

Population Coding

“ Now listen to me closely, young gentlemen. That brain is thinking. Maybe it’s thinking about music. Maybe it has a great symphony all thought out or a mathematical formula that would change the world or a book that would make people kinder or the germ of an idea that would save a hundred million people from cancer. This is a very interesting problem, young gentlemen, because if this brain does hold such secrets, how in the world are we ever going to find out?”

Dalton Trumbo, Johnny Got His Gun

We have seen how activities such as hand motion can be represented by the firing rates of a population of cells and the population vector method gives a simple procedure for recovering (or inferring) this motion from observed firing rates. In its most general form the population vector method can be written as

$$\hat{x} = \kappa \sum_i r_i \bar{x}_i \quad (1)$$

where x is some quantity represented by the cells (e.g. hand position or velocity), \hat{x} is the estimated quantity, κ is some scaling or normalizing constant, r_i is the firing rate of cell i and \bar{x}_i is the quantity the cell encodes (e.g. its preferred direction).

Until now, this equation has been presented as a simple, intuitive, model that, with some tuning of κ , produces reasonable reconstructions of the subject’s action. But how do we arrive at such a model? How good is it? What assumptions does it make? By understanding the answers to these questions, or hope is to be able to formulate better models with more principled assumptions and that lead to more accurate reconstruction.

Bayesian Inference

Let us start by posing the problem we are trying to solve in a general probabilistic framework. Let $p(X = x | \mathbf{R} = \mathbf{r})$ represent the probability that the state of the system is x given that the firing rates of all the cells are $\mathbf{r} = [r_1, \dots, r_n]^T$. X and \mathbf{R} are random variables that can take on different values and $p(X|R)$ represents the entire probability distribution over these values. For notational simplicity we will simply write $p(X = x | \mathbf{R} = \mathbf{r})$ as $p(x|\mathbf{r})$.

If we can formulate this probability distribution then we can do a variety of things such as find the value of x that maximizes the probability given \mathbf{r} or we can find the expected value of x . The problem remains however as to how to formulate such a probabilistic model. Often it is convenient to use the laws of probability theory to rewrite this equation. Using *Bayes' rule* we rewrite it as

$$p(x|\mathbf{r}) = \frac{p(\mathbf{r}|x)p(x)}{p(\mathbf{r})}. \quad (2)$$

The first thing to notice is that if we are interested in estimating something about x then the denominator, $p(\mathbf{r})$ on the right hand side is not going to be relevant; it is constant with respect to x . Formally, this normalizing term can be computed by marginalizing out x

$$p(\mathbf{r}) = \int_x p(\mathbf{r}|x)p(x)dx.$$

In general, it does not need to be estimate explicitly but rather is taken to be a constant such that

$$\int_x p(x|\mathbf{r})dx = 1.$$

Bayes' Rule

Bayes' rule is derived from a simple law of probability stating that the joint probability of two random variables A and B can be written as

$$p(A, B) = p(A|B)p(B)$$

or

$$p(A, B) = p(B|A)p(A).$$

From these we have

$$p(A|B)p(B) = p(B|A)p(A)$$

or Bayes' rule

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}.$$

Likelihood

Bayes rule allows us to represent $p(x|\mathbf{r})$ as the product of two terms which have special significance. The first, $p(\mathbf{r}|x)$, represents the *likelihood* of observing the data (firing rates), \mathbf{r} , given the state x . If you think of x as a control knob which can be tuned to different values, then the likelihood tells us the probability of observations given a particular parameter setting.

Prior

The second term in the numerator, $p(x)$, is called the *prior*. This represents the *a priori* probability of x ; that is, the probability prior to making any measurements. This term is used to represent information we may know about x that is independent of the firing rates \mathbf{r} . A particularly important source of prior information will be temporal. Often we want to estimate x at some particular instant in time t where we know the value of x at the previous time instant $t - 1$. We can write this temporal prior as

$$p(x_t) = \int p(x_t|x_{t-1})p(x_{t-1})dx_{t-1}.$$

We will return to this later.

Posterior

Finally, $p(x|\mathbf{r})$ is called the *posterior* probability; that is, the probability of x after taking into account the measurements. In Latin, *a posteriori* means “reasoning from observed facts back to their causes” (Microsoft Encarta World English Dictionary). The key here is that Bayes’ rule formalizes the “reasoning” process. It tells us how to make inferences that take into account both the observed data and our prior knowledge about the world.

We will spend the rest of the course essentially exploring this idea and how to make it practical for computers to infer activity from brains. Below we start with the likelihood term and eventually will see how the population vector method can be thought of in this Bayesian framework. Right now you should be wondering what the simple population vector algorithm means in terms of a likelihood and a prior. Hold that thought...

The Likelihood

One of the benefits of a Bayesian approach to the decoding problem is that it forces us to make our assumptions explicit. In modeling the likelihood term we will start with some simple assumptions and later in the course explore how to make them more realistic.

Conditional Independence

First we can write the likelihood of the state x given the firing, \mathbf{r} , of n cells as

$$p(r_1, r_2, \dots, r_n | x).$$

If we know x and the firing rate of a cell depends on x , independent of the firing rates of other cells, then we may assume that the probability of firing for the different cells, *conditioned on* x , are independent. Formally, if

$$p(A|B, C) = p(A|B)$$

then A and C are said to be conditionally independent given B . In other words, once we know B , knowing C does not give any more information. Note that this does not imply the stronger statement that A and C are independent which would imply $p(A, C) = p(A)p(C)$. In our case, we only assume that if we know the state x that the firing rates of the cells are conditionally independent:

$$p(r_1, r_2, \dots, r_n | x) = p(r_1 | x, r_2, \dots, r_n) p(r_2, \dots, r_n | x)$$

$$\begin{aligned}
&= p(r_1|x)p(r_2, \dots, r_n|x) \\
&= p(r_1|x)p(r_2|x)p(r_3, \dots, r_n|x) \\
&= \prod_{i=1}^n p(r_i|x).
\end{aligned}$$

This assumption means that the joint likelihood over all the cell's firing rates is simply represented as the product of the individual likelihoods. Note that this simplification is just an approximation. Since the cells we record from may influence each other, their firing rates may well be dependent on each other.

Generative Model

To specify the likelihood function it is useful to explicitly state our *model* of how cells encode the state, x_t , at time t in terms of their activity. In particular, we specify a *generative model* of, for example, firing rate at time t for cell i as

$$r_{i,t} = f_i(x_t) + \eta$$

where $f_i(\cdot)$ specifies some function that takes the state and produces the expected neural firing and η is some noise which suggests that our model f_i is uncertain or that the data is stochastic.

Once again, we are making our assumptions explicit and throughout the course we will consider a number of different generative models with different models (or *tuning functions*) and different noise properties.

For now consider a simple tuning function in which cells have some preferred state \bar{x}_i at which they fire maximally and that the activity drops off as the current state x_t differs more from the preferred state. This can be captured using a simple Gaussian tuning function

$$f_i(x_t) = k e^{-\frac{(x_t - \bar{x}_i)^2}{2\sigma^2}}$$

where k is a normalizing constant that is $1/(\sqrt{2\pi}\sigma)$ if $f_i(x)$ is a probability distribution (which integrates to 1) but more generally can be scaled so that $f_i(\cdot)$ produces a predicted firing rate (rather than a probability).

While $f_i(\cdot)$ models how the firing rate varies with x , it does not say anything about the noise we expect in our observations, $r_{i,t}$. If the firing rate is estimated in relatively short time bins ($\leq 200ms$), then this variation the activity is well modeled by a Poisson distribution

$$p(r_{i,t}|x_t) = \frac{1}{r_{i,t}!} e^{-f_i(x_t)} f_i(x_t)^{r_{i,t}}$$

where, recall $r_{i,t}$ is an integer as it is the observed firing at time t (computed in some finite time bin). We call this an *inhomogeneous* Poisson model since the mean firing rate, $f_i(x_t)$ varies over time according to the state as specified by the tuning function.

Now combined with the conditional independence assumption we have fully specified the likelihood as

$$p(\mathbf{r}|x_t) = \prod_{i=1}^n e^{-f_i(x_t)} \frac{f_i(x_t)^{r_{i,t}}}{r_{i,t}!}.$$

Inference

Inference often involves extracting some value (or values) from $p(\mathbf{r}|x)$. There are a number of possibilities. A common approach involves computing a *maximum likelihood* estimate of x

$$\hat{x}_{ML} = \operatorname{argmax}_x p(\mathbf{r}|x).$$

This is only a Bayesian estimate if the prior, $p(x)$ is uniform.

Alternatively we can seek the *maximum a posteriori*, or *MAP*, estimate

$$\hat{x}_{MAP} = \operatorname{argmax}_x p(x|\mathbf{r})$$

which takes into account both the likelihood and the prior and where

$$p(x_t|\mathbf{r}) \propto \prod_{i=1}^n e^{-f_i(x_t)} \frac{f_i(x_t)^{r_{i,t}}}{r_{i,t}!} p(x_t).$$

Maximizing $p(x|\mathbf{r})$ is equivalent to minimizing its negative logarithm. For the case described above with a Gaussian tuning function and Poisson noise, if we further assume a uniform prior then we can write the $-\log$ as

$$-\log p(x|\mathbf{r}) = \sum_{i=1}^n f_i(x) - \sum_{i=1}^n r_i \log(f_i(x)) - k_1$$

where $k_1 = -\sum_i \log(r_i!)$ is a constant independent of x .

Since the tuning function is Gaussian, this further simplifies to

$$-\log p(x|\mathbf{r}) = \sum_{i=1}^n f_i(x) + \frac{1}{2\sigma^2} \sum_{i=1}^n r_i (x - x_i)^2 - k_2$$

where k_2 now include both k_1 and the log of the constant scale multiplying the tuning function.

We further assume that the space of possible values of x is evenly represented by the population of cells we are recording from. This is the case in primary motor cortex where cells representing movement direction appear randomly distributed and where we can record from a sufficiently large population. When this assumption holds $\sum_{i=1}^n f_i(x)$ will be roughly constant, independent of x . Therefore it need not be considered in the minimization of $-\log p(x|\mathbf{r})$.

To minimize $-\log p(x|\mathbf{r})$ we take its derivative with respect to x , set it equal to zero, and solve for x

$$\frac{d}{dx} -\log p(x|\mathbf{r}) = \frac{1}{2\sigma^2} \sum_{i=1}^n 2r_i(x - x_i) = 0.$$

Simplifying gives

$$\sum_{i=1}^n r_i x - \sum_{i=1}^n r_i x_i = 0$$

or finally

$$x = \frac{\sum_{i=1}^n r_i x_i}{\sum_{i=1}^n r_i}.$$

This is simply a scaled version of the population vector result.

Summarizing, the *ad hoc* population vector method can be understood as embodying a number of implicit assumptions. These are

- conditional independence of the firing rates of cells,
- cells have a preferred value, x_i , and a Gaussian tuning function,
- the generative model assumes an inhomogeneous Poisson process,
- the prior probability of x is uniform,
- the cells are sampled uniformly with respect to their encoding,
- there exists a single best (MAP or ML) estimate of x .

Note there is also an inconsistency between the original assumption of cosine tuning for cells in motor cortex and the actual assumption of Gaussian tuning. They are similar but not the same.

We argue that this analysis is useful because it makes explicit what was hidden implicitly in the algorithm. By making the assumptions explicit we can begin to see how to improve on them. We will do so one by one in the coming weeks.