Thanks

WCMC Neurology and Neuroscience

Keith Purpura
Ferenc Mechler
Anita Schmid
Ifije Ohiorhenuan
Qin Hu

Collaborators

Sheila Nirenberg (WCMC)
Dan Gardner (WCMC)
Bruce Knight (Rockefeller)
Patricia DiLorenzo (SUNY Binghamton)

Former students

Dmitriy Aronov
Danny Reich
Michael Repucci
Bob DeBellis

Support

NEI
NINDS
WCMC Neurology and Neuroscience
Outline

• Information
  – Why calculate information?
  – Definition and properties
  – Why is it challenging to estimate?
  – Useful strategies
    • Generic
    • Spike trains
  – Application examples

• Maximum entropy methods
  – Data analysis
  – Stimulus design
Information: Why Calculate It?

• An interesting, natural quantity
  – Compare across systems (e.g., “one spike per bit”)
  – Determine the constraints on a system (e.g., metabolic cost of information)
  – See where it is lost (e.g., within a neuron)
  – Insight into what the system is “designed” to do

• To evaluate candidates for neural codes
  – What statistical features are available?
    • Can precise spike timing carry information?
    • Can neuronal diversity carry information?
  – What codes can be rigorously ruled out?

• A non-parametric measure of association
Even in visual cortex, the neural code unknown

What physiologists say:
- Neurons have definite selectivities ("tuning")
- Tuning properties can account for behavior

What physiologists also know:
- Responses depend on multiple stimulus parameters
- Response variability (number of spikes, and firing pattern) is substantial and complicated

Hubel and Wiesel 1968
Some coding hypotheses

• At the level of individual neurons
  – Spike count
  – Firing rate envelope
  – Interspike interval pattern, e.g., bursts

• At the level of neural populations
  – Total population activity
  – Labeled lines
  – Patterns across neurons, e.g., synchrony
  – Oscillations
Coding by intervals can be faster than coding by count

Signaling a step change in a sensory input

coding by count (rate averaged over time)

coding by interval pattern

One short interval indicates a change!
Coding by rate envelope supports signaling of multiple attributes.

More spikes

More transient

Time

Codes based on spike patterns can also support signaling of multiple attributes.
A direct experimental test of a neural coding hypothesis is difficult

- Count, rate, and pattern are interdependent

  “Time is that great gift of nature which keeps everything from happening at once.” (C.J. Overbeck, 1978)

- We’d have to manipulate count, rate, and pattern selectively AND observe an effect on behavior

- So, we need some guidance from theory
A Principled Approach to Neural Coding

• Goal: account for the behavior

• *Information* determines the limits of behavior
  – We can measure it from behavior (in principle)
  – We can measure it from neural activity (in principle)

• Two properties of information
  – Data Processing Inequality
  – Independent channels combine additively

• Good news and bad news
  – These properties imply a unique definition of information
  – But they also imply a fundamental problem in implementing the definition

• Addressing this problem refocuses us on biology
Information =
Reduction in Uncertainty
(Claude Shannon, 1948)

- Reduction in uncertainty from 6 possibilities to 2
- Information = $\log(6/2)$
In a bit more detail:

**A priori knowledge**

Observe a response

**A posteriori knowledge**
Second-guessing shouldn’t help

No spikes
One spike
Two spikes

Maybe there really should have been a spike?
Maybe these two kinds of responses should be pooled?

The “Data Processing Inequality”: information cannot be increased by re-analysis.
Information on independent channels should add

**Color channel**

- Observe a response
- \( \log(6/2) = \log(3) \)

**Shape channel**

- Observe a response
- \( \log(6/3) = \log(2) \)

**Both channels**

- Observe a response
- \( \log(6/1) = \log(6) \)

\[ \log(6) = \log(3 \times 2) = \log(3) + \log(2) \]
Surprising Consequence

Data Processing Inequality

+ 

Independent channels combine additively

+ 

Continuity

= 

Unique definition of information, up to a constant
Information: Difference of Entropies

Information = \{ \text{Entropy of the a priori distribution of input symbols} \} 
minus 
\{ \text{Entropy of a posteriori distribution of input symbols, given the observation } k, \text{ averaged over all } k \}
Entropy: Definition (discrete)

\[ H = - \sum_j p_j \log p_j \]  
( use log_2 to get “bits”)

Equivalent forms, useful for analytic purposes:

\[ H = - \frac{1}{\log 2} \sum_j \frac{d}{d \alpha} p_j^\alpha \bigg|_{\alpha=1} \]

\[ H = - \frac{1}{\log 2} \lim_{\alpha \to 1} \frac{1}{\alpha - 1} \left( \sum_j p_j^\alpha - 1 \right) \]

Essentially, \( H \) describes the limiting behavior of the moments of the probability distribution.
Entropy: Properties

Mixing property \( H\{(1 - \lambda)p_j + \lambda q_j\} \geq (1 - \lambda)H\{p_j\} + \lambda H\{q_j\} \)

Chain rule (refinement)

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c}
& & & & & & \rho & & & & \\
\hline
H(X) & H(Y) & H(Z) &=& H(X) + pH(Y) \\
\end{array}
\]

Chain rule (independent variables in tables)

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c}
& & & & & & & & \ & & & \\
\hline
H(X) & & & & & & & & & & & \\
& H(Y) & & & & & & & & & & \ \\
\hline
& & & & & & & & & & & \\
& & & & & & & & & & & \\
\hline
& & & & & & & & & & & \\
& & & & & & & & & & & \\
\hline
& & & & & & & & & & & \\
& & & & & & & & & & & \\
\hline
& & & & & & & & & & & \\
& & & & & & & & & & & \\
\hline
& & & & & & & & & & & \\
& & & & & & & & & & & \\
\hline
& & & & & & & & & & & \\
& & & & & & & & & & & \\
\end{array}
\]

\( H(Z) = H(X) + H(Y) \)
Information: Symmetric Difference of Entropies

Information = \{\text{entropy of output}\} + \{\text{entropy of input}\} - \{\text{entropy of table}\}

\[ I = H\{p(j,\bullet)\} + H\{p(\bullet,k)\} - H\{p(j,k)\} \]

where \( H\{p\} = -\sum p \log p \)
Information: Key Properties

\[ I = H\{p(j,\bullet)\} + H\{p(\bullet,k)\} - H\{p(j,k)\} \]

- \( I \) is symmetric in input and output
- \( I \geq 0 \), and \( I = 0 \) if and only if \( p(j,k) = p(j,\bullet)p(\bullet,k) \)
- \( I \) is invariant w.r.t. labeling of input and output
- And for continuous distributions, \( I \) is invariant w.r.t. smooth distortions of input and output
- Utility as a nonparametric measure of association

Data Processing Inequality: if \( Y \) determines \( Z \), \( I(X,Y) \geq I(X,Z) \)

Utility for investigating neural coding and behavior
**Information: Related Quantities**

*Channel capacity*
maximum information for any input ensemble

*Efficiency*
\[
\frac{\text{Information}}{\text{Channel capacity}}
\]

*Redundancy*
\[
1 - \frac{\text{Information from all channels}}{\text{sum of informations from each channel}}
\]

*Redundancy Index*
\[
1 - \frac{\text{Information from all channels}}{\text{sum of informations from each channel}}
\]
\[
1 - \frac{\text{Information from all channels}}{\text{maximum of informations from each channel}}
\]
Entropy: Discrete vs. Continuous

Discrete: \[ H = - \sum_j p_j \log p_j \]

Continuous: replace \( p_j \) by \( p(x) \Delta x \), replace \( \sum \Delta x \) by \( \int dx \)

\[ H = - \int p(x) \log p(x) \Delta x dx \]

\[ = - \int p(x) \log p(x) dx - \int p(x) \log(\Delta x) dx \]

\[ = - \int p(x) \log p(x) dx - \log(\Delta x) \quad \text{“differential entropy”} \]

The dimension- and unit-dependent term is cancelled when entropies are subtracted to yield information.

But that does not entitle us to neglect the distinction between “discrete” and “continuous” data when we choose estimators.
Investigating neural coding: not Shannon’s paradigm

• Shannon
  – symbols and codes are known
  – joint (input/output) probabilities are known
  – what are the limits of performance?

• Neural coding
  – symbols and codes are not known
  – joint probabilities must be measured
  – ultimate performance often known (behavior)
  – what are the codes?
Information estimates depend on partitioning of stimulus domain

Finely partitioned

\[ H = \log(4) \text{ bits} \]

Coarsely partitioned

\[ H = \log(2) \text{ bits} \]

Unambiguous; detail is lost.
Information estimates depend on partitioning of response domain.

- Finely partitioned: unambiguous; $H = \log(4)$ bits
- Coarsely partitioned: ambiguous; $H = \log(2)$ bits
Revenge of the Data Processing Inequality

Should these responses be grouped into one code word?

Data Processing Inequality says **NO**: If you group, you underestimate information.
The Basic Difficulty

We need to divide stimulus and response domains finely, to avoid underestimating information (“Data Processing Theorem”).

We want to determine $<p \log p>$, but we only have an estimate of $p$, not its exact value. Dividing stimulus and response domains makes $p$ small. This increases the variability of estimates of $p$.

*But that’s not all…*

$p \log p$ is a nonlinear function of $p$.

Replacing $<p \log p>$ with $<p> \log <p>$ incurs a bias.

How does this bias depend on $p$?
Biased Estimates of $-p \log p$

$f(p) = -p \log p$

Downward bias is greatest where $f''$ is greatest.

$f''(p) = -\frac{1}{p}$
The Classic Debiaser: Good News/ Bad News

We don’t have to debias every $p \log p$ term, just the sum.

The good news (for entropy of a discrete distribution):

The plug-in entropy estimate has an asymptotic bias proportional to $(k-1)/N$, where $N$ is the number of samples and $k$ is the number of different symbols (Miller, Carlton, Treves, Panzeri).

The bad news:

Unless $N \gg k$, the asymptotic correction may be worse than none at all.

More bad news:

We don’t know what $k$ is.
Another debiasing strategy

Toy problem:
\[ \langle x^2 \rangle \neq \langle x \rangle^2 \]

For a parabola, bias is constant.

This is why the naïve estimator for variance can be simply debiased:
\[ \sigma^2_{\text{est}} = \langle (x - \langle x \rangle)^2 \rangle / (N-1) \]

Our problem:
\[ \langle -p \log p \rangle \neq -\langle p \rangle \log \langle p \rangle \]

Bias depends on the best local parabolic approximation. This leads to a polynomial debiaser. (Paninski)

Better than classical debiaser, but \( p=0 \) is still worst case. And it still fails in the extreme undersampled regime.
The “Direct Method”  
(Strong, de Ruyter, Bialek, et al. 1998)

- Discretize the response into binary “words”

\[
\begin{array}{cccccccc}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
\end{array}
\]

- \(T_{\text{letter}}\) must be small to capture temporal detail
  - timing precision of spikes: <1 ms
- \(T_{\text{word}}\) must be “large enough”
  - insect sensory neuron: 12 - 15 ms may be adequate
  - Vertebrate CNS: 100 ms at a minimum

- \(2^{(T_{\text{word}}/T_{\text{letter}})}\) probabilities to estimate
- \(2^{12}\) is manageable, \(2^{100}\) is not
“Birthday Paradox” Methods: Overview

a.k.a. NSB (Nemenman, Shafee, Bialek 2002, 2004)

- Basic insight: don't estimate individual log($p_j$)'s, just their average
  - Warm-up problem: how many $p_j$'s?
  - “Birthday paradox”: Counting coincidences of birthdays is an efficient way to estimate the number of days in a year; if there are $k$ days in a year, you need approximately $N=\sqrt{k}$ observations.

- Implications for the entropy-estimation problem: naively, need $N>>k$ observations, but with NSB, need $N>>\sqrt{k}$ observations.

- Caveat for NSB methods: distribution of probabilities has to be like Zipf's-law ($m$th-highest probability proportional to $m^b$)

- A big improvement, but usually not nearly big enough
  - $k\sim2^{(T_{\text{word}}/T_{\text{letter}})}$
  - $k$ can be comparable to $N^2$, not $N$
  - This doubles the accessible $T_{\text{word}}/T_{\text{letter}}$
Multiple neurons: a severe sampling problem

- One dimension for each bin and each neuron

\[2^{L \left( \frac{T_{\text{word}}}{T_{\text{letter}}} \right)}\] probabilities must be estimated.
What else can we do?

- Spike trains are events in time
- So there are relationships between them:
  - a continuous topology
  - discrete relationships: how many spikes? (and on which neurons?)

- How can we exploit this?
Strategies for Estimating Entropy and Information

- Spike train as a symbol sequence?
  - NO
- Relationship between symbol sequences?
  - NO -> direct
  - YES

- Smooth dependence on spike times?
  - NO
- “Smooth” dependence on spike count?
  - NO
  - YES
  - metric space

- Spike train as a point process?
  - YES
- Relationship between symbol sequences?
  - NO
  - direct
  - YES
  - LZW compression
  - bottleneck/codebook
  - context tree
  - Markov

- “Smooth” dependence on spike count?
  - NO
  - binless embedding
  - power series
  - stimulus reconstruction

most require comparison of two entropy estimates
Strategies for Estimating Entropy and Information

Spike train as a symbol sequence
- NO
  - Relationship between symbol sequences?
    - NO: direct
    - YES: LZW compression, bottleneck/codebook, context tree, Markov

Smooth dependence on spike times?
- YES
  - Spike train as a point process
    - NO: “Smooth” dependence on spike count?
      - NO: binless embedding, power series, stimulus reconstruction
      - YES: metric space
    - YES: stimulus reconstruction
A General Approach to Undersampling

• Create a parametric model for spike trains
• Fit the model parameters
• Calculate entropy from the model (numerically or analytically)

Toy example: Binomial model

\[ p(0010001000100) = p(0) \times p(0) \times p(1) \times \ldots \]

We can determine probabilities of all spike trains from \( p_0 = p(0) \).

Entropy of an \( n \)-bin segment: \( H_n = nH_1 \) (bins are independent).

\[ H_1 = -p_0 \log p_0 - p_1 \log p_1 \]

A good estimate of the entropy of the model -- but a bad model: spiking probabilities are not independent.
Incorporating Simple Dependencies

Say we measure \( p(00), p(01), p(10), p(11) \).

Use these to estimate (model) probabilities of longer trains:

Assuming that dependence (memory) lasts only one bin,

\[
p(ab) = p(a)p(ab|a), \text{ so } p(ab|a) = \frac{p(ab)}{p(a)}.
\]

Iterating,

\[
p(abc) = p(a)p(ab|a)p(bc|b)p(cd|c) = p(a) \times \frac{p(ab)}{p(a)} \times \frac{p(bc)}{p(b)} \times \frac{p(cd)}{p(c)}.
\]

\[
= \frac{p(ab) \times p(bc) \times p(cd)}{p(b) \times p(c)}.
\]
The Markov Model

Say we measure $m$-block probabilities $p(\text{abcd})$:

Use these to estimate (model) probabilities of longer trains:

$$p(\text{abcdef}) = \frac{p(\text{abcd}) \times p(\text{bcde}) \times p(\text{cdef})}{p(\text{bcde}) \times p(\text{cde})}.$$

Entropy of an $n$-bin segment: $H_n \leq nH_1$ (bins are dependent).

Recursion: $H_{n+1} - H_n = H_n - H_{n-1}$ for $n \geq m$ (not obvious)

Analytic expression for entropy per bin: $$\lim_{n \to \infty} \frac{H_n}{n} = H_m - H_{m-1}$$

As $m$ (the Markov order) increases, the model fits progressively better, but there are more parameters to measure ($2^{m-1}$). If $m$ needs to be large to account for long-time correlations, it is not clear whether we’ve gained much over the direct method.
Context Tree Method in a nutshell

- $p$(spike) modeled as a variable-depth tree (Markov: constant-depth)
- Model parameters: tree topology, probability at terminal nodes
- Entropies determined from model parameters (Wolpert-Wolf)
- Weighting of each model according to “minimum description length”
- Confidence limit estimation via Monte Carlo resampling

Kennel, Shlens, Abarbanel, Chichilnisky
Neural Computation (2005)
Strategies for Estimating Entropy and Information

Spike train as a symbol sequence?
- NO
  - Relationship between symbol sequences?
    - NO
      - direct
    - YES
      - LZW compression
      - bottleneck/codebook
      - context tree
      - Markov

Smooth dependence on spike times?
- YES
  - Spike train as a point process?
    - NO
      - “Smooth” dependence on spike count?
        - NO
          - binless embedding
            - power series
            - stimulus reconstruction
        - YES
          - metric space
          - stimulus reconstruction
Binless Method: Strategy

If every spike train had only 1 spike:
then estimation of entropy of spike trains would be the same as empiric estimation of the entropy of a one-dimensional distribution \( p(x) \).

If every spike train had \( r \) spikes:
then estimation of entropy of spike trains would be the same as empiric estimation of the entropy of an \( r \)-dimensional distribution \( p(x_1,x_2,\ldots,x_r) \).

But spike trains can have 0, 1, 2, … spikes.
We can estimate a “spike count” contribution to information. Then, for each \( r \), we can use continuous methods to estimate entropies and a contribution to spike timing information.
Binless Embedding Method in a nutshell

Embed responses with $r$ spikes as points in an $r$-dimensional space.

In each $r$-dimensional stratum, use Kozachenko-Leonenko (1987) nearest-neighbor estimate:

$$l \approx \frac{r}{N \ln 2} \sum_j \ln \left( \frac{\lambda_j}{\lambda_j^*} \right) - \frac{1}{\ln 2} \sum_k \frac{N_k}{N} \ln \frac{N_k - 1}{N - 1}$$

Strategies for Estimating Entropy and Information

Smooth dependence on spike times?

Spike train as a symbol sequence

NO

Relationship between symbol sequences?

NO
  direct

LZW compression
bottleneck/codebook
context tree
Markov

YES

Spike train as a point process

NO

“Smooth” dependence on spike count?

NO
  binless embedding
  power series
  stimulus reconstruction

YES

metric space

Relationship between symbol sequences?

YES

Spike train as a symbol sequence

NO

Smooth dependence on spike times?

YES
Coding hypotheses: in what ways can spike trains be considered similar?

- Similar spike counts
- Similar spike times
- Similar interspike intervals
Measuring similarity based on spike times

- Define the “distance” between two spike trains as the simplest morphing of one spike train into the other by inserting, deleting, and moving spikes.

- Unit cost to insert or delete a spike.
- We don’t know the relative importance of spike timing, so we make it a parameter, $q$: shift a spike in time by $\Delta T$ incurs a cost of $q\Delta T$.
- Spike trains are similar only if spikes occur at similar times (i.e., within $1/q$ sec), so $q$ measures the informative precision of spike timing.
Identification of Minimal-Cost Paths

The algorithm is closely analogous to the Needleman-Wunsch & Sellers (1970) dynamic programming algorithms for genetic sequence comparisons.

“World lines” cannot cross. So, either
(i) The last spike in A is deleted,
(ii) The last spike in B is inserted
(iii) The last spike in A and the last spike in B must correspond via a shift
Distances between all pairs of responses determine a response space

responses to stimulus 1

responses to stimulus 2

responses to stimulus 3

etc.

calculate all pairwise distances
Configuration of the response space tests whether a hypothesized distance is viable.

Random: responses to the four stimuli are interspersed.

Systematic clustering: responses to the stimuli are grouped and nearby groups correspond to similar stimuli.
Metric Space Method in a nutshell

Postulate a parametric family of edit-length metrics (distances”) between spike trains

Allowed elementary transformations:
- insert or delete a spike: unit cost
- shift a spike in time by $\Delta T$: cost is $q \Delta T$

Create a “response space” from the distances

Cluster the responses

Information = row entropy + column entropy - table entropy

Victor and Purpura, Network (1997)
Visual cortex: contrast responses
Visual cortex: contrast coding

- Spike count code
- Interspike interval code
- Shuffle code
- $q_{\text{max}}$

$q$ (cost per unit time to move a spike)
Multiple visual sub-modalities

- contrast
- orientation
- spatial frequency
- texture type
Attributes are coded in distinct ways

in primary visual cortex (V1)

and even more so in V2
Analyzing coding across multiple neurons

A multineuronal activity pattern

Distances between labeled time series can also be defined as the minimal cost to morph one into another, with one new parameter:

- Cost to insert or delete a spike: 1
- Cost to move a spike by an amount $\Delta T$: $q \Delta T$
- Cost to change the label of a spike: $k$

$k$ determines the importance of the neuron of origin of a spike.

- $k=0$: summed population code
- $k$ large: labeled line code
Multineuronal Analysis via the Metric-Space Method: A two-parameter family of codes

- Change the time of a spike: cost/sec = $q$
  - $q=0$: spike count code

- Change the neuron of origin of a spike: cost = $k$
  - $k=0$: summed population code: (neuron doesn’t matter)
  - $k=2$: labelled line code (neuron matters maximally)
Preparation

- Recordings from primary visual cortex (V1) of macaque monkey

- Multineuronal recording via tetrodes – ensures neurons are neighbors (ca. 100 microns)
The stimulus set: a cyclic domain

16 kinds of stimuli in the full stimulus set
Spatial phase coding in two simple cells
Summary: Applying Information Theory to Neurophysiological Data

• It is challenging but approachable
• There is a role for multiple methodologies
  – Methodologies differ in implicit or explicit assumptions about how the meaningful structure of the response space
  – The way that information estimates depend on this partitioning provides insight into neural coding
• It can provide some biological insights
  – Using $I$ as a nonparametric measure of association
    • Which features of neural activity carry information?
    • How much information do they carry?
  – Using $I$ because of the Data Processing Inequality
    • What kinds of neural codes can be ruled out? (Jacobs et al. 2008)
Temporal coding of taste

DiLorenzo et al., 2009

It’s not just the envelope!
Temporal pattern supports full representation of the 4 primary tastes and their 6 mixtures in a logical arrangement.

P. DiLorenzo et al., 2009
What does spike pattern contribute?

Response space geometry is determined by rate envelope, but temporal pattern contributes to discrimination.

- Actual responses
- Shuffled responses
- Inhomogeneous Poisson exchange resampled

50 spikes per sec 5 sec

Salty  bitter  Sour  Sweet

Reconstructed response spaces
Conclusions

• Understanding how neurons represent information is intrinsically both a mathematical and experimental question
  – A focus on the structure of the topology of spike trains
  – Comparison of different information estimates provides insight

• Temporal firing patterns can represent a multidimensional sensory domain
  – In V1 and V2, different submodalities (contrast, orientation, texture) are temporally multiplexed
  – In broadly-tuned neurons of the NTS, the complete classical “taste space” is represented by temporal pattern
  – Firing patterns augment the add to the information contained in the rate envelope

• Neuronal activity is not “intended” to be averaged over time, but to be decoded spike by spike.

The un-answereded question

• Of the statistical features available for coding, what is actually used?