is nearly 90% of the improvement gained by computing all pairwise correlations.

Effect of Population Size Stevenson et al (2012) report that the firing rate of about 10-30 other neurons, depending on the dataset, can predict the firing rate of a particular left out neuron at a accuracy comparable to when stimulus-related information is used to predict the neuron's firing rate. We perform a similar analysis on our data to examine how many neurons are necessary to perform as well as the stimulus-related information. Because our stimulus information is represented by the mean response to the stimulus and therefore typically has lower errors than using a stimulus filter, we might expect that more neurons would be required to achieve this level of performance. However, we find that, depending on the recording session, populations of fewer than 20 sites are generally sufficient to predict firing rate with error equal to the mean stimulus response - the exception being Array 1, whose sites had very low noise correlations. Still, in Figure 4-3 the stimulus-related information tends to continue to improve the encoding model when both it and other neurons' firing rates are used for most datasets - with a notable exception being the data from Array 3.

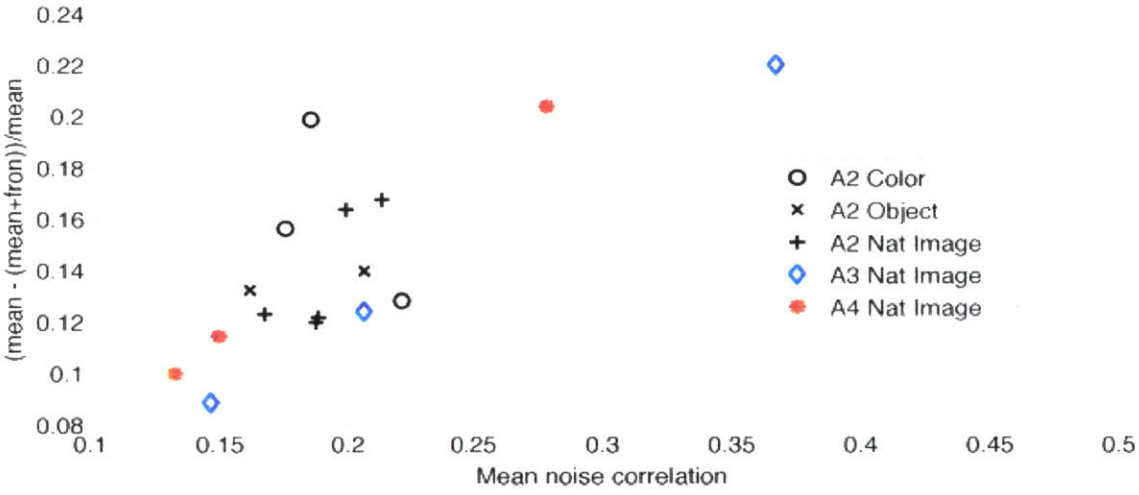
Comparison to Decoding Analyses and Signal and Noise Correlations In this section, we examine the relationship between the improvement in encoding models gained by correlations with metrics computed in other chapters, such as $(A - A_{shuffled})/A$, $(A - A_{diag})/A$, mean signal correlation, mean noise correlation and signal-noise slope. We use the same method of computing correlations between two different metrics as we did in Chapter 3. The Spearman rank correlation coefficients are computed separately for each array/stimulus set combination, and the results are averaged together to obtain the overall correlation. The metric we use to represent

Figure 4-3: Effect of population size on normalized root mean squared error of correlation model. For most recording days, 8-20 neurons are sufficient to achieve the same error as the stimulus mean



for encoding and decoding, besides the fact that the linear model tended to do best for data from Array 2. For the correlation models, though there was a positive relationship between the benefit of correlations in decoding analyses (measured by $(A - A_{shuffled})/A$) and the improvement gained by using correlations in encoding models, the former was more closely related to the signal-noise slope of a recording session, and the latter was better explained by the mean noise correlation of that session.

Figure 4-4: Relationship between encoding model improvement with correlations and mean noise correlation. Encoding model improvement with correlations is defined as the difference between prediction errors with the the stimulus mean + firing rates of other neurons model and the stimulus mean model, divided by the error with the stimulus mean model.



correlations - the global noise model. In this model, the responses on a given trial were assumed to be given by each site's mean response to that stimulus and linear and/or multiplicative noise constants, that were identical for all sites on that trial. Though one array's data was best explained by the linear noise model for both decoding and encoding, another array's data was best explained but the additive noise model for decoding, but was predicted by all three noise models with relatively equal performance for encoding. Thus, the relationship between these models for encoding and decoding is unclear and is room for future work.

Future work in both decoding and encoding might also focus on developing other classifiers and prediction models. The full and diagonal covariance classifiers and the linear least squared estimator are all optimized for data that follows a Gaussian distribution. However, there are alternatives, including using Gibbs sampling to estimate the joint distribution of the neural responses given the stimulus - as done by Stevenson (2010) - for decoding. For encoding, a linear-nonlinear model (Paninski, 2004) is meant to reflect the fact that neural responses tend to follow a Poisson, rather than Gaussian, distribution.

Analysis of more data might also better characterize the relationships between signal correlation, noise correlation, signal-noise slope, and the effect of correlations on decoding and encoding models. Our finding that mean noise correlation is most predictive of improvements made by adding correlations to encoding models makes sense, given that prediction with the stimulus mean and noise of other neurons performed similarly to using the stimulus mean and firing rates of other neurons, indicating that noise correlations are responsible for the improvement. However, our finding that a higher signal-noise slope is strongly linked to a higher $\Delta A_{shuffled}$ contradicts theoretical results from other studies. Confirming and quantifying this relationship is an important future direction.

Chapter 5

Conclusion

In this thesis, we studied neuronal correlations in the contexts of both population decoding and encoding. For population decoding, we first compared decoding accuracies in populations with and without correlations, finding that the results vary by recording session. We found that recordings from V4 tended to have a significant reduction in decoding accuracy when using shuffled data with noise correlations, but for recordings from V1 and IT the effect was smaller, and sometimes involved a gain in decoding accuracy on on shuffled data. More data would need to be analyzed to answer whether this trend holds true for other recordings from these regions. We then asked whether ignoring the correlations that *do* exist in the population significantly reduces decoding accuracy. We again found mixed results, with some datasets showing a significant difference between the two conditions and others showing a insubstantial one.

For encoding we used models that incorporated the behavior of other neurons (correlation information) and/or a site's mean response to a stimulus (stimulus information) to predict that site's firing rate. Using both the correlation and stimulus information improved prediction accuracy by nearly 14% over using just the stimulus information. For some datasets - especially those recorded on Array 3, which had a large population size, using both correlation and stimulus information did not provide significant improvement to a model that just used correlation information

For both encoding and decoding, we also looked at a more simplified model of

[11] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology (London)*, 160:106–154, 1962.

[12] A. Jackson and E.E. Fetz. Compact movable microwire array for long-term chronic unit recording in cerebral cortex of primates. *J Neurophysiol*, 98(5):3109–18, 2007.

[13] K. Kawasaki and D.L. Sheinberg. Learning to recognize visual objects with microstimulation in inferior temporal cortex. *J Neurophysiol*, 100(1):197–211, 2008.

[14] Nikolaus Kriegeskorte, Marieke Mur, Douglas A. Ruff, Roozbeh Kiani, Jerzy Bodurka, Hossein Esteky, Keiji Tanaka, and Peter A. Bandettini. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60:1126–1141, 2008.

[15] E. M. Maynard, N. G. Hatsopoulos, C. L. Ojakangas, B. D. Acuna, J. N. Sanes, R. A. Normann, and J. P. Donoghue. Neuronal Interactions Improve Cortical Population Coding of Movement Direction. *The Journal of Neuroscience*, 19(18):8083–8093, September 1999.

[16] Ethan M. Meyers. The neural decoding toolbox, 2013.

[17] S. Nirenberg, S. M. Carcieri, A. L. Jacobs, and P. E. Latham. Retinal ganglion cells act largely as independent encoders. *Nature*, 411(6838):698–701, 2001.

[18] Liam Paninski. Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15(4):243–262, November 2004.

[19] A. Pasupathy and C. E. Connor. Population coding of shape in area v4. *Nature Neuroscience*, 5:1332–1338, 2002.

[20] Jasper Poort and Pieter R. Roelfsema Roelfsema. Noise correlations have little influence on the coding of selective attention in area v1, 2009.

[21] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999.

[22] AW Roe, L Chelazzi, CE Connor, BR Conway, I Fujita, J Gallant, H Lu, and W Vanduffel. Towards a unified theory of visual area v4. *Neuron*, 74:12–29, 2012.

[23] I.H. Stevenson, B.M. London, E.R. Oby, N.A. Sachs, J. Reimer, B. Englitz, S.V. David, S.A. Shamma, T.J. Blanche, K. Mizuseki, A. Zandvakili, N.G. Hatsopoulos, L.E. Miller, and K.P. Kording. Functional connectivity and tuning curves in populations of simultaneously recorded neurons. *PLoS Comput Biol*, 8(11):e1002775, 2012.

Bibliography

[1] L. F. Abbott and Peter Dayan. The effect of correlated variability on the accuracy of a population code. *Neural Computation*, 11:91–101, 1999.

[2] Bruno B. Averbeck, Peter E. Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature reviews. Neuroscience*, 7(5):358–366, may 2006.

[3] Bruno B. Averbeck and Daeyeol Lee. Neural noise and movement-related codes in the macaque supplementary motor area. *The Journal of Neuroscience*, 23(29):7630–7641, 2003.

[4] Bruno B. Averbeck and Daeyeol Lee. Effects of Noise Correlations on Information Encoding and Decoding. *Journal of Neurophysiology*, 95(6):3633–3644, June 2006.

[5] Carlo Baldassi, Alireza Alemi-Neissi, Marino Pagan, James J. DiCarlo, Riccardo Zecchina, and Davide Zoccolan. Shape similarity, better than semantic membership, accounts for the structure of visual object representations in a population of monkey inferotemporal neurons. *PLoS Comput Biol*, 9(8):e1003167, 08 2013.

[6] Charles Cadieu, Minjoon Kouh, Anitha Pasupathy, Charles E. Connor, Maximilian Riesenhuber, and Tomaso Poggio. A model of v4 shape selectivity and invariance. *J Neurophysiol*, 98(3):1733–1750, 2007.

[7] Marlene R. Cohen and John H. R. Maunsell. Attention improves performance primarily by reducing interneuronal correlations. *Nature Neuroscience*, 12(12):1594–1600, November 2009.

[8] M.R. Cohen and A. Kohn. Measuring and interpreting neuronal correlations. *Nat Neurosci*, 14(7):811–9, 2011.

[9] Adam S. Dickey, Aaron Suminski, Yali Amit, and Nicholas G. Hatsopoulos. Single-Unit Stability Using Chronically Implanted Multielectrode Arrays. *Journal of Neurophysiology*, 102(2):1331–1339, August 2009.

[10] Y. Gu, S. Liu, C.R. Fetsch, Y. Yang, S. Fok, A. Sunkara, G.C. DeAngelis, and D.E. Angelaki. Perceptual learning reduces interneuronal correlations in macaque visual cortex. *Neuron*, 71(4):750–61, 2011.

[24] M Vidne, Y Ahmadian, J Shlens, J Pillow, J W Kulkarni, E P Simoncelli, E J Chichilnisky, and L Paninski. A common-input model of a complete network of ganglion cells in the primate retina. In *Computational and Systems Neuroscience (CoSyNe)*, Salt Lake City, Utah, February 2010.