Chapter 3. Analysis of Individual Spike Trains

Action potentials, or more simply put, spikes, are the main units of information transfer in mammalian brains, because spikes can propagate down the axon away from one neuron's cell body to impact other neurons that may be far away. The rapid transient deflection of the membrane potential that comprises a spike can be detectable by an electrode inserted into the surrounding neural tissue. Such measurements of spikes, carried out *in vivo,* provided our first insights into how neurons in the brains of living mammals respond to sensory stimuli. In this chapter, we will ignore the particular dynamics of the membrane potential that generate a spike and the manner in which spikes are detected, but simply treat the spikes of each cell according to the discrete points in time at which they occur.

Responses of single neurons

Receptive fields

A receptive field is a description of the stimulus, or stimuli, that generate a vigorous response in a neuron. The clearest response of a neuron to a stimulus—and often the only one considered—is a change in its firing rate. In the same way that we saw the number of spikes per second produced by a neuron changes if its input current changes, at the simplest level of description all neural responses correspond to changes in their firing rates. These firing-rate changes are brought about by stimulus-dependent changes in the neurons' input currents.

The standard procedure for finding receptive fields is to present an animal with many different stimuli and measure the neuron's firing rate as a function of stimulus. In the simplest method one simply counts the number of spikes produced by a neuron while a stimulus is presented and divides by the duration of stimulus presentation to obtain the neuron's average firing rate.

When a parameter of the stimulus—such as spatial location of a visual stimulus, pitch of an auditory stimulus, frequency of a vibrational stimulus— can be altered in a graded manner, it is straightforward to plot the neuron's firing rate as a function of that parameter. The resulting graph is a plot of the neuron's *tuning curve*. The receptive field corresponds to the range of stimuli for which the firing rate is significantly different from its rate in the absence of stimulus. The tuning curve then describes the properties or characteristics of the neuron's receptive field.

In Figure 3.1, some examples of receptive field are shown. In Figure 3.1A the receptive field is a spatial area, indicating where on the skin surface a touch can produce a response in a particular neuron. This is an example of a standard use of receptive field, indicating the shape and size of the region of locations that produce a response. Indeed, the first use of the term receptive field, by Sherrington in 1906, referred to the regions of body surface that produced a reflexive movement mediated by neural activity[1].

For visual stimuli, the location is typically referred to in visual space, *i. e.*, angular coordinates centered on the direction of eye gaze. For auditory stimuli, the location more often refers to location on the frequency spectrum (*i.e.*, the pitch) of a stimulus rather than its physical location. These differences arise because the initial sensory cells of vision on

the retina respond to light from a particular location relative to eye position[2], while the initial sensory cells of audition in the cochlear respond to sound of a particular frequency[3].

A more complex use of receptive field is shown in Figure 3.1B, indicating the *type* of stimulus that produces a response, as well as its location. In this example, the basic structure of the receptive field shows where a positive neural response is produced by a lighter than background stimulus and by a darker than background stimulus, in white and black respectively.

The center-surround organization of Figures 2.1B1 and 2.1B2 is likely due to lateral inhibition, which is a common feature of neural circuits[4-8]. Lateral inhibition causes a neuron to be inhibited by neurons with different, but perhaps similar, receptive fields. Therefore, a neuron receiving lateral inhibition responds most strongly when it receives direct input from the "center" of its receptive field **and** when those neurons, which would respond most to its "surround", do not receive input and so remain inactive.

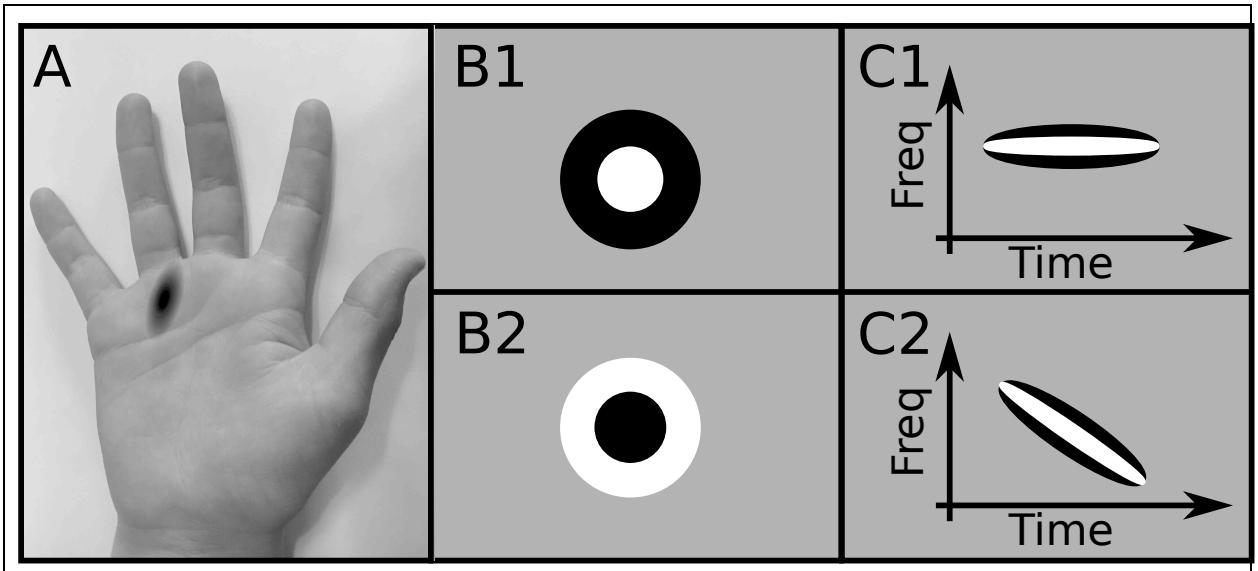


Figure 3.1. Sketches of receptive fields and their general structure. A) A neuron in primary somatosensory cortex is most active when a particular area of skin is touched (marked dark). **B)** A ganglion cell in the retina responds best to a center-surround arrangement of visual stimulus, either light center with dark surround (upper panel, 'On-center') or a dark center with a light surround (lower panel, 'Off-center'). White = light. Black = dark. **C)** The spectro-temporal receptive field of a neuron in auditory cortex can indicate a response to a particular, fixed, frequency of a tone (upper panel) or to a sweep of frequency (a downward sweep in the lower panel). The response is accentuated if

there is no sound intensity at surrounding frequencies. White = high intensity. Dark = low intensity.

Figure 3.1C includes a further complexity, namely the dynamics of a stimulus that most readily causes a neural response. In particular, the lower panel, C2, indicates the receptive field of a neuron that responds most vigorously when a tone starts at a high frequency then sweeps down to a lower frequency over a brief period of time. In Tutorial 3.1 you will produce and characterize a similar spatio-temporal receptive field (see also Figure 3.5).

As we consider neurons increasingly further removed from sensory receptors in the periphery there is a trend of increasing size and complexity of receptive fields. Moreover, neural responses become more and more dependent on context and the history of prior stimuli. Therefore, the concept of receptive field is only applicable for neurons near the periphery where there is a chance that the ensemble of potential stimuli can be suitably tested. For example, if a neuron responds most to a dark vertical bar at a particular location of the visual scene, it is possible to discover this by placing bars of different orientations in different positions until an increase in firing rate is detected. For such simple receptive fields, a bar partially within the receptive field or a bar at a near-optimal orientation would also produce a neural response, so one can home in on the receptive field and find the stimuli that produce the strongest response.

However, if a neuron happens to be most active whenever I look down my street toward my home, the odds of this response being found in a systematic stimulus-focused manner are miniscule (given a billion possible homes and a myriad of possible features other than a home that could be tried). Rather, knowledge of what is important to the subject would need to play a role.

Interestingly, fortuitousness played a role for Hubel and Wiesel in the early measurements of receptive fields in the primary visual cortex of cats[9,10]. The accidental movement of a notepad across a projector screen caused a dramatic burst of neural activity in the anaesthetized animals. The two scientists later received the Nobel Prize in 1981 for their work, in which they described such responses that were sensitive to the orientation of moving edges or slits.

Neurons a few synapses or more from the periphery—such as those in the cerebral cortex—are not driven primarily by direct sensory input. Even neurons in primary sensory cortex receive far more connections from other neurons in the cortex than from neurons preceding it in the sensory pathway. While feedforward sensory responses dominate studies of neural responses in anaesthetized animals, it is only because the stronger feedback input to cortical cells is silenced by anesthesia. Therefore, care is needed when interpreting the results of experiments using methods—such as anesthesia—that simplify the behavior of a neural circuit. Such simplification may provide the interpretable, constraining data that is necessary in the early stages of discovery to put together the building blocks of a circuit. However, the simplified situation may provide only a small piece of any description of that neuron's role in natural behavior.

Time-varying responses and the peri-stimulus time histogram (PSTH)

In vivo neural responses are highly variable—if an identical stimulus is presented multiple times (*i. e.* on multiple trials), both the timing of spikes with respect to stimulus onset and

the total number of spikes change from trial to trial. Therefore, it is typical to average across trials simply to get a better estimate of a neuron's mean response to a stimulus. Moreover, the averaging of spike trains across trials can allow us to uncover the dynamics of a neuron's response to a stimulus, not just the mean number of spikes. The peri-stimulus time histogram (PSTH) is the principal method for such averaging.

The PSTH estimates the dynamics of the response by aligning spike trains to the time of stimulus onset then averaging across trials to obtain an estimate of the mean firing rate as a function of time. We might ask whether the mean response across trials is at all relevant beyond being descriptive, since the spikes on each individual trial can only impact the animal's behavior on that trial. One argument that extends the relevance of across-trial averaging is that we are measuring one neuron out of many, perhaps hundreds, of a population with a similar trial-averaged response. The trial-averaged response could be reproduced on each individual trial by the average on that trial measured across the population—and it is the population's activity that impacts other neurons and the animal's behavior. If neurons were uncorrelated on a trial-by-trial basis, this would be a good argument, but as we will discuss in Chapter 8, more sophisticated methods should be applied when correlations are important.

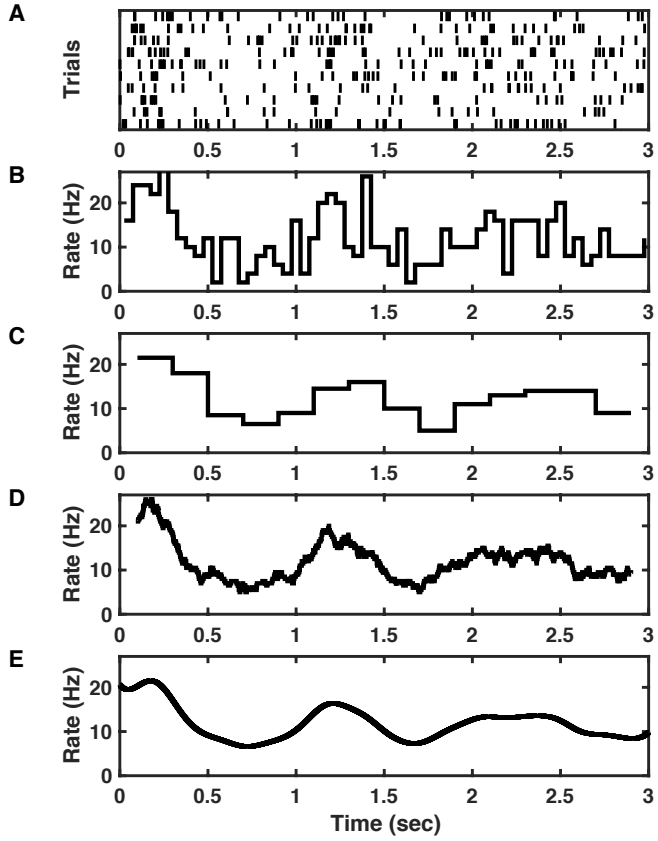


Figure 3.2. PSTH for an example of oscillating activity. A) Each of the 10 rows contains the series of spike times from a single neuron in one trial. Spikes are produced as an inhomogeneous Poisson process (see text or code) with the firing rate in Hz as $r(t) = 12 + 6\sin(2\pi t)$ where t is time in seconds. **B)** Spikes are accumulated across trials in consecutive time-bins of width 50ms. **C)** Spikes are accumulated across trials in consecutive time-bins of width 200ms. **D)** Spikes are accumulated across trials in a sliding bin of width 100ms. **E)** Each spike is filtered (smoothed) with a Gaussian of width 100ms to produce a continuous, smooth function. In this example, because the rate is itself defined as a smooth function, the smoothed spike train produces the best approximation to the underlying rate. This figure was created by the available online code `PSTH_osc.m`.

Two distinct methods can be used to combine across trials the temporal information associated with each spike. In the most straightforward method, time is binned into small windows, then the number of spikes in each time-window is counted and averaged across trials (see Figure 3.2B-C). The resulting PSTH is a series of adjacent rectangles whose height indicates the mean firing rate with an accuracy that depends on the number of spikes contributing to each time-bin. Therefore, there is a trade-off, with larger time-bins containing more spikes so producing a more accurate estimate of the mean rate, but at the same time the larger time-bin has less sensitivity to time-dependent variation of the neuron's activity.

In the standard binning method, all information regarding the spikes' relative position in the bin is lost. One might think that a spike falling near the boundary of a bin could contribute equally to the next bin to avoid such loss of information. Alternative methods aim to overcome such information loss. In a variant of the straightforward method, a sliding window can be used (Figure 3.2D), so that each spike enters a series of time-bins with centers ranging a half of the bin-width before the spike to a half of the bin-width after the spike.

A final alternative is to smooth each spike with a Gaussian filter (see Figure 3.2E for the result and Figure 3.3 for an introduction to linear filters). In this method, each spike is replaced with a Gaussian function of unit area, centered on the spike-time. Such a filter is linear, as the contributions of all spikes are added together linearly. The time resolution can be quite fine as the Gaussian is a continuous function. The width of the Gaussian filter plays a similar role to bin-width in the straightforward method—the wider the Gaussian, the less noisy the resulting PSTH, but the lower the temporal resolution.

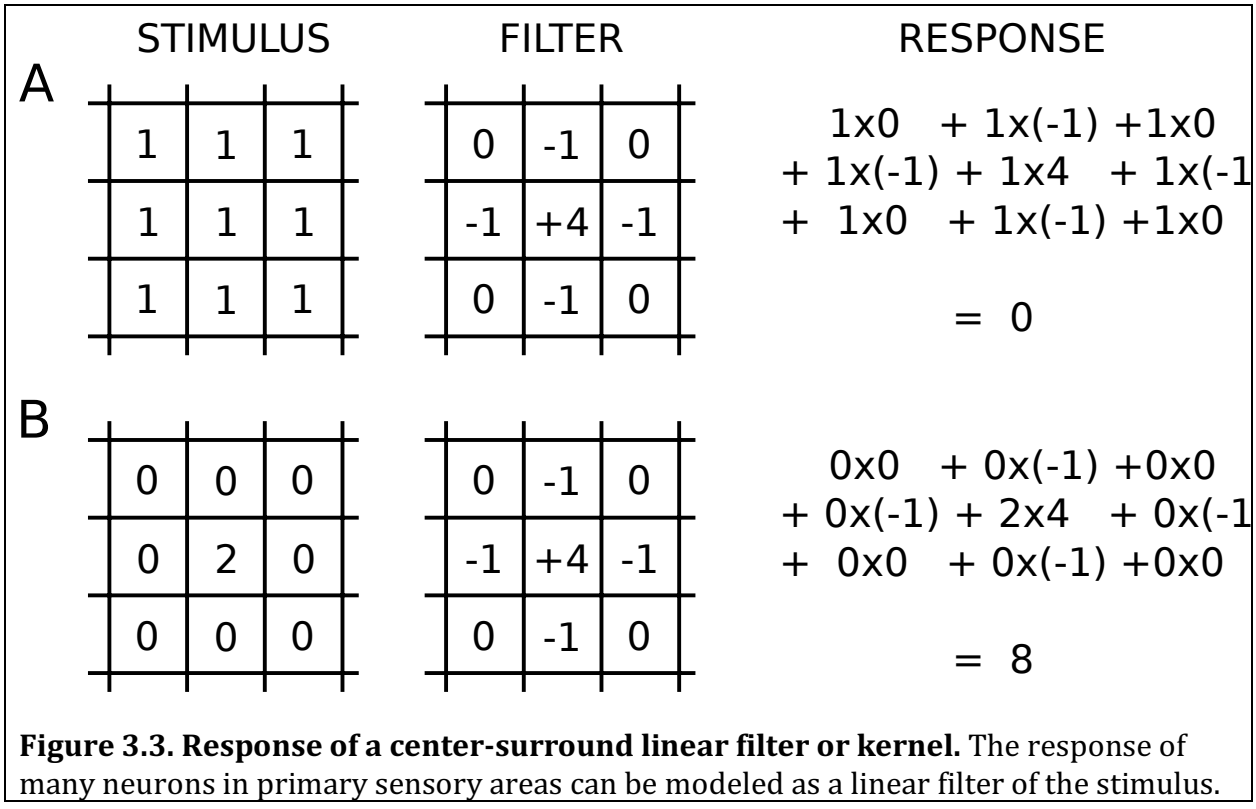


Figure 3.3. Response of a center-surround linear filter or kernel. The response of many neurons in primary sensory areas can be modeled as a linear filter of the stimulus.

In the simplified example shown, the neuron filters a spatial stimulus by multiplying the stimulus amplitude at each spatial location (left) by the corresponding value of the filter—also called the kernel—(center) to produce a response (right). **A.** For the center-surround filter shown (middle) a spatially constant stimulus (left) generates no response (right) because the positive contribution at the center of the filter is canceled by the negative contributions of the surround. **B.** Such a filter produces a strong response to a spatially localized stimulus.

Neurons as linear filters and the linear-nonlinear model

When receptive field structure is indicated in the manner of Figure 3.1B-C, the suggestion is that the neuron is acting as a linear filter. To understand the response of a linear filter, we can consider Figure 3.3, which shows the response of a neuron with a center-surround (On-center) receptive field to two different stimuli. The neuron's response is defined by the filter function—or kernel—shown in the middle column. The value of the stimulus at each point within the receptive field is multiplied by the value of the kernel at that location. The contributions obtained in this manner from each point within the receptive field are then summed together to produce the neuron's response. The filter is called linear, because the neuron's response is a linear combination—*i. e.*, a weighted sum—of all the stimulus values. In the example shown, the linear combination is obtained as a sum over individual discrete stimulus values—more generally the kernel would be a continuous function and the sum would be replaced by an integral.

Just as a linear filter can be used to approximate a neuron's response to the spatial properties of a stimulus, so too can it be used to approximate a neuron's time-varying response to a time-varying stimulus. We will use methods that assume such linear responses in the next section and in Tutorial 3.1.

As we have seen in Chapter 1, neural responses, even in the simplest of models, are not linear. In fact, since a firing rate cannot be negative, it must be a nonlinear function of input for any cell, simply because of the threshold for response (the threshold-nonlinearity). However, a straight-forward extrapolation of the linear filter model, namely the "linear-nonlinear model" has proven to be of great value in describing neural responses[11-13]. In the linear-nonlinear model, a weighted sum of the stimulus of the receptive field, as in Figures 2.1B-C and 2.3, produces the inputs to a model neuron. The model neuron's response is then a non-linear transformation—equivalent to sending the input through an f-I curve—of that linearly weighted input.

Spike-triggered average

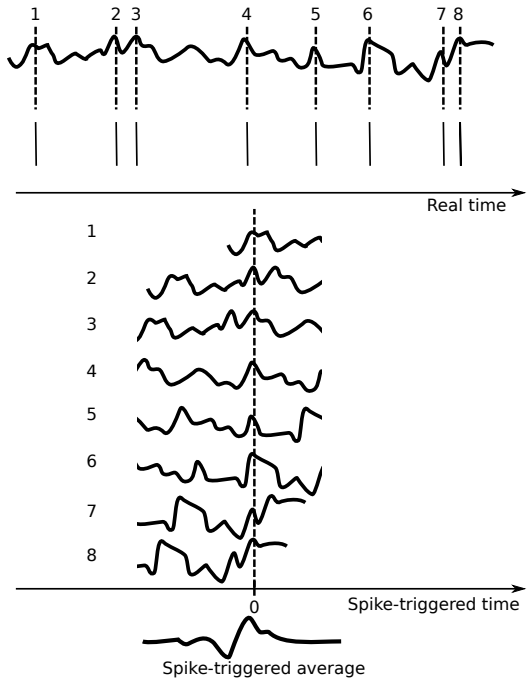


Figure 3.4. Producing a spike-triggered average. A time-varying signal (solid continuous line, top) produces a sequence of 8 spikes (vertical solid lines below the stimulus). The stimulus is then realigned to the time of each spike (vertical dashed lines top and middle) to produce eight samples of the stimulus, one sample around the time of each spike. These samples are averaged to produce the spike-triggered average (STA, bottom). The STA preceding the neuron's spike (set to the time of zero) is the average stimulus to cause a spike. Since the spike cannot cause the stimulus, the period after the spike-time over which the STA decreases to zero is an indication of the time-scale of correlations within the stimulus. Any appreciable decay time—indicating stimulus-correlations do not rapidly decay to zero—is a confound if this method is used to estimate properties of the neuron.

A technique that can be used to characterize simple receptive fields is the spike-triggered average[14]. With this technique, a neuron is recorded while it responds to a multitude of different stimuli. Each time the neuron produces a spike, the stimulus that caused the spike is stored. All of the stored spike-causing stimuli are then averaged together. The process can be carried out for spatial receptive fields, in which case each stimulus is maintained long enough to ascribe each spike to the correct spatially patterned stimulus—for example, in visual processing it takes a few tens of milliseconds for a stimulus to produce a spike in V1).

More commonly, the temporal properties of neural responses are also considered. The sequences of stimuli preceding each spike for 100-200ms are then stored. As a useful control, those stimuli following each spike can also be stored. When the stored sequences of temporal stimuli are averaged together, they should be aligned to the time of each spike. The same stimulus can contribute at multiple time-points in such a time-dependent spike-triggered average. For example, if two spikes occur, one 50ms after the stimulus and the other 100ms after the stimulus, then that stimulus contributes to the −50ms and to the −100ms bins of the spike-triggered average (see Figure 3.4).

White-noise stimuli for receptive field generation

As suggested earlier in this chapter, when characterizing a neuron by its receptive field, great care is needed when designing the set of stimuli. At the simplest level, it is important

to realize that any correlation included in the stimuli will be found to be present in a receptive field.

In the early days of neural networks, one network was trained to identify tanks in a pattern recognition test. The network was astoundingly successful, beyond what seemed reasonable given the subtle features, such as the end of a tank's gun turret, present in some of the pictures. On further analysis, it was found that pictures of tanks were taken on one day, whereas pictures without tanks were taken on a separate day. Differences reflecting changes in the weather, such as brightness of the sky, could be used to distinguish tank from non-tank. Thus, accidentally, one stimulus feature—presence of a tank—was completely correlated with another stimulus feature—the weather pattern. If the output of the neural network is thought of as a neuron, then its trained response to sky brightness appeared like a response to a tank, because of the unfortunate correlation.

The type of stimulus that avoids any such confounding correlations is white noise[15]. Production of white noise stimuli is limited by the resolution of the projecting apparatus—light intensity is always correlated on the length scale of a pixel and on the time scale of a frame. However, the 'white noise' stimuli used to measure receptive fields are typically discretized at a much coarser level than these limits, with correlations over the range of many pixels, so that there is sufficient chance of generating a response. Use of stimuli with correlations limits the resolution of any receptive field produced, but ensures the stimuli produce a sufficient number of responses before averaging, as explained below.

Consider a neuron that responds best to illumination covering a circle of radius one degree of the visual field. If a stimulus contains randomly chosen luminance over each pixel covering one-thousandth of a degree, the number of independently chosen luminances within the neuron's receptive field would be on the order of one million. These one million independent values would have to be correlated in the right way to generate a significant response. The likelihood of sufficient correlation by chance would be so small that the number of such stimuli needed to map the receptive field would be prohibitively large.

Indeed, one can estimate how likely it is that any one combination of a million independent stimuli would produce 1% greater total input to a neuron. For example, if each of a million pixels are equally likely to be "on" or "off", the mean number "on" is 500,000. If the cell responds when 505,000 are "on" (a 1% increase in input) the chances of this occurring in any one image turn out to be 2×10^{-45}, an astronomically small probability! (This number is arrived at by using the Binomial distribution to get the standard deviation as $\sqrt{Np(1-p)} = \sqrt{10^6 \cdot \frac{1}{2} \cdot \frac{1}{2}} = 500$ then using the Normal distribution to find the chances of being 10 standard deviations or more from the mean). If such stimuli were presented at the rate of one every 10ms then there would be negligible chance of seeing a response even if the experiment lasted for the lifetime of the universe!

The problem here is the mathematically equivalent to the combinatorial problem when using stimulus-driven statistics to determine the response of a complex neuron a priori—the probability of finding a stimulus that can drive a cell using random features alone becomes astronomically small. More progress can be made by considering the types of stimuli that are relevant to an organism based on its lifetime of experiences. Alternatively,

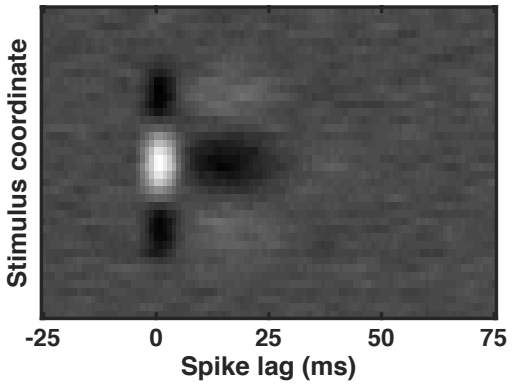


Figure 3.5. Model of a spatio-temporal receptive field of a V1 simple cell. A single spatial dimension is plotted (y-axis), corresponding to the location of a bar of optimal orientation, combined with a time dimension (x-axis). The center-surround spatial response combines with a biphasic temporal response. Light indicates high positive stimulus, dark indicates negative stimulus. This figure was generated by the online code `spatial_AELIF_randomI.m`.

in many studies, the relevant stimuli are imposed on the animal over the course of several months of repetitive experiments. This latter procedure enhances the likelihood of the scientist finding a neuron that responds to the experimentally relevant stimulus—though at the risk of producing responses that are not found in an animal's natural setting.

Spatio-temporal receptive fields

Receptive fields were first described for tactile stimuli, to indicate which regions of an animal's skin when touched would produce a response in a given neuron (Fig. 3.1A). Today, visual and auditory stimuli are the most commonly used to measure receptive fields (Figs. 3.1B,C). In all modalities, the dynamics of a stimulus is important. Auditory stimuli are inherently dynamic, even after transformation to the spectrum of pitch (frequency) versus time, because we rarely hear a constant pitch. For visual stimuli, any change over time indicates movement, which is of great importance to an animal. In fact, objects may be invisible within a scene until they move as a coherent whole and immediately pop out to our attention. Thus, it should not be a surprise that many neural responses depend on the dynamics of a stimulus.

We have already observed neural responses that depend on stimulus dynamics when we considered spike-rate adaptation in Chapter 1. At its extreme, the combined effect of several such adaptation processes can render a neuron insensitive to a constant stimulus and only responsive to stimulus changes. Such an effect would appear in the spike-triggered average as a biphasic response—the neuron is most likely to spike when a "positive" stimulus follows a period of below-average stimulus. We will investigate an example of this in Tutorial 3.1.

When the spatial and temporal components are combined, the resulting spatio-temporal structure of the receptive field of a neuron can be plotted as a colormap, or in grayscale (as in Figure 3.5), or as a contour plot in two dimensions.

Table 3.1. Structure of code for Tutorial 3.1, Part A.

`Main Code`

```
   1. Generate stimulus vector.
      Simulate vector of spike times.
   2. Down-sample stimulus and spike-time vectors using the
      function expandbin.
   3. Send down-sampled vectors to the function STA.
   4. Plot the spike-triggered average returned from STA.

function expandbin
   1. Define a new_vector of length new_dt/old_dt times the
      length of old_vector.
   2. Find the mean of sets of points of length new_dt/old_dt in
      old_vector and add them to a single point in new_vector.


function STA
   1. Define a time-window vector.
   2. Initialize to zero the sta vector.
   3. Loop through all spikes adding to the sta vector the
      stimulus offset by the time of each spike.
   4. Divide the sta vector by the number of spikes.
```

Tutorial 3.1: Generating receptive fields with spike-triggered averages.

Neuroscience goal: analysis of receptive fields using the spike-triggered average

Computational goals: using a function; collapsing or expanding data into sets with different widths for time-bins.

In this tutorial, you will simulate an AELIF neuron, following the dynamics described in Tutorial 2.3 Q.2, and using the following parameters: $E_L = -60\text{mV}$, $V_{th} = -50\text{mV}$, $V_{reset} = -80\text{mV}$, $\Delta_{th} = 2\text{mV}$, $G_L = 8\text{nS}$, $C_m = 100\text{pF}$, $a = 10\text{nS}$, $b = 0.5\text{nA}$ and $\tau_{SRA} = 50\text{ms}$. In both parts of this tutorial the goal is to use the spike-triggered average to assess what time of stimulus is best at producing spikes in the model neuron.

Part A: Time-varying stimulus.

In Part A, we will treat the stimulus as a single input current that varies over time and assess what temporal variation of the stimulus is most likely to lead to a spike in the model neuron. To achieve this, we will use a stimulus that is held constant for 5ms then randomly replaced with a new stimulus.

1) The first goal is to simulate the neuron and record the spike times.

a) Produce a vector of 40,000 values for the applied current, with each value chosen randomly from a uniform distribution between -0.5nA and +0.5nA. These values will be used to produce successive 5ms blocks with the current fixed in each 5ms block.

b) Create a time vector in steps of 0.02ms up to a total time of 40,000 × 5ms and an applied current vector of the same size as the time vector. The current vector should contain the

40,000 randomly generated values from 1a), with each value repeated for 250 time-steps, so that the applied current is constant for 5ms before changing to a new value.

c) Simulate the AELIF neuron with the applied current vector from 1b) using the parameters above and the equations for Tutorial 2.3, Q.2. Use a simulation time step, $\Delta t = 0.02$ms. Ensure all spikes are recorded in a vector of the size of the time vector with 1 in each time-bin that a spike occurs and zeros otherwise.

2) To improve the efficiency of the analysis we will *down-sample* the stimulus and spike vector (see Figure 3.6). This is best achieved by writing a function that we will call "`expandbin`". The function will take as inputs an initial vector, an initial bin-width, and a final bin-width. It will return an output vector of smaller size than the initial vector, with the size reduced by a factor equal to the ratio of the bin-widths. In particular, in this tutorial, we are simulating the neuron with a time-bin of 0.02ms but are only interested in changes on a time-scale of 1ms or more (and our applied current only changes on a 5ms time-scale). So, we want to replace our stimulus vector and spike vector with vectors that contain a value for each 1ms of time rather than for each 0.02ms. This requires down-sampling by a factor of 50.

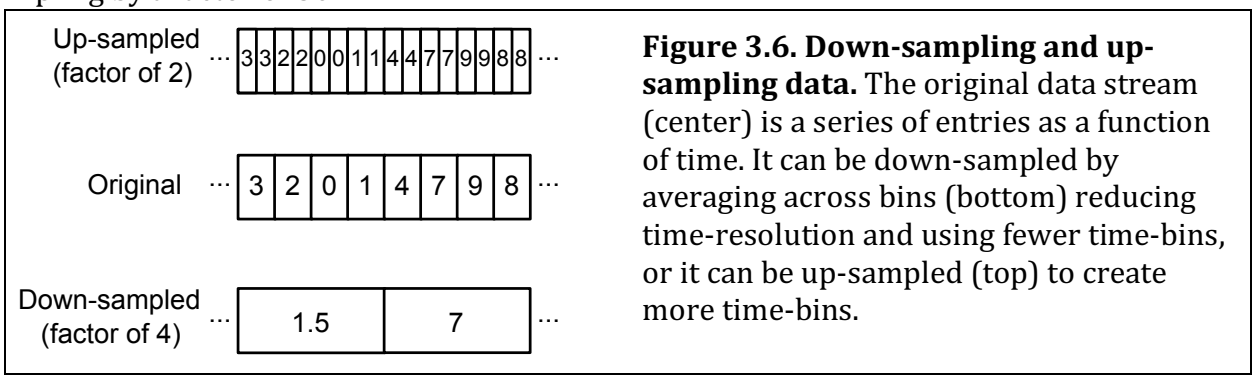


Figure 3.6. Down-sampling and up-sampling data. The original data stream (center) is a series of entries as a function of time. It can be down-sampled by averaging across bins (bottom) reducing time-resolution and using fewer time-bins, or it can be up-sampled (top) to create more time-bins.

a)i) Create a function, `expandbin`, with inputs and outputs as follows:

```
function [new_vector] = expandbin(old_vector, old_dt, new_dt)
```

ii) Within the function (that will be saved in the file "`expandbin.m`") find the length of `old_vector` then the scaling ratio of `new_dt` divided by `old_dt` calculated to the nearest integer.

iii) Calculate the size of `new_vector` by dividing the size of `old_vector` by the scaling ratio. Define `new_vector` as a set of zeros of this size.

iv) For successive elements of `new_vector` calculate the mean of successive blocks of entries of `old_vector`. The length of each block of entries is the scaling factor. In this tutorial, the mean of a block of 50 of the 0.02ms time-bins in `old_vector` will be placed into a 1ms time-bin of `new_vector`.

b) Send to the function, `expandbin`, which you have just created, the applied current vector used to update the membrane potential in the simulations with its time-step of `dt`, (note that while your values of applied current change every 5ms, the vector used in the simulations should have a value every `dt` and the latter is used here) along with a variable

`new_dt` defined as 1ms. The function will return down-sampled versions of this vector. Repeat by sending the simulated vector of spike times to the function, `expandbin`. Since spikes are defined as binary, 0 or 1, you should then change all the reduced spike values in the returned vector back to 1.

3) You will produce and plot the spike-triggered average derived from the two down-sampled vectors (the stimulus and the spike times). You will create and use another function, `STA`, to achieve this.

a)(i) Create the function, `STA`, with outputs and inputs in the following format:

```
function [sta, tcorr] = STA(Iapp, spikes, dt, tminus, tplus)
```

where the inputs are respectively: applied current vector; vector of spike times, with 1 at the time of every spike; the time-step used in these vectors; the time of the stimulus before a spike to begin recording; and the time of the stimulus after a spike to stop recording. Note that the vectors `Iapp` and `spikes` must be the same size.

(ii) It is useful in a function to supply default values that are used if a variable is not passed to it. If only three variables are passed to the function `STA` then the last two will not exist within the function. In this case, `tminus` and `tplus` define the time window over which the spike-triggered average is calculated. Within the function, they can be created if they do not already exist using a conditional statement and the in-built function "`exist`":

```
if (~exist('tminus'))
```

Note that the tilde, "`~`", means "not", so the line of code following the conditional is only executed if the variable "`tminus`" has not yet been defined.

Include two such conditional statements to set default values for `tminus` and `tplus` of 75ms and 25ms respectively.

(iii) Produce integer variables `nminus` and `nplus` that correspond to the number of time bins before and after a spike over which the stimulus is recorded for averaging.

(iv) Define the time window, `tcorr`, as a vector spanning from "`-nminus*dt`" to "`nplus*dt`" in steps of "`dt`". This will provide the x-coordinates when the spike-triggered average is plotted.

(v) Initialize the variable `sta` to be zero in a vector the same size as `tcorr`. This will accumulate values to provide the y-coordinates when the spike-triggered average is plotted.

(vi) Use "`find`" to find the time bins that contain spikes.

(vii) Loop over all the spike-times found in (vi), for each spike defining a window that begins a number of bins, `nminus`, before the spike and ends a number of bins, `nplus`, after the spike. Ensure the window neither begins earlier than the stimulus nor ends after the stimulus is over (you can ignore the spike in such cases).

(viii) Within the loop add to the vector `sta` the values of the stimulus vector within the window you have just defined for the spike (as in Figure 3.4).

(ix) After the loop, to calculate the average stimulus, just divide `sta` by the total number of spikes.

Note: It is easiest just to ignore those spikes too near the beginning or the end of the stimulus. However, if the time window for the STA is not much smaller than the stimulus duration it is better if they are not ignored (to avoid loss of a large fraction of the data). In that case, care must be done when averaging the stimulus: values in each time bin of `sta` should be divided only by the number of spikes that can contribute to that time bin, rather than the total number of spikes. The value of the stimulus before and after the end of the trial is assumed to be zero, so for example, a spike very early in the trial would contribute a lot of unwanted zeros to the average stimulus at earlier times.

b) Now send the down-sampled applied current vector and spike train to the function `STA`, which will return the spike-triggered average and the time-window.

c) Plot the spike-triggered average as a function of the time-window around the stimulus. To adopt conventional coordinates, the x-axis should be the negative of the time-window, so as to represent the time-lag of the spike after the stimulus that produced it. With this transformation, the STA at positive values of x denotes a causal relationship.

d) Comment on how any specific parameters of the AELIF model that you have simulated helped to determine any features of the spike-triggered average in c).

OPTIONAL CHALLENGE, Part B: Spatio-temporal stimulus.

We now extend the tutorial to include a stimulus that varies in both space and time. We will also assume that the neuron's input current is a linear combination of the values of the stimulus at each spatial location. (A linear combination is a weighted sum where the weights can be negative.) Therefore, the neuron receives a time-varying input current as in Part A, but at each point in time the value of the input current depends on many values of the spatially varying stimulus. It is worth noting that while we discuss the additional coordinate of the stimulus as being spatial, corresponding to a visual stimulus, the mathematics and the associated computer code are identical if the stimulus coordinate is the frequency of an auditory stimulus.

1a) Generate an array of 40 rows by 40,000 columns to contain the stimulus, $S(x,t)$, with each entry randomly chosen from a uniform distribution between -0.5nA and 0.5nA.

b) Assume the row number corresponds to a spatial coordinate, x, (with $x_{max} = 40$) and that the input current to the cell, $I_{app}(t)$, is given by the weighted sum of the stimulus at each spatial coordinate according to:

$$I_{app}(t) = \sum_{x=1}^{x_{max}} W(x)S(x,t) \text{ with } W(x)=\cos\left[4\pi\left(\frac{x-x_0}{x_{max}}\right)\right] e^{-10\left(\frac{x-x_0}{x_{max}}\right)^2}$$

where $x_0 = 20.5$ is at the center of the spatial stimulus. Plot the input weight vector, $W(x)$ (which must be a 1x40 row vector) and generate the complete time-dependent input current as a row vector by direct matrix multiplication of the weight vector by the stimulus array.

c) Simulate the AELIF cell following the instructions of Part A, except with two of the parameters altered: $a = 40$nS, $b = 1$nA; and with the new input current. As usual, record the times of spikes as 1 in an array that otherwise contains zeros.

d) Down-sample the spikes array and "up-sample" the stimulus array to produce new, smaller arrays. Care must be taken with the stimulus array, as you must either carry out the up-sampling one row at a time, or you must adapt the function, `expandbin`, you created in Part A to handle arrays of greater than one dimension. While down-sampling involves taking the mean of values from many bins to enter into one new bin, up-sampling involves taking the value from one old bin and repeatedly entering it into many new bins.

2) Generate a new function, `STA_spatial`, by adapting the function `STA`. The new function will take a two-dimensional array (like $S(x, t)$) as input and return a two-dimensional array as the spike-triggered average, `sta`.

Hint: a few changes are needed to make the conversion, but you might want to use a command like `[Nspace, Nt] = size(stim_array)` to extract the number of spatial and temporal bins and to be careful when using "`mean`" to take the mean of a block of columns (since time varies from column to column) when down-sampling rather than the mean of a block of rows.

3) Use your newly created function to calculate the spatio-temporal receptive field. Use the command `imagesc(fliplr(sta))` to visualize the receptive field in the conventional manner. (The command `fliplr` reverses the order of columns, *i. e.* flips the array from left to right).

4a) Plot rows 12, 20, and 28 of the receptive field. How do they compare?

b) Plot columns 25, 50, and 75 of the receptive field. How do they compare?

Spike-train statistics

If a constant or regularly varying input current is provided to a neuron, either *in vitro,* or in a model (*in silico),* any resulting activity of the neuron is itself regular except in rare cases. Yet the activity of neurons in the brains of wakeful animals is highly irregular. The traditional solution in neural models is to add noise—in the sense of random fluctuations—to the input current or conductance. However, in a real circuit, these random fluctuations in a neuron's input are largely due to the irregularity of spike times of other neurons receiving similar random fluctuations in their inputs. A constraint arises then, when understanding the behavior of neural circuits, because the observed irregularity in spike trains must be able to account for the noisy inputs to cells that can generate such irregularity. Therefore, it can be as important to measure the irregularity of spike trains—*i. e.*, at least the second-order spike-train statistics—as it is to measure a neuron's mean response.

Coefficient of Variation (CV) of inter-spike intervals

The coefficient of variation of any set of values is the standard deviation divided by the mean of the values. In the case of measured inter-spike intervals (ISIs), a highly regular spike train has ISIs of nearly the same value, so the set of ISIs has very small standard deviation. Therefore, the CV of a regular spike train is close to zero. Spike trains of cortical neurons measured in wakeful mammals typically have CVs near one or greater than one.

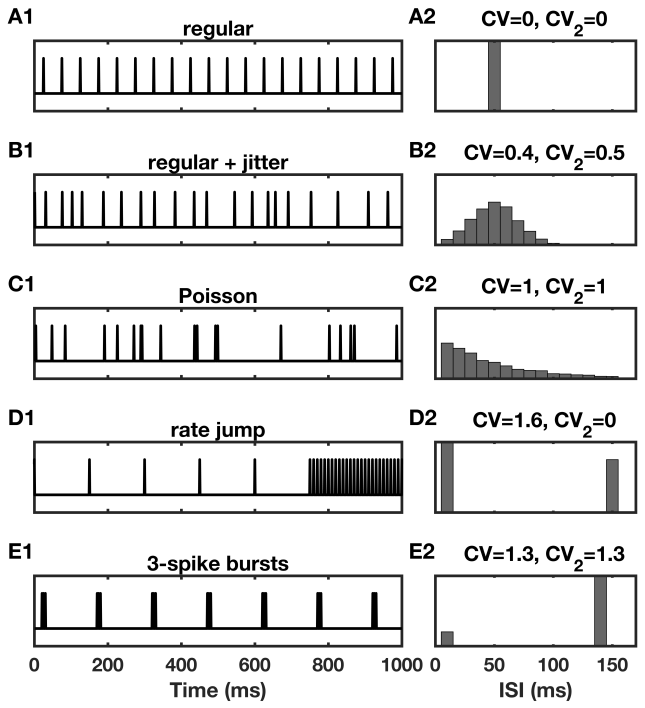


Figure 3.7. Spike-trains, ISI distributions, and statistics. A. A regular spike train with a fixed ISI of 50ms. **B.** A spike train with ISIs arising randomly from a normal distribution with mean 50ms and standard deviation of 20ms. **C.** A Poisson spike train with mean ISI of 50ms. **D.** A regular spike train with ISIs of 150ms switches to a regular spike train with ISIs of 10ms. **E.** A spike train with bursts of 3 spikes, with 150ms between bursts and 4 or 5 ms between spikes within a burst. **Left, A1-E1.** Example of 1 sec of each spike train. **Right, A2-E2.** Histograms show the distribution of ISIs generated by a 300-sec simulation. CV and CV_2 of each distribution is given above each example. This figure

was created by the online code `spikes_cv_isis.m`.

The CV is directly related to the histogram of the ISIs. The full histogram of ISIs contains more information than the CV alone—which is one number that depends simply on its width and the position of its mean—so it can be better to plot or fit with a curve the full distribution of ISIs. For irregular spike trains, the distribution of ISIs can be shaped like an exponential decay, albeit with a small dip near an ISI of zero corresponding to the refractory period. In fact, the exponential ISI distribution has a CV of one and is a feature of the Poisson process (to be described below), so the Poisson process is sometimes used to model individual spike trains.

Since the CV depends on all of the spike intervals measured over a period of time, a high CV does not imply a highly irregular spike train (although an irregular spike train will produce a high CV, see Figure 3.7). A high CV, meaning a large range of inter-spike intervals relative to their mean, can arise because the neuron's firing rate is changing over time, either slowly, or rapidly but periodically, as it does for regularly bursting neurons. Therefore, an improved measure of irregularity was proposed, based on the similarity between neighboring ISIs. Since only two ISIs were compared at a time, the measure is called CV_2. If there is no time-dependence, so that the value of one ISI has no correlation with the value of the previous one—*i. e.*, each ISI is a random sample from the complete distribution of ISIs—then CV_2 is the same as the CV. However, if the neuron's firing rate varies slowly, on a timescale longer than typical ISIs, then CV_2 will be smaller than the CV and CV_2 will better correspond to the level of irregularity.

Of course, CV_2 is itself a poor measure of irregularity when, for example, a neuron fires regular doublets of spikes with large separations between two closely spaced spikes. Such doublet firing is an example of bursting, producing an ISI distribution that is bimodal (contains two peaks, Figure 3.7E). For bursting neurons, the ISIs within bursts should be analyzed separately from the times between bursts (*i. e.*, the inter-burst intervals).

Fano factor

The Fano factor is another measure of variability, in this case one that considers the total number of spikes in a given time interval instead of the inter-spike intervals. It is defined as the variance of the number of spikes, $N(T)$, in a time bin, T, divided by the mean:

$$F(T) = \frac{\mathrm{Var}[N(T)]}{\overline{N(T)}} = \frac{\overline{N^2(T)} - \left[\overline{N(T)}\right]^2}{\overline{N(T)}} \qquad \text{Eq. 3.1}$$

where the overline denotes the mean of a quantity.

As with the CV, the Fano factor is increased by fluctuations that cause irregular timings of spikes and by variation in the underlying firing rate. The Fano factor can be calculated from a single series of consecutive time bins, in which case one number is produced that may depend on the chosen size of time bins, so can be plotted as a function of this size. Alternatively, time-bins can be aligned across trials to a specific time in the trial. In the latter case the Fano factor can vary both with bin-size and as a function of time during the trial. Such time-dependence of the Fano factor can indicate when, in a task, neural activity

is particularly constrained and well-defined, so can suggest "resets" or the presence of fixed activity states that may be important steps during information processing.

A Poisson process with constant firing rate (called a homogeneous Poisson process, described in the next subsection) has a Fano factor of one, independent of the size of time-bin used (see Appendix A). If the firing rate varies within each trial, but in an identical manner on each trial, then the Fano factor in each time-bin can still be calculated by comparing time-bins across trials with the same underlying firing rate. In this case, a Poisson process (now called inhomogeneous) still has a Fano factor of 1, even though the CV of ISIs would be greater than one because of the within-trial variation.

The homogeneous Poisson process—a random point process for artificial spike trains

At times, we may need to simulate a series of spikes that has the gross features of an *in vivo* spike train—a given mean rate that may vary in time and a similar irregularity of spike times—and not wish to take the time to adjust parameters in a model neuron until it reproduces these features. A good reason to need such simulated spike trains is to compare the results of analyses on such control or "dummy" datasets with the results of identical analyses carried out on datasets acquired *in vivo*.

A simulated spike train is an example of a point process, which means a process whose contents are entirely represented by a set of discrete points—in this case the spike times. The point process is in continuous time, as the discrete spike times are instantaneous values that could arise at anywhere in a range of values of the continuous variable, time. Here we focus on the Poisson process, which is an example of such a continuous-time point process, in part because its statistics are similar to those of neurons measured *in vivo*, and in part because it is the simplest such process. Therefore, it is a good spring board for those who want to delve more deeply into point processes at a later date.

A Poisson process is defined by the probability of an emission in any small time-interval, δt. The probability is proportional to the length of the small time-interval, so can be written as $r \cdot \delta t$. For a homogeneous Poisson process the proportionality constant, r, which corresponds to the firing rate of a neuron, is fixed across time. The single value of r is the only parameter needed to fully define the homogeneous Poisson process.

A key result (derived in Appendix A) for a Poisson process is the probability of a given number, N, of spikes in a time interval, T. For a Poisson process the entire probability distribution, $P[N]$, depends on a single parameter, $\lambda = rT$, via

$$P[N] = \frac{\lambda^N e^{-\lambda}}{N!}. \qquad \text{Eq. 3.2}$$

Equation 2.2 defines the Poisson distribution, with of $\lambda = rT$ being the expectation value for N—the expected mean number of spikes from many examples of the process. We see that equating r with the firing rate of a neuron is justified by this result as the mean number of spikes in a time window is expected to be rate × time.

The distribution defined in Eq. 3.2, may appear to be in conflict with the definition of a Poisson process, which is based on the probability of a single spike in a small time-interval

being $r \cdot \delta t$, and includes no exponential term. However, the definition of a Poisson process assumes the small time-interval, δt, is in the limit that the product $r \cdot \delta t$ is much less than 1. In this case $e^{-r \cdot \delta t} \approx 1$ so that Eq. 3.2 with $N = 1$ can be simplified from $P[1] = \frac{(r \cdot \delta t)^1 e^{-r \cdot \delta t}}{1!}$ to simply $P[1] = r \cdot \delta t$ as required.

From the Poisson distribution, we can produce the distribution of inter-spike intervals for a Poisson process. To calculate this, we need to find the probability that following any spike the next spike is in a small time-interval, from T until $T + \delta t$ later. The dependence of this probability on T is the ISI distribution. It is evaluated by multiplying together the probability that there are no spikes in the interval of size T, by the probability, $r \cdot \delta t$, of a spike in the following small time interval. Substituting $N = 0$ into Eq. 3.2 yields $P[0] = e^{-\lambda} = e^{-rT}$. Therefore, the probability of the inter-spike interval being between T and $T + \delta t$ for a Poisson process is $r \cdot \delta t \cdot e^{-rT}$, or:

$$P(T)\delta t = r \cdot \delta t \cdot e^{-rT}. \qquad \text{Eq. 3.3}$$

A Poisson process of rate r can be simulated most easily by choosing time-bins of width, Δt, much smaller than $1/r$, so that the probability of more than one spike in the interval is negligible, in which case $P[0] \approx 1 - r\Delta t$ and $P[1] \approx r\Delta t$ (respectively the probability of no spikes or a single spike in each time-bin). Such a spike train can be implemented in a single line of code, *e. g.*, in Matlab:

```
spikes = rand(size(tvec)) < rate*dt;
```

where `tvec` is the vector of time-points, and `rate*dt` is the same as $r\Delta t$.

Alternatively, and in a method that can be generalized to other processes with different ISI distributions, successive ISIs can be randomly chosen from the ISI distribution. To achieve this, the cumulative sum of the ISI distribution should be plotted and scaled between 0 and 1 on the y-axis. A random number between 0 and 1 is then chosen as the y-coordinate. The corresponding x-coordinate is read from the graph and used as the ISI. For the Poisson process, with an exponential distribution of ISIs, the cumulative sum becomes an integral, which is also an exponential distribution. The resulting ISI chosen becomes:

```
ISI = -log(rand())/rate;
```

If `rand()` is replaced by `rand(1,Nspikes)` then a set of `Nspikes` spike times is generated from the set of ISIs as

```
spiketimes = -cumsum(log(rand(1,Nspikes)))/rate;
```

Comments on analyses and use of dummy data

Data are analyzed in order to test whether they support one hypothesis or another. In some cases, the data expected by one (or both) of the hypotheses can be simulated. In such an ideal situation, the data generated for each hypothesis can be made as variable as the real data, by either additional noise or by parameter variation. Results of the analyses of these different samples of simulated data can be directly compared with the analysis of the real

data. In principle, statistically significant results can be confirmed when the real analysis falls within the spread of results generated by the many random instantiations of one hypothesis and not the other.

Even if simulated data are produced in too simplistic a manner to be compared directly to the real data, one can, and should, still analyze simulated data to test the validity of any analysis. For example, if we see a change in the Fano factor in the real data, it is worth simulating many sets of spike trains with similar statistics to the real data, using a single underlying model. We can then measure how much the Fano factor can vary across trials when the underlying process is constant. If the observed variation in the real data falls into the range of simulated variation, we realize that such statistical noise could easily explain our observations and do not need to pursue other explanations. That is, while the change in Fano factor across our data would not rule out other explanations, it would not provide evidence for them.

Finally, even before carrying out an experiment designed to distinguish two hypotheses, it is valuable to simulate data in a simple manner (such as via an inhomogeneous Poisson process) according to each hypothesis. The spike trains produced by the simulated data should be analyzed in the same manner as planned for the real data. Following many such "simulated experiments" it will be clear whether the two hypotheses can be distinguished reliably by the planned methods. If the simulated datasets are not reliably distinguishable, then new analyses, or a different experiment, or more samples are needed.

Tutorial 3.2. Statistical properties of simulated spike trains.

Part A. AELIF neuron with noise.

1) Simulate an AELIF neuron with the following parameters: $E_L = -70\text{mV}$, $V_{th} = -50\text{mV}$, $V_{reset} = -80\text{mV}$, $\Delta_{th} = 2\text{mV}$, $G_L = 10\text{nS}$, $C_m = 100\text{pF}$, $a = 2\text{nS}$, $b = 0\text{nA}$ and $\tau_{SRA} = 150\text{ms}$. Use a time-step, Δt, of 0.01ms and simulate a duration of 100s. Set the input current, I_{app}, to have a mean of zero but with a different value on each time step selected from the normal distribution with a standard deviation of $\sigma/\sqrt{\Delta t}$. (Compare Tutorial 2.1, Q.2).

a) Initially set $\sigma = 50$ pA·s$^{0.5}$ (the exponent of 0.5 in the time units is canceled when divided by $\sqrt{\Delta t}$). Record the spike times and the set of inter-spike intervals (ISIs) by taking the difference between the spike times.

(i) Plot the histogram of ISIs (use "`histogram`" with 25 bins for the ISI).

(ii) Calculate the CV of the ISIs as the standard deviation divided by the mean.

(iii) Calculate the number of spikes in each consecutive 100ms window. Calculate the variance and mean of these numbers (you can use in-built functions) and use these results to calculate the Fano factor.

(iv) Repeat (iii) but using a loop that allows the window-size used to range from 10ms up to 1 second. Plot Fano factor against window-size.

b) Repeat all of the steps of 1a) but with AELIF parameter $b = 1\text{nA}$. Explain any differences in your results from a) and comment on any dependence on the time-window of the Fano

factor in (iv). *Hint: how does the variance compare to the mean of a set of values that are only 1s or 0s?*

c) Repeat steps (i)-(iii) of 1a) with AELIF parameter $b = 0$nA, while reducing the noise to set $\sigma = 20$ pA·s$^{0.5}$. Add to the input current constant terms of 0nA, 0.1nA, and 0.2nA. Comment on how the results change with added input current.

OPTIONAL Part B. Homogeneous and inhomogeneous Poisson process.

Note: You can use the online function "`alt_poissrnd`*", or an in-built function to generate a Poisson process, or write your own function for this part of the tutorial. If you write your own function you can approximate the Poisson process by assuming a single spike occurs in each time-bin with probability of rate multiplied by bin-width and otherwise no spikes occur. Such an approximate Poisson process becomes the Poisson process by definition, when the bin-width approaches zero. However, for finite bin-widths the possibility of more than one spike per time-bin must be included for an exact Poisson process. Since neurons never fire two spikes within 1ms of each other, the approximate Poisson process provides a better approximation to spike trains of real cells than the exact Poisson process, so the approximate process can be used for this tutorial.*

2a) Simulate a 100-second spike train as a homogeneous Poisson process (*i. e.*, one of fixed rate) with a rate of 20Hz, using time-bins of size 0.1ms. Proceed through parts (i)-(iv) of question 1a) to assess the statistics of the Poisson spike train.

b) Simulate a set of 1000 trials, each of 10-seconds duration, of the homogeneous Poisson process. Store the spikes as an array of 1s and 0s, with each row corresponding to a trial.

c) Use the function '`cumsum`' to accumulate across columns (the 2nd dimension of the array) the total spike count as a function of time, $N(t)$, in each trial.

d) Now calculate the variance, Var$[N(t)]$, and mean, $\overline{N(t)}$, across rows (the 1st dimension of the array) of the cumulative spike count. Plot the ratio to see the Fano factor as a function of elapsed time, $F(t)$.

e) In an alternative method, use a fixed size of 200ms to produce a series of successive time-windows. Count the numbers of spikes in each 200ms time-window independently in each trial. Then compare values across trials as in d) to calculate the Fano factor for each 200ms time-bin through the trial.

3) Simulate spike trains for 1000 trials of 10 seconds as an **inhomogeneous** Poisson process (one with a time-varying rate) that follows a firing rate, $r(t) = 25 + 20\sin(2\pi t)$ on each trial. Repeat 2c)-2e) above.

CHALLENGE Part C. Doubly stochastic point processes.

4a) Simulate spike trains for 1000 trials of 10 seconds as an inhomogeneous Poisson process according to a firing rate, $r(t)$, with an initial value of $r(0) = 5$Hz and a final value of $r(10) = 25$Hz. For each trial select a different random time point with uniform likelihood between 2.5s and 7.5s. At that point in the trial change the rate from the initial

rate to the final rate.

b) Plot the mean and the variance of firing rate across trials as a function of time (*i. e.*, the rate in each time-point can now have a different value on different trials—for each time-point calculate the mean and variance of these values).

c) Repeat 2c)-2e) for this doubly stochastic point process (the process is doubly stochastic because underlying rate varies randomly and spikes are produced randomly according to that rate).

5a) Simulate spike trains for 1000 trials of 10 seconds as an inhomogeneous Poisson process according to a firing rate, $r(t)$, with an initial value of $r(0) = 25$Hz that accumulates noise in the firing rate as a random walk according to the equation:

$$r(t + \Delta t) = r(t) + \sigma_r \sqrt{\Delta t} \widetilde{w}_n$$

with $\sigma_r = 5$Hz and where $\widetilde{w}_n$ is a random number selected from a Gaussian (Normal) distribution with zero mean and unit variance (as produced by the '`randn`' function in Matlab: see Chapter 0). Notice that the term added can be first calculated as an array then the function '`cumsum`' can be used to accumulate the changes in firing rate.

Use the '`max`' and '`min`' functions to ensure that no value in the array of firing rates is less than zero or greater than 50Hz.

b) Calculate the across-trial mean and standard deviation of the firing rate as a function of time using the values of $r(t)$ that you have generated.

c) Repeat 2c)-2e) with the new doubly stochastic point process.

The online function "`fano.m`*" can be used to carry out the Fano factor analyses as needed.*

Receiver-operating characteristic (ROC)

Given the variability of neural firing, both in the presence and absence of a stimulus, one might wonder whether a single neuron's response can indicate with any reliability whether a stimulus is present or absent. A means of assessing the reliability of such signal detection—perhaps better termed the discriminability—is through the receiver-operating characteristic (ROC) to be described below. ROC methods are valuable beyond neuroscience—for example in medical diagnostic tests—whenever a single, potentially unreliable measurement is used to determine which one of two conditions exists.

Producing the ROC curve

A receiver-operating characteristic (ROC) is used when assessing how much evidence a given response provides in favor of one alternative hypothesis over another. In signal detection theory, one hypothesis would be the presence of a signal; the alternative would be the absence of that signal. For example, in medical tests the response indicates the presence or absence of a disease with different degrees of likelihood. Tests are rarely 100% accurate, so a threshold is set at a level based on the tradeoff between avoiding false positives (better achieved with a high threshold for detection) versus increasing the power

to capture all true positives correctly (better achieved with a low threshold for detection). The receiver-operating characteristic can be viewed as a curve, by plotting the probability of a true positive result (True +ve, or the hit rate) versus the probability of a false positive (False +ve) result as the threshold for detection is varied (Figure 3.8).

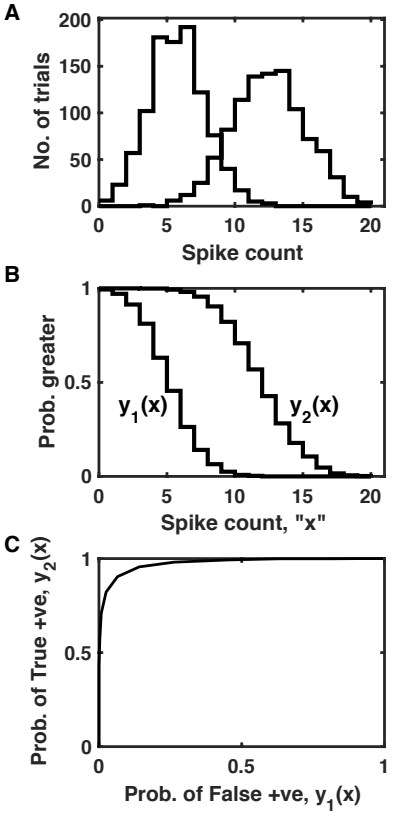


Figure 3.8. The ROC curve. A) Histograms of the number of spikes produced by a noisy AELIF neuron across 1000 trials with a time-window of 0.5 seconds and a stimulus of 0.1nA (right histogram) or no stimulus (left histogram) combined with input noise with $\sigma = 20$ pA.s$^{0.5}$. **B)** The cumulative sum of each histogram is calculated and divided by the total number of trials, then subtracted from 1 to obtain the fraction of trials with more spikes than a given spike count. **C)** The y-values from each of the two curves in B) at the same x-value are plotted against each other to produce the ROC curve as the x-value varies. That is, each point on the ROC curve can be produced by choosing a threshold at a particular spike count (x-value) such that a positive (+ve) test result is produced if the spike count is above that threshold. The fraction of trials with above-threshold spike counts in the absence of stimulus is the false positive value on the x-axis (y_1). The fraction of trials with above-threshold spike counts in the presence of a stimulus is the true positive value (or hit rate) on the y-axis (y_2). Parameters for these data are those given in Tutorial 3.3, Q1.

ROC analysis can also be used to assess how much the spike train of a single neuron (or the aggregate of a group of neurons) can contribute to an animal's ability to distinguish between two stimuli, or distinguish the presence from the absence of a stimulus. Even if the firing rates are very different across the two conditions, the duration of the stimulus or the time across which any circuit in the animal's brain can accumulate spikes may be limited. Therefore, if spikes are emitted by the neuron as a noisy process, it is possible that fewer spikes are counted on some trials with a higher expected firing rate than are counted on some trials with a lower expected firing rate. That is, if the numbers of trials with different spike counts are accumulated and plotted as a histogram (Figure 3.8A) then it is possible that the two distributions overlap. Such overlap makes it impossible to produce a threshold such that all trials of one condition produce a spike-count that is higher than the threshold, while all trials of the other condition produce a spike-count that is lower than the threshold.

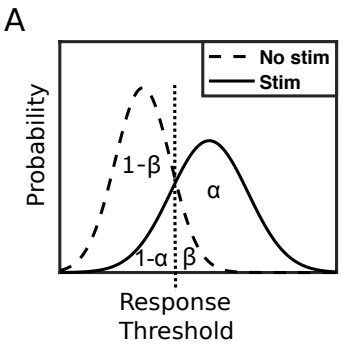
The overlap of the two distributions is reduced—and thus an observer's ability to discriminate the two conditions increased—if either the separation of the means of the

distributions increases, or if the standard deviations of the distributions decrease. A measure of this overlap is the discriminability index, d', which is defined by $d' = \frac{\Delta(mean)}{\sigma}$ where $\Delta(mean)$ is the difference between the means of the two distributions and where $\sigma = \sqrt{\frac{1}{2}(\sigma_A^2 + \sigma_B^2)}$ with σ_A and σ_B being the standard deviations of each individual distribution. If the noise does not vary between the two distributions, such that their variances are equal, then $\sigma = \sigma_A = \sigma_B$.

A little calculus (see Appendix B) reveals that if the two distributions are Gaussian with equal variances, then the probability of an error, $P(\text{Error})$, (where an error is either a "miss" or a false positive) given an optimal positioning of the threshold (optimized to minimize probability of an error) is equal to

$$P(\text{Error}) = \frac{1}{2}P(\text{Miss}) + \frac{1}{2}P(\text{False +ve}) = \frac{1}{2}\text{erfc}\left[\frac{\Delta(mean)}{\sqrt{2}\sigma}\right] = \frac{1}{2}\text{erfc}\left(\frac{d'}{2\sqrt{2}}\right). \quad \text{Eq. 3.4}$$

Optimal position of the threshold



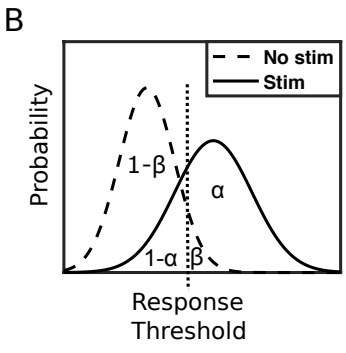


Figure 3.9. Optimal position of the threshold for stimulus detection. A. If both possibilities are equally likely *a priori* then threshold should be at the intersection of the two probability distributions. **B.** Raising the threshold above the optimum reduces false positives (β) but reduces true positives (α)

by a greater amount.

If our goal is to minimize the probability of an error when *a priori* (before measurement) a stimulus is as equally likely to be present as absent, the position of the threshold for stimulus detection should be at the intersection of the two probability distributions (Fig. 3.9A).

To see this, consider moving the threshold from the intersection to a position slightly to the right (Fig. 3.9B). In this case, the higher threshold would reduce the number of false positives (fraction of stimulus-absent trials to the right of the threshold) while increasing the number of misses (fraction of stimulus-present trials to the left of the threshold). However, the reduction in false positives is less than the increase in misses (the stimulus-present curve is above the stimulus-absent curve to the right of the intersection) so the net effect is to increase the fraction of errors.

Similarly, if the threshold were decreased from the intersection then the number of misses would decrease, but the number of false positives would increase by more, again generating more errors. So, in this case the minimum number of errors is at the intersection.

It is useful to notice that when the threshold is at the intersection, the rate of change of false positives with a change in threshold exactly matches the rate of change of true positives (hits) with a change in threshold. This translates to the ROC curve having a gradient of one, as the rate of change of y_1 equals the rate of change of y_2.

The argument can be generalized to cases where the known prior likelihood of a stimulus differs from ½. For example, if the fraction of trials with a stimulus were ¼ and we now want to choose an optimal threshold, we will be minimizing $P(\text{Error}) = \frac{1}{4}P(\text{Miss}) + \frac{3}{4}P(\text{False +ve}) = \frac{1}{4}(1-\alpha) + \frac{3}{4}\beta$. We can use the same argument as above if we plot the stimulus-absent probability distribution at three times the size of the stimulus-present probability distribution (since the number of stimulus-absent trials is three times the number of stimulus-present trials) and evaluate the new intersection. Given this scaling the new intersection occurs at a threshold where the rate of change of true positives is three times the rate of change of false positives. On the ROC curve this corresponds to a point where the gradient is three.

In general, *a priori* the probability of a stimulus being absent can be some number, which we will call k, times more likely than the probability of a stimulus being present. We then scale the stimulus-absent probability distribution up by k-fold and look at its intersection with the stimulus-present probability distribution. The threshold for stimulus detection is set at the intersection, which is where the stimulus-present probability is k times the un-scaled stimulus-absent probability (for a proof see Appendix B). This means the rate of change of true positives is k times the rate of change of false positives, so the optimal threshold appears on the ROC curve where the gradient is k.

Finally, in real-life situations the costs of a false positive and a miss are rarely equal. For a house-fly, the cost of a false positive when responding to the possibility of a looming object such as a hand descending upon it, is the waste of energy entailed in flying away. However, the cost of a false negative, or a miss, is likely death as the hand squashes it. In

experimental trials the relative costs can be controlled by adjusting the amount of reward (*e. g.*, the number of drops of fruit juice) an animal receives for the various outcomes of the trial. In such cases, one wants to weight the probability of each type of error by the cost, or the loss, associated with that error.

The loss associated with an error is the difference between the value to the animal of a correct response and the value to the animal of an incorrect response for that stimulus. To select an action optimally, a common unit, such as value, is needed to compare and weigh the potential outcomes against each other. Given such rational accounting, a missed reward counts as a loss just as does a punishment that could have been avoided. For example, in the unfortunate case of a harmful stimulus that cannot be avoided even if correctly identified, then there is in fact no loss associated with a failure to respond to it—the outcome is equally detrimental whether the test for the stimulus is a hit or a miss.

If the loss functions associated with the different stimulus-response possibilities are known, a rational goal for any animal is to minimize its overall loss. In the case of stimulus detection, we minimize expected loss, $\langle L \rangle$, given by

$$\langle L \rangle = L(\text{Miss})P(\text{Stim})P(\text{Miss}) + L(\text{False +ve})[1 - P(\text{Stim})]P(\text{False +ve}) \qquad \text{Eq. 3.5}$$

where $L(\text{Miss})$ and $L(\text{False +ve})$ are the losses associated with a miss and a false positive respectively.

We see that minimization of the loss requires almost the same calculation as minimization of the probability of error, except with the prior probability of a stimulus, $P(\text{Stim})$, replaced by the product of that probability with its associated loss. Similarly, the prior probability of the absence of a stimulus, $[1 - P(\text{Stim})]$ should be weighted by its associated loss. Therefore, following the previous arguments for optimal positioning of the threshold (and the associated calculations of Appendix B) the threshold for stimulus detection should be set so that the ratio of the probabilities for response with stimulus-present to response with stimulus-absent—*i. e.*, the gradient of the ROC curve (Fig. 3.8C)—is given by multiplying the loss ratio by the ratio of prior stimulus probability

$$\frac{dy_2}{dy_1} = \frac{L(\text{False +ve})[1 - P(\text{Stim})]}{L(\text{Miss})P(\text{Stim})}. \qquad \text{Eq. 3.6}$$

In summary, a change in the loss ratio should have the same impact on behavior as a change in the prior stimulus probability. For example, making the reward for a true positive three times as great as the reward for a correct rejection (true negative) should have an equivalent effect to presenting the stimulus on three times the number of trials as omitting it. In experiments either technique can be used to alter the threshold used in binary response tasks.

Uncovering the underlying distributions from binary responses: recollection versus familiarity

Figure 3.10 indicates how ROC analysis can be used to gain information about the underlying probability distributions used for making different responses. It can be particularly useful to know if behavior arises from a bimodal probability distribution, because bimodality is an indication of two distinct underlying processes. For example, the

underlying pitch of adult speakers has a bimodal distribution arising from differences between male and female speakers—and that the pitch is bimodal allows us to quite reliably tell the gender of someone speaking.

Here we will consider the usefulness of ROC analysis for revealing whether a particular cognitive function is based on one or two distinct underlying processes. If some trials initiate one process while other trials initiate a different process, then the overall probability distribution of responses would be a combination of the two distinct underlying probability distributions. Such a combined distribution may be bimodal (though is not necessarily so).

An ongoing debate in psychology, with considerable evidence for two distinct processes based on ROC analysis, is whether recollection involves a process that is distinct from recognition memory, or familiarity. Recollection occurs when we recall some of the details of an event, so that we can be quite certain that the stimulus was observed (although false memories are common). Recollection is suggestive of a binary, all-or-none, process—we recall doing something or we do not. The associated neural activity is likely to be bimodal, with a lack of recollection—meaning no memory of an event—being equivalent to the absence of the event we are asked to recollect (Fig. 3.10A1).

Recognition occurs when we think we have seen the stimulus before but do not recall the event of seeing the stimulus—we are not certain the way we are when we manage to recall an event. A recognized event would suggest neural activity that is somewhat distinct but not completely distinct from that which would arise in the absence of the event.

If the processes of recollection and recognition are the same, one might expect a broad distribution of responses when trying to remember a real event (as in Figure 3.10A3), in which case the demarcation between recognition and recollection just appears when activity is sufficiently distinct from the activity in the absence of the event. However, if recollection relies on a different process from mere recognition, one might expect to see a bimodal distribution (one with two peaks as in Figure 3.10A2) in some of the neural responses.

Given the difficulty of acquiring the underlying neural responses in humans, attempts are made to infer the underlying distributions from aspects of the behavior, such as response speed. Alternatively, an experimenter can use confidence in the reliability of a memory as a readout of neural response, and use confidence as a substitute for response threshold to produce an ROC curve (all choices with a given confidence or greater are considered "above threshold"). The confidence in a memory can be requested, or attempts to measure the confidence can be made by altering the relative losses associated with false-positives or misses.

Different distributions of confidence produce different ROC curves. If a subject were only either 100% certain (and then correct) or simply guessing, the resulting ROC curve would be linear with a positive y-intercept (Figure 3.10B1). In this case, the intercept would indicate the fraction of trials on which the subject is certain. If a subject's confidence were based on two distinct process that produced either recall or recognition, then the corresponding neural response distributions would be distinct from each other and distinct from the stimulus-absent response distribution (Figure 3.10A2). The resulting ROC curve

would be non-linear (Figure 3.10B2). However, such a curve can appear very similar to the curve produced when the stimulus produces just a single, broad response distribution arising from a single memory process (Figure 3.10A3).

The probabilities of a false positive and a true positive can be converted to z-scores, where the z-score is the number of standard deviations away from the mean assuming a Gaussian distribution. Thus, a cumulative probability of one half corresponds to a z-score of zero since half of the area lies above the mean of a Gaussian distribution. The details of such a conversion are provided in Appendix B with the results reproduced in Figure 3.10C. In this case, the significant non-linearity produced by bimodal probability distributions (Figures 2.10C1 and 2.10C2) disappears when the stimulus distribution is a broad Gaussian (Figure 3.10C3), allowing for a distinction between a single process (one broad peak) or multiple underlying processes (producing two peaks).

In practice, great care is needed in producing such curves, as the significant deviations from linearity in Figure 3.10C2 lie below z-scores of -2 for the false positive responses. Only approximately 2.5% of responses lie in the range of 2 or more standard deviations above the mean, so a lot of data must be collected before one can be confident in the position of the curve in this region.

More generally, whenever such curves are produced following multi-step data analysis, it is recommended that results are compared with those arising from identical analysis of dummy data. A scientist can sample from the inferred response distributions to produce a dummy data set with the same number of data points as the original data. The process can be repeated with a large number of dummy data sets, so a range of results can be obtained and significance of the original data can be assessed accordingly.

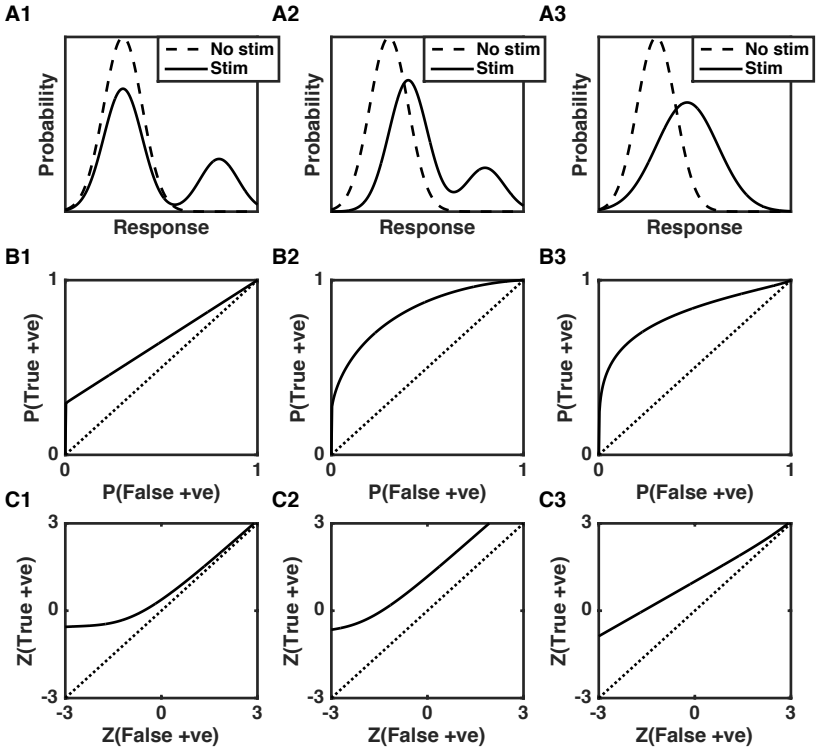


Figure 3.10. ROC analysis to reveal the stimulus-response distribution. A) Three alternative stimulus-response distributions are depicted. **A1)** The response to the stimulus is bimodal, with a strong response on a fraction of trials (recall), but otherwise no distinction from the absence of a stimulus (guessing). **A2)** The response is also bimodal, but the weak response (recognition) differs from the no-stimulus condition. **A3)** The stimulus-response arises from a single Gaussian distribution with increased mean and variance compared to the no-stimulus condition. **B)** ROC curves produced from the distributions in A) (solid) with the chance diagonal line (dotted). **B1)** The fraction of trials with high response produces a non-zero y-intercept. Other trials arise from the same distribution as the no-stimulus condition, so the curve is linear with gradient equal to the ratio of the sizes of the lower distributions. **B2)** The fraction of high-response trials still produces a positive y-intercept, but now the low-response trials are sampled from a distribution with higher mean than the no-stimulus condition, so the curve is concave. **B3)** A single broad stimulus-distribution produces a similarly shaped ROC curve to B2) making it non-trivial to distinguish distributions A2 and A3 from ROC curves alone. **C)** ROC curves plotted on z-score coordinates, where z-score is number of standard deviations above or below the mean assuming a single Gaussian distribution. **C1, C2)** Sampling from two Gaussians for response to the stimulus leads to a curved plot. **C3)** When both stimulus-present and stimulus-absent conditions each produce a single Gaussian distribution in A) then a plot of the z-scores against each other is linear. The gradient is the ratio of the standard deviations. Code to produce these figures is available online as `ROC_distributions.m`.

Tutorial 3.3. Receiver-operating characteristic of a noisy neuron.

In this tutorial, we assume a neuron receives one type of input current when a stimulus is present and a second type of input current in the absence of a stimulus. The two types of input current can differ in their mean value and/or their standard deviation. We will assume the number of spikes over a certain time interval can be detected and used to indicate the presence or absence of the stimulus. We will assess how well these two cases can be distinguished by setting different values of the threshold for stimulus-detection.

For all questions, you will simulate an AELIF neuron, following the dynamics described in Tutorial 2.3 Q.2, using the parameters: $E_L = -70\text{mV}$, $V_{th} = -50\text{mV}$, $V_{reset} = -80\text{mV}$, $\Delta_{th} = 2\text{mV}$, $G_L = 10\text{nS}$, $C_m = 100\text{pF}$, $a = 2\text{nS}$, $b = 0\text{nA}$ and $\tau_{SRA} = 150\text{ms}$.

1) Set the mean input current in trials with no stimulus to be 0 and in trials with a stimulus present to be 0.1nA. For each time step, add a random number taken from the normal distribution of zero mean and standard deviation of $\sigma/\sqrt{\Delta t}$ (see Chapter 0) where Δt is the time-step of 0.01ms. Set $\sigma = 20\text{pA.s}^{0.5}$ in both conditions (stimulus-absent and stimulus-present).

a) Simulate each condition for 1000 trials of duration 0.5 sec.

b) Count and store the number of spikes in each trial of each condition.

c) Calculate the mean firing rate of each condition.

d) Use the "`histcounts`" function (or your own function) to count the number of trials with a given number of spikes in each condition separately. To produce identically sized vectors for the two conditions, use a range from 0 to the maximum number of spikes found in a trial in either condition for the possible values of number of spikes per trial.

e) Plot the histogram of the number of spikes per trial for each condition (use a staircase plot with the function "`stairs`") (as in Figure 3.8A).

f) Calculate the fraction of trials with greater than a given spike count by using the cumulative sum ("`cumsum`") of the histograms, normalizing by (*i. e.*, dividing by) the total number of trials and subtracting from one—or instead of subtracting from one you can use "`cumsum(histogram_output,'reverse')`". Plot this monotonically decreasing fraction for each condition (as in Figure 3.8B).

g) For each value of spike count as a threshold, plot the fraction of trials with greater than that spike count in the stimulus-on condition as "True positives", α, on the y-axis against the fraction of trials with greater than that spike count in the stimulus-off condition as "False positives", β, on the x-axis (as in Figure 3.8C). This is the receiver operating characteristic curve.

2) Repeat Q1, but with a stimulus duration of 0.2 sec instead of 0.5 sec. For each of the three figures explain any differences you observe from the results of Q1.

3) Repeat Q1 with a stimulus duration of 0.5 sec, but with the following alterations:

Mean applied current is 0.5nA in the presence of a stimulus (and still 0nA in the absence of a stimulus). The noise is reduced in the presence of a stimulus ($\sigma = 5\text{pA.s}^{0.5}$) and increased in the absence of a stimulus ($\sigma = 50\text{pA.s}^{0.5}$). Comment on and explain any difference in the results due to these altered stimulus and noise properties. Just by counting spikes in a trial,

could you ever be nearly certain that a stimulus is present? Could you ever be nearly certain that a stimulus is absent?

Appendix A: The Poisson process

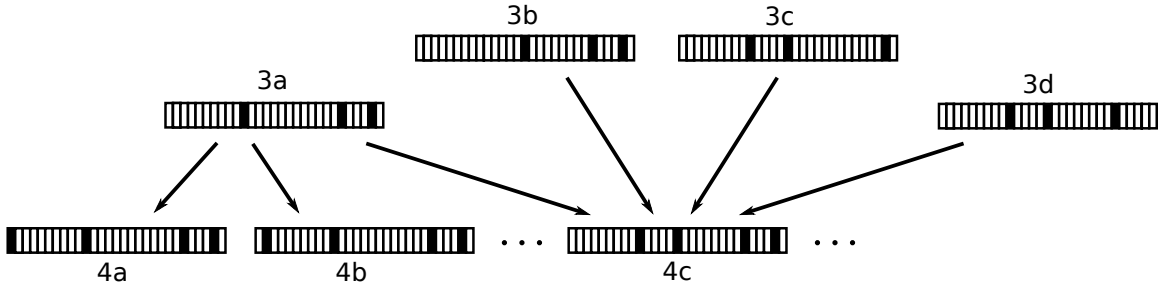


Figure 3.11. Relating $P(N)$ to $P(N-1)$. Any combination of events that contributes to the probability of 3 events in an interval—such as the combination labeled 3a above—is one event away from each of N_T combinations of events that contribute to the probability of 4 events in an interval, like those labeled 4a, 4b, and 4c above. However, each of those 4-event combinations can be arrived at 4 different ways—for example, combination labeled 4c above is reached by adding one event to any of the combinations labeled 3a, 3b, 3c, or 3d. Therefore, the actual number of 4-event combinations is $N_T/4$ times the number of 3-event combinations. Since addition of an extra event occurs with probability $r\delta t/(1-r\delta t)$, we find—by multiplying the relative probabilities of a single combination by the relative numbers of combinations—that the probability of 4 events, $P(4)$, is related to the probability of 3 events, $P(3)$, by the formula $P(4) = P(3) \cdot r\delta t N_T/[4(1-r\delta t)]$.

When considering a general whole number of events, N, this formula becomes $P(N) = P(N-1) \cdot r\delta t N_T/[N(1-r\delta t)]$ via similar logic.

The homogeneous Poisson process is a point process that describes a series of events, which occur at discrete instants of time, with the probability of an event at any one point in time being identical to the probability at any other point in time. The process is a Markov process, which means that the probability is not dependent on history—the series of outcomes of fair coin flips is a Markov process, because whether we have just flipped a series of heads has no impact on whether the next flip is a head or a tail.

The Poisson distribution

In this section, we will derive the Poisson distribution (Eq. 3.2), which is the probability of a given number of events in a certain time interval produced by a Poisson process. We have already seen that the formula for the Poisson distribution can be used to show that the distribution of times between events (the inter-event interval distribution, or for neurons the inter-spike interval, ISI, distribution) is an exponential decay.

The definition of a Poisson process is one in which the probability of an event in a very small interval, δt, is a constant, $r\delta t$. We consider a much longer interval, T, comprised of N_T small intervals ($T = N_T\ \delta t$) and ask the probability of a given number, N, of events. We assume that the small interval, δt, is small enough that $r\delta t \ll 1$, and therefore—since most time-bins are empty—that $N \ll N_T$.

We will calculate the probability of N events in an interval based on the probability of $N-1$ events in the interval then use the fact that the probabilities must sum to one when all possible values of N are combined. For example, we define the probability of no events in the long interval, T, as $P(0)$ (which we need to calculate). The probability of a single event in a particular interval and no others is related to $P(0)$, but has an additional factor, because one time bin that had no spike now contains a spike—a factor of $(1 - r \cdot \delta t)$ that contributed to $P(0)$ is replaced by $r \cdot \delta t$ when contributing to $P(1)$.

Therefore, the probability of an event in one particular small time bin and no other is $P(0) \cdot r\delta t/(1-r\delta t)$.

There are N_T such small intervals and an event in any one of them could give rise to a single spike in the total interval. Summing these N_T different ways of getting one spike yields $P(1) = N_T r\delta t P(0) = rTP(0)/(1-r\delta t)$.

We can assume the factor $(1-r\delta t) \approx 1$ (because $r\delta t \ll 1$) and see that $P(1) = rTP(0)$.

We can make a similar argument in general for $P(N)$ as any arrangement of N spikes in an interval is given by an arrangement of $N-1$ spikes that contributes to $P(N-1)$ but with one time bin that did not contain a spike now containing a spike. As with comparing $P(1)$ with $P(0)$, this difference, when accounting for any time-bin the extra spike can be in, leads to a factor of $(N_T - N)r\delta t = rT$ (where we have used $N \ll N_T$, so the result is true as $\delta t \mapsto 0$). Notice that with point processes, each event has a duration of 0, so however many events are in a time interval, the total time in between events that contains no events is equal to the time interval itself.

The story does not end there, as a little care is needed (see Figure 3.9). Any one of the N spikes in an arrangement that contributes to $P(N)$ could be omitted to produce an arrangement that contributes to $P(N-1)$. That is, if we took all arrangements of spikes that contribute to $P(N-1)$ with a spike added to any time-bin, we would find each arrangement that contributes to $P(N)$ appearing N times. To combat such over-counting, we must divide by N when evaluating $P(N)$ from $P(N-1)$.

Therefore, the general formula is

$$P(N) = P(N-1)\frac{rT}{N} \quad (\text{for } N > 0). \qquad \text{Eq. 3.7}$$

Using Eq. 3.7 to iterate up from $P(0)$, we see that

$$P(1) = P(0)rT$$

$$P(2) = P(1)\frac{rT}{2} = P(0)\frac{(rT)^2}{2!}$$

$$P(3) = P(2)\frac{rT}{3} = P(0)\frac{(rT)^2}{3!}$$

and so on. This can be written more generally as

$$P(N) = P(0)\frac{(rT)^N}{N!}. \qquad \text{Eq. 3.8}$$

All that remains is to find $P(0)$ from the normalization

$$\sum_{N=0}^{\infty} P(N) = 1, \qquad \text{Eq. 3.9}$$

which formalizes the certainty that in any interval there will be a particular number of events that is a non-negative integer.

Finally, recognizing that the exponential function is defined by

$$e^x = \sum_{N=0}^{\infty} \frac{x^N}{N!}$$

we can evaluate $P(0)$ by combining Eqs. 3.8 and 2.9 to yield

$$\sum_{N=0}^{\infty} P(N) = P(0)\sum_{N=0}^{\infty} \frac{(rT)^N}{N!} = P(0)e^{rT} = 1,$$

hence $P(0) = e^{-rT}$ and in general the Poisson distribution is given by

$$P(N) = e^{-rT}\frac{(rT)^N}{N!}. \qquad \text{Eq. 3.10}$$

Expected value of the mean of a Poisson process

The expected value of any distribution is given by multiplying the value of each possible outcome by the probability of that outcome then summing these values together. This leads to the expected value, $\langle N \rangle$, for the number of events in an interval T of a Poisson process with rate parameter r as:

$$\begin{aligned}
\langle N \rangle &= \sum_{N=0}^{\infty} N \cdot P(N) \\
&= e^{-rT} \sum_{N=0}^{\infty} N \cdot \frac{(rT)^N}{N!} \\
&= rTe^{-rT} \sum_{N=1}^{\infty} \frac{(rT)^{N-1}}{(N-1)!} \\
&= rTe^{-rT} \sum_{N'=0}^{\infty} \frac{(rT)^{N'}}{(N')!} \qquad \text{(now use the formula } \sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x\text{)} \\
&= rTe^{-rT} e^{rT} \\
&= rT.
\end{aligned}$$

Eq. 3.11

This result confirms that r is the mean rate of events for a Poisson process.

Fano factor of the Poisson process

The Fano factor of a process is the variance divided by the mean of the number events. To evaluate the variance, we need to calculate the expected value of the square of the number of events, $\langle N^2 \rangle$. Using a similar procedure to the one followed when calculating the mean, and rewriting N^2 as $N(N-1) + N$ to simplify the calculation, we find:

Eq. 3.12

$$= e^{-rT} \sum_{N=2}^{\infty} \frac{(rT)^N}{(N-2)!} + \langle N \rangle$$

$$= (rT)^2 e^{-rT} \sum_{N'=0}^{\infty} \frac{(rT)^{N'}}{N'!} + \langle N \rangle$$

$$\langle N^2 \rangle = (rT)^2 + rT.$$

Therefore the variance in N, obtained by combining Eqs. 3.9 and 2.10, is:

$$\begin{aligned} \mathrm{Var}(N) &= \langle N^2 \rangle - \langle N \rangle^2 \\ &= (rT)^2 + rT - (rT)^2 \\ &= rT \end{aligned} \qquad \text{Eq. 3.13}$$

and the Fano factor, $F(N)$ is given as:

$$F(N) = \frac{\mathrm{Var}(N)}{\langle N \rangle} = 1. \qquad \text{Eq. 3.14}$$

This proves the result that a Poisson process has a Fano factor of 1.

The coefficient of variation (CV) of the ISI distribution of a Poisson process.

The CV of a distribution is its standard deviation divided by its mean. The CV of an exponential distribution—such as the distribution of inter-spike intervals (ISIs) when spike emission follows a Poisson process—can be shown to be 1.

For such a Poisson process, Eq. 3.3 shows that the probability of an ISI being between T and $T + \delta T$ is given by

$$P(T)\delta T = r\delta T e^{-rT} \qquad \text{Eq. 3.15}$$

The mean ISI is calculated using integration by parts:

$$\begin{aligned} \langle T \rangle &= \int_0^{\infty} TP(T)\, dT \\ &= \int_0^{\infty} Tre^{-rT}\, dT \\ &= [-Te^{-rT}]_0^{\infty} - \int_0^{\infty} -e^{-rT}\, dT \\ &= 0 + \left[-\frac{e^{-rT}}{r}\right]_0^{\infty} \\ &= \frac{1}{r}. \end{aligned} \qquad \text{Eq. 3.16}$$

Similarly, the mean of the ISI-squared is given by:

$$\begin{aligned}\langle T^2 \rangle &= \int_0^\infty T^2 P(T)\, dT \\ &= \int_0^\infty T^2 r e^{-rT}\, dT \\ &= [-T^2 e^{-rT}]_0^\infty - \int_0^\infty -2Te^{-rT}\, dT \\ &= 0 + \frac{2}{r}\langle T \rangle \\ &= \frac{2}{r^2}.\end{aligned} \qquad \text{Eq. 3.17}$$

Hence, combining Eq. 3.17 and 2.16, the standard deviation of the ISI distribution is:

$$\begin{aligned}\sigma_T &= \sqrt{\langle T^2 \rangle - \langle T \rangle^2} \\ &= \sqrt{\frac{2}{r^2} - \left(\frac{1}{r}\right)^2} \\ &= \frac{1}{r}.\end{aligned} \qquad \text{Eq. 3.18}$$

Therefore, the standard deviation is equal to the mean of an exponential probability distribution (from Eq. 3.16 and Eq. 3.18) and its coefficient of variation, the ratio of standard deviation to the mean, is equal to one.

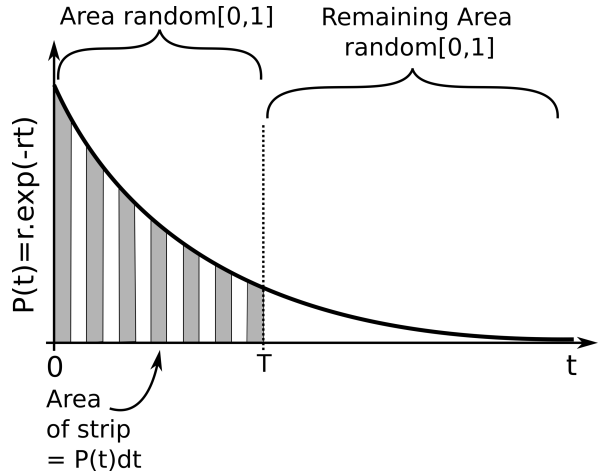


Figure 3.12. Random selection from a known distribution. The area to the left of a randomly chosen interval, T, is the sum of the areas of strips of height $P(t)$ and width dt. If a point is chosen at random with equal probability from within the total area of the distribution, its likelihood to fall on a given strip is equal to the area of that strip, which is $P(t)dt$. Such selection can be made by choosing a random number from a uniform distribution between zero and 1 and requiring either the area to the left, or the area to the right of the chosen interval, T, is equal to that random number.

Selecting from a probability distribution: generating ISIs for the Poisson process

If we know the probability distribution of ISIs, we can generate an artificial spike train with that distribution. If one ISI has no impact on successive ISIs then the sequence of ISIs is a

Markov process, meaning it is independent of history. In such cases the artificial spike train can be statistically equivalent to the original spike train, and so be useful as a control in any statistical analyses.

The procedure for selecting numbers according to a given distribution is sketched in Figure 3.12. The goal is to select a number with probability proportional to the height of the ISI distribution. If we correctly normalize the ISI distribution to be a probability distribution with area 1 (that is, the integral under the curve is 1) then selecting an ISI is equivalent to randomly selecting a point uniformly within this area and reading off the x-coordinate. For very complicated functions, this can be achieved by selecting random pairs of (x,y) coordinates. Testing whether the pair falls under the curve. Recording the first value that is under the curve and therefore within the distribution. Clearly any x-coordinate at which the probability distribution is greater will have more likelihood of being chosen, as a randomly chosen y-coordinate is more likely to fall under the curve.

For distributions that can be integrated, the process can be made computationally more efficient. For example, when the ISI distribution is exponential, as is the case for the Poisson process, then we can use the result that the integral of an exponential is also an exponential, $\int_0^T re^{-rt}dt = 1 - e^{-rT}$ and $\int_T^\infty re^{-rt}dt = e^{-rT}$. Selecting a point at random within the area of the distribution can then be achieved by randomly choosing the area that it to the left of that point or to the right of that point. For the exponential distribution of ISIs that allows us to choose an ISI of length, T, by requiring that e^{-rT} is a uniformly distributed random number between 0 and 1. A little rearrangement leads to:

$$T = -\frac{1}{r}\ln[random(0,1)] \qquad \text{Eq. 3.19}$$

where $random(0,1)$ means select a random number from a uniform distribution between 0 and 1, *i. e.*, use `rand()` in Matlab.

Appendix B: Stimulus discriminability

We consider a test for the presence of a stimulus based on whether a neuron's response is above a given threshold. The goal is to set the level of the threshold so as to minimize the probability of an error. An error can be a miss (a false negative) meaning the stimulus was present but went undetected ($1 - \alpha$ in Fig. 3.7), or can be a false positive meaning the response was above threshold, indicating the presence of a stimulus when in fact the stimulus was absent (β in Fig. 3.7).

We assume that the test produces a mean response of μ_A when the stimulus is absent and a mean response of μ_B (where $\mu_B > \mu_A$) when the stimulus is present. In each case responses are distributed about those means as Gaussians with standard deviations σ_A and σ_B respectively. If we define the threshold for stimulus detection as θ then the probability of a false positive, which is the probability the response is above-threshold when the stimulus is absent, becomes:

$$P(r > \theta|A) = \frac{1}{\sqrt{2\pi{\sigma_A}^2}}\int_\theta^\infty e^{-\frac{(x-\mu_A)^2}{2{\sigma_A}^2}}dx. \qquad \text{Eq. 3.20}$$

Similarly, the probability of a false negative, *i. e.* a miss, which is the probability the response is below-threshold when the stimulus is present, becomes:

$$P(r < \theta|B) = \frac{1}{\sqrt{2\pi{\sigma_B}^2}} \int_{-\infty}^{\theta} e^{-\frac{(x-\mu_B)^2}{2{\sigma_B}^2}} dx. \qquad \text{Eq. 3.21}$$

Optimal value of threshold

The overall probability of an error is then:

$$P(\text{Error}) = P(r > \theta|A)P(A) + P(r < \theta|B)P(B), \qquad \text{Eq. 3.22}$$

where $P(A)$ and $P(B)$ are respectively the *a priori* probabilities of the stimulus being absent or present (*a priori* meaning the probability prior to any information regarding the neuron's response).

The first term of Eq. 3.20 decreases as we raise the threshold while the second term increases when we raise the threshold. As discussed in Chapter 2, the optimal threshold is where the rate of change from the two terms cancel out. In calculus, this arises because the minimum of any function (such as $P(\text{Error})$ as a function of a variable threshold) corresponds to a point where its gradient is zero. That is, if look at the derivative, $\frac{dP(\text{Error})}{d\theta}$, and find a value of θ where the derivative is zero it may be the optimal value. (It could also be the value with highest error, so one should compare the error at each value of zero derivative and select the highest so long as the error is below ½, which is the value obtained in the limit of very high or very low threshold.)

If we evaluate the derivative of Eq. 3.20, we obtain

$$\begin{aligned} \frac{dP(\text{Error})}{d\theta} = {} & P(A)\frac{1}{\sqrt{2\pi{\sigma_A}^2}}\frac{d}{d\theta}\int_{\theta}^{\infty} e^{-\frac{(x-\mu_A)^2}{2{\sigma_A}^2}} dx \\ & + P(B)\frac{1}{\sqrt{2\pi{\sigma_B}^2}}\frac{d}{d\theta}\int_{-\infty}^{\theta} e^{-\frac{(x-\mu_B)^2}{2{\sigma_B}^2}} dx \\ = {} & P(A)\frac{-1}{\sqrt{2\pi{\sigma_A}^2}} e^{-\frac{(\theta-\mu_A)^2}{2{\sigma_A}^2}} + P(B)\frac{1}{\sqrt{2\pi{\sigma_B}^2}} e^{-\frac{(\theta-\mu_B)^2}{2{\sigma_B}^2}} \end{aligned} \qquad \text{Eq. 3.23}$$

so $\frac{dP(\text{Error})}{d\theta} = 0$ occurs at a value of θ defined by:

$$P(A)\frac{1}{\sqrt{2\pi{\sigma_A}^2}}e^{-\frac{(\theta-\mu_A)^2}{2{\sigma_A}^2}} = P(B)\frac{1}{\sqrt{2\pi{\sigma_B}^2}}e^{-\frac{(\theta-\mu_B)^2}{2{\sigma_B}^2}}, \qquad \text{Eq. 3.24}$$

which can be rewritten as

$$\frac{\frac{1}{\sqrt{2\pi{\sigma_B}^2}}e^{-\frac{(\theta-\mu_B)^2}{2{\sigma_B}^2}}}{\frac{1}{\sqrt{2\pi{\sigma_A}^2}}e^{-\frac{(\theta-\mu_A)^2}{2{\sigma_A}^2}}} = \frac{P(A)}{P(B)}. \qquad \text{Eq. 3.25}$$

In words this means that the ratio of the values of the probability distributions (left hand side) match the inverse ratio of their prior probabilities (right hand side).

Calculating the probability of an error

The position of the optimal threshold can be written more simply in the special case where $P(A)\sigma_A = P(B)\sigma_B$, as for example would happen if the two distributions have equal variance and a stimulus is present on half of the trials.

In this case Eq. 3.23 shows that the optimal threshold is defined by

$$e^{-\frac{(\theta-\mu_B)^2}{2{\sigma_B}^2}} = e^{-\frac{(\theta-\mu_A)^2}{2{\sigma_A}^2}} \qquad \text{Eq. 3.26}$$

so

$$\frac{(\theta-\mu_B)^2}{2{\sigma_B}^2} = \frac{(\theta-\mu_A)^2}{2{\sigma_A}^2} \qquad \text{Eq. 3.27}$$

and because the threshold lies between the two means, $\mu_A < \theta < \mu_B$ this rearranges to give $\sigma_A(\theta-\mu_B) = \sigma_B(\theta-\mu_A)$. Writing the difference between the means as $\Delta = \mu_B - \mu_A$ and rearrangement of Eq. 3.25 allows us to write the threshold as either $\theta = \mu_A + \frac{\sigma_A\Delta}{\sigma_A+\sigma_B}$ or $\theta = \mu_B - \frac{\sigma_B\Delta}{\sigma_A+\sigma_B}$.

Given this value for the threshold, the probability of error becomes, by substituting $x' = x - \mu_A$ in the first term and $x' = \mu_B - x$ in the second term:

$$P(\text{Error}) = P(A)\frac{1}{\sqrt{2\pi{\sigma_A}^2}}\int_{\frac{\sigma_A\Delta}{\sigma_A+\sigma_B}}^{\infty} e^{-\frac{x'^2}{2{\sigma_A}^2}}\,dx'$$

$$+ P(B)\frac{1}{\sqrt{2\pi{\sigma_B}^2}}\int_{\frac{\sigma_B\Delta}{\sigma_A+\sigma_B}}^{\infty} e^{-\frac{x'^2}{2{\sigma_B}^2}}\,dx'$$

$$= P(A)\frac{1}{\sqrt{\pi}}\int_{\frac{\Delta}{\sqrt{2}(\sigma_A+\sigma_B)}}^{\infty} e^{-x^2}\,dx + P(B)\frac{1}{\sqrt{\pi}}\int_{\frac{\Delta}{\sqrt{2}(\sigma_A+\sigma_B)}}^{\infty} e^{-x^2}\,dx$$

$$= \frac{1}{2}\text{erfc}\left[\frac{\Delta}{\sqrt{2}(\sigma_A+\sigma_B)}\right] \qquad \text{Eq. 3.28}$$

where we have used $P(A) + P(B) = 1$ in the final line and used the complementary error function, $\text{erfc}(z)$, which is defined in terms of the Gaussian integral as $\text{erfc}(z) = \frac{2}{\sqrt{\pi}}\int_z^{\infty} e^{-x^2}\,dx$.

In the special case where $P(A) = P(B)$ and $\sigma_A = \sigma_B$ then $P(\text{Error}) = \frac{1}{2}\text{erfc}\left[\frac{\Delta}{2\sqrt{2}\sigma_A}\right] = \frac{1}{2}\text{erfc}\left[\frac{d'}{2}\right]$ as used in Chapter 2.

Generating a z-score from a probability

ROC curves are usually plotted as the probability of a true positive (a hit) against the probability of a false positive. They can also be plotted as the z-score of true positives against the z-score of false-positives. In the latter case, a Gaussian distribution of responses is assumed to convert the probability that a response is greater than a given value to the number of standard deviations above or below the distribution's mean corresponding to that value. The complementary error function is needed because the integral from a value to infinity of a Gaussian probability distribution is simply the fraction of the area higher than that value, which is the probability of a response greater than that value.

For a Gaussian distribution of standard deviation, σ, the area below a value, r_0, is given from the definition of a Gaussian by

$$P(r < r_0) = \frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{r_0} e^{-\frac{x^2}{2\sigma^2}}\,dx \qquad \text{Eq. 3.29}$$

which becomes

$$P(r < r_0) = \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\frac{r_0}{\sqrt{2}\sigma}} e^{-x^2}\,dx = 1 - \frac{1}{2}\text{erfc}\left[\frac{r_0}{\sqrt{2}\sigma}\right]. \qquad \text{Eq. 3.30}$$

So to calculate the probability, p, that the response, r, is greater than a number of standard deviations, z, from the mean, we write $z = \frac{r_0}{\sigma}$,

$$p = P(r < \sigma z) = 1 - \frac{1}{2}\mathrm{erfc}\left[\frac{z}{\sqrt{2}}\right]. \qquad \text{Eq. 3.31}$$

From Eq. 3.29 the inverse transformation is needed to calculate z from the probability, p:

$$z = \sqrt{2}\,\mathrm{inverfc}[2(1-p)]. \qquad \text{Eq. 3.32}$$

where inverfc is the inverse of the complementary error function.

1. Sherrington CS. *The Integrative Action Of the Nervous System.* New York: Schribner & Sons; 1906.
2. Spillmann L. Receptive fields of visual neurons: the early years. *Perception.* 2014;43(11):1145-1176.
3. Nobili R, Mammano F, Ashmore J. How well do we understand the cochlea? *Trends Neurosci.* 1998;21(4):159-167.
4. Urban NN. Lateral inhibition in the olfactory bulb and in olfaction. *Physiol Behav.* 2002;77(4-5):607-612.
5. Windhorst U. On the role of recurrent inhibitory feedback in motor control. *Prog Neurobiol.* 1996;49(6):517-587.
6. Maler L. Neural strategies for optimal processing of sensory signals. *Prog Brain Res.* 2007;165:135-154.
7. Helmstaedter M, Sakmann B, Feldmeyer D. Neuronal correlates of local, lateral, and translaminar inhibition with reference to cortical columns. *Cereb Cortex.* 2009;19(4):926-937.
8. Hirsch JA, Gilbert CD. Synaptic physiology of horizontal connections in the cat's visual cortex. *J Neurosci.* 1991;11(6):1800-1809.
9. Hubel DH, Wiesel TN. Receptive fields of single neurones in the cat's striate cortex. *J Physiol.* 1959;148:574-591.
10. Hubel DH, Wiesel TN. Republication of The Journal of Physiology (1959) 148, 574-591: Receptive fields of single neurones in the cat's striate cortex. 1959. *J Physiol.* 2009;587(Pt 12):2721-2732.
11. Ringach DL. Mapping receptive fields in primary visual cortex. *J Physiol.* 2004;558(Pt 3):717-728.
12. Rodieck RW, Stone J. Analysis of receptive fields of cat retinal ganglion cells. *J Neurophysiol.* 1965;28(5):832-849.
13. Rodieck RW, Stone J. Response of cat retinal ganglion cells to moving visual patterns. *J Neurophysiol.* 1965;28(5):819-832.

14. Sharpee TO. Computational identification of receptive fields. *Annu Rev Neurosci.* 2013;36:103-120.
15. Buno W, Jr., Bustamante J, Fuentes J. White noise analysis of pace-maker-response interactions and non-linearities in slowly adapting crayfish stretch receptor. *J Physiol.* 1984;350:55-80.