


86-631/42-631 Neural Data Analysis
Lecture 10: Estimation and classification I

- 1) Slides – locked in syndrome.  **ALS**
- 2) MLE
 - a. For parameter estimation
 - b. For linear regression
 - c. For categorical decoding.

Maximum Likelihood Estimation

Most of you have performed maximum likelihood estimation without even realizing it. Every time you estimate the mean of a distribution: by summing all the values and dividing by N, you are taking a maximum likelihood estimate. Here's why:

We've talked about a lot of different probability distributions in this class: Gaussian, Poisson, Binomial, etc. One thing many of them have in common is that they have some kind of free parameter: μ and σ for the Gaussian distribution, λ for the Poisson distribution, and q for the Binomial distribution. Let's say we are given a bunch of measurements which we believe correspond to one of these distributions: like we record whether or not a neuron spikes on several repetitions of a stimulus – we know this follows the Binomial distribution – but what is the appropriate value of q ?

Sure, sure. We know the average number of spikes we observe should be nq , so if we count the number we did observe, (say, k), we could just divide by n to get q . But how do we know this is the *best* answer? *Why* is this the answer?

Okay, so we observed k responses. What is the probability of having observed them?

$$p(k) = \binom{n}{k} q^k (1-q)^{n-k}$$

Binomial
↙

Now, we already know what k is, and we presumably know what n is – all we need is q . We could look at this in another way:

$$L(q) = \binom{n}{k} q^k (1-q)^{n-k}$$

All we've done is taken the probability, and now we look at it as a function of q instead of a function of k ! When we look at it this way, we call it the **likelihood function**. The likelihood function tells us how *likely*, or *probable*, are the data we observe for different values of assumed q .

To find q , we could try to *maximize* the likelihood. If all we care about is the maximum (or, at least, which values of q are more probable and which are less probable), then we don't care about multiplicative factors that don't depend on q – they don't affect the outcome. So, the n choose k term can be neglected. Really, we say $L(q) \propto p(k; q)$.

Let's consider the case where $n=10$ and $k=6$. (Out of 10 trials, our neuron spiked on 6 of them.) What q best describes the data? Well, when $q=0$, $L(q)=0$. Similarly, when $q=1$, $L(q)=0$. This makes sense – if q were zero we wouldn't have observed *any* spikes, and if q were one we wouldn't have observed any failures. So, q must be in the middle. How about $q=0.5$? Then $L(q)=q^k(1-q)^{n-k}=(1/2)^{10}=1/1024$. Better than 0 or 1, but is it the best? Let's plug it into Matlab and find out.

See slides of the likelihood of a binomial.

pmf \rightarrow discrete (sum=1)
pdf \rightarrow continuous ($\int_0^1 = 1$)
 $L(x) \rightarrow$ continuous (doesn't have to
sum to 1)

Okay, so Matlab tells us the answer: for $n=10$, $k=6$, the best q is 0.6, and for $n=10$, $k=7$, the best q is 0.7. We could have solved this analytically by maximizing the likelihood by hand: find where $dL/dq=0$, then verify that this is in fact a maximum.

Now, one thing that will make our lives a lot easier: finding the maximum of a function is the same as finding the maximum of the log of that function. Why? Because the log is a monotonically increasing function. The same would be true for finding the maximum of, say, the sqrt of the function or the exponential of the function. However, it makes our lives easier because the log of a product is the sum of logs...

$$\arg \max \{ f(x) \} = \arg \max \{ \log(f(x)) \}$$

$$L(q) = \binom{n}{k} q^k (1-q)^{n-k}$$

$$l(q) = \log \left(\binom{n}{k} q^k (1-q)^{n-k} \right) = \log \left(\binom{n}{k} \right) + k \log(q) + (n-k) \log(1-q)$$

$$\frac{dl(q)}{dq} = 0 \Rightarrow \frac{k}{q} - \frac{(n-k)}{1-q} = 0 \Rightarrow k(1-q) = q(n-k) \Rightarrow k = qn - kq + kq$$

$$\frac{dl(q)}{dq} = 0 \Rightarrow q = \frac{k}{n}$$

And it's not hard to see that $l''(q=k/n) < 0$, proving this is a maximum.

Okay, so we've proven what we really already knew to be true. So what? The cool part is that this approach is so very general – we can use it in all sorts of situations.

Mean of the Normal Distribution:

Say we measure N responses of a neuron in some task (like, for example, a delayed reaching task – we present the target, so the monkey knows where he's going to reach, but we don't give him the go cue yet indicating that he's allowed to initiate the reach). When we record from neurons in a particular area of the brain (PMd), they show *plan* activity – their activity ramps up (or down) after they've seen the target but before they reach. And the activity change for each neuron depends on the target direction!

(see slides)

So, we record N responses of one neuron to a *particular* target. If we assume these responses come from a Gaussian distribution, what's the mean of the distribution?

If the responses are independent, then

$$P(X_1 \dots X_N; \mu, \sigma) = P(X_1; \mu, \sigma) P(X_2; \mu, \sigma) \dots P(X_N; \mu, \sigma)$$

① Find probability distribution

$$P(X_i; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

② Recognize that this probability distribution is the likelihood function!

So, the likelihood function is:

independent, so can
multiply \rightarrow product
 $\prod_{i=1}^N$

$$L(\mu) = \prod_{i=1}^N \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

Therefore, the log-likelihood function is:

$$l(\mu) = \log \left(\prod_{i=1}^N \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}} \right) = \sum_{i=1}^N \log \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}} \right)$$

$$l(\mu) = \sum_{i=1}^N \log \left(\frac{1}{\sigma\sqrt{2\pi}} \right) - \frac{(x_i - \mu)^2}{2\sigma^2}$$

$$l(\mu) = N \log \left(\frac{1}{\sigma\sqrt{2\pi}} \right) - \frac{1}{2\sigma^2} \sum_{i=1}^N (x_i - \mu)^2 = N \log \left(\frac{1}{\sigma\sqrt{2\pi}} \right) - \frac{1}{2\sigma^2} \sum_{i=1}^N x_i^2 - 2x_i\mu + \mu^2$$

③ Maximize the $l(\mu)$ or $\log l(\mu)$

$$\frac{dl(\mu)}{d\mu} = 0 \Rightarrow \frac{1}{2\sigma^2} \sum_{i=1}^N -2x_i + 2\mu = 0$$

$$\frac{dl(\mu)}{d\mu} = 0 \Rightarrow \sum_{i=1}^N 2x_i = 2N\mu \Rightarrow \mu = \frac{1}{N} \sum_{i=1}^N x_i$$

We've re-derived the sample mean estimator! In other words, setting our *estimate* of μ , $\hat{\mu}$, equal to the sum of the x 's divided by N , is the most *probable* result given the data. It's the *likeliest*.

Another way of stating this: The sample mean is a Maximum Likelihood Estimator.

Linear regression as MLE:

Even linear regression is an example of maximum likelihood estimation!

Linear regression: you have some stimulus x , and some response y . (Say, x could be upcoming movement velocity in the horizontal direction, y could be firing rate.) You observe a bunch of data taken with different x 's. You want to find the best linear relationship between y and x :

$$y_i = b_0 + \beta x_i$$

So, how do we do the regression? Well, we observed some data when we ran the experiment to fit the model, so we know some of the y 's and some of the x 's. We also know what the y 's *should* be – we have

1. Model
2. Recognize pdf
3. Maximize it!

a guess from our linear model: y_i should be $b_0 + \beta x_i$, on average. In ordinary least-squares regression we minimize the difference between our observed data and our guesses. That is, we solve:

$$\arg \min_{b_0, \beta} \left(MSE = \frac{1}{N} \sum_{i=1}^N (y_i - b_0 - \beta x_i)^2 \right)$$

And we remember from last class that this yields

$$b_0 = \bar{y} - \beta \bar{x} \quad \text{and} \quad \beta = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^N (x_i - \bar{x})^2}$$

Or... we could do MLE! What if we had the following model:

$$y_i = b_0 + \beta x_i + \varepsilon_i \quad \varepsilon_i \sim N(0, \sigma^2) \quad \leftarrow \text{generative model}$$

This is the same as before, except we included ε . That is, we are specifically saying that y is equal to a linear function of x plus some unknown **Gaussian noise**. We need to include the ε , because we know that our firing rate isn't going to exactly mirror the movement: firing rates have noise. ε captures this noise process. Assuming the noise is Gaussian is a standard technique in the field, though there are extensions that assume the noise is distributed in other ways, like Poisson. Now, the x 's are *known*. So, what's the distribution of Y ?

$$Y \sim N(b_0 + \beta X, \sigma^2) \quad y_i \sim N(b_0 + \beta x_i, \sigma^2)$$

So, the distribution of N independent observations of Y (y_i) is:

$$\textcircled{1} \quad P(y_i; x_i, b_0, \beta) = \prod_{i=1}^N \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(y_i - b_0 - \beta x_i)^2}{2\sigma^2}}$$

This has corresponding likelihood function

$$\textcircled{2} \quad L(b_0, \beta) = \prod_{i=1}^N \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(y_i - b_0 - \beta x_i)^2}{2\sigma^2}}$$

And log-likelihood function

$$l(b_0, \beta) = N \log \left(\frac{1}{\sigma \sqrt{2\pi}} \right) - \sum_{i=1}^N \frac{(y_i - b_0 - \beta x_i)^2}{2\sigma^2}$$

Differentiating w.r.t. b_0 gives us ...

Go through this proof!!

$$l(b_0, \beta) = N \log \left(\frac{1}{\sigma \sqrt{2\pi}} \right) - \sum_{i=1}^N \frac{(y_i - b_0 - \beta x_i)^2}{2\sigma^2}$$

$$\textcircled{3} \quad \frac{\partial l(b_0, \beta)}{\partial b_0} = 0 \Rightarrow \sum_{i=1}^N \frac{2(y_i - b_0 - \beta x_i)}{2\sigma^2} = 0 \Rightarrow \sum_{i=1}^N (y_i - b_0 - \beta x_i) = 0$$

Look familiar?

$$\frac{\partial l(b_0, \beta)}{\partial b_0} = 0 \Rightarrow b_0 = \bar{y} - \beta \bar{x}$$

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$$

Therefore:

$$l(\beta) = N \log \left(\frac{1}{\sigma \sqrt{2\pi}} \right) - \sum_{i=1}^N \frac{(y_i - \bar{y} + \beta \bar{x} - \beta x_i)^2}{2\sigma^2} = N \log \left(\frac{1}{\sigma \sqrt{2\pi}} \right) - \sum_{i=1}^N \frac{(y_i - \bar{y} - \beta(x_i - \bar{x}))^2}{2\sigma^2}$$

and

$$\frac{dl(\beta)}{d\beta} = 0 \Rightarrow \sum_{i=1}^N \frac{2(x_i - \bar{x})(y_i - \bar{y} - \beta(x_i - \bar{x}))}{2\sigma^2} = 0 \Rightarrow \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y} - \beta(x_i - \bar{x})) = 0$$

$$\frac{dl(\beta)}{d\beta} = 0 \Rightarrow \beta = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})}$$

All this is just meant to show how ubiquitous Maximum Likelihood Estimation is – it's pretty much behind every analysis technique we do.

OLS regression is ~ to solving MLE
under assumption that the noise is
independent and normally distributed.

$$\left. \begin{array}{l} y_i = \alpha + \beta x_i + \epsilon_i \\ \epsilon_i \sim N(0, \sigma^2) \\ y_i \sim \text{Pois}(\lambda_i) \\ x_i = \alpha + \beta x_i \end{array} \right\} \begin{array}{l} \text{Extensions to} \\ \text{linear regression} \\ \text{w/ MLE} \end{array}$$

An application of MLE – BCI spelling devices:

So how does this help us solve the original problem: building a spelling device for our locked-in patient?

Well, it turns out neurons in a certain region of the brain (dorsal pre-motor cortex) are tuned to the direction of the upcoming reach, even before it occurs. They respond to “planned reaches”.

(See slides.)

Let’s take the firing rate we record for the 200 ms window before the trial starts (during delay). We can take repeated reaches to a single target and fit a probability distribution (say, a Gaussian) to that data. That is, we fit μ_θ and σ_θ for every target θ . In fact, we can do this for every neuron. Thus, we know that the probability of seeing a certain firing rate y from a particular neuron i would be:

$$P(y; \mu_{i,\theta}, \sigma_{i,\theta}) = \frac{1}{\sigma_{i,\theta} \sqrt{2\pi}} e^{-\frac{(y - \mu_{i,\theta})^2}{2\sigma_{i,\theta}^2}}$$

(See slides.)

You can fit one of these probability distributions for each neuron. Then, when you record a sample of neural activity from each neuron (during the delay period, say), you could assume they are independent, so you get:

$$P(\{y\}; \{\mu_\theta\}, \{\sigma_\theta\}) = \prod_{i=1}^N \frac{1}{\sigma_{i,\theta} \sqrt{2\pi}} e^{-\frac{(y_i - \mu_{i,\theta})^2}{2\sigma_{i,\theta}^2}}$$

This gives you the log-likelihood function:

$$l(\theta) = \sum_{i=1}^N \log \left(\frac{1}{\sigma_i(\theta) \sqrt{2\pi}} \right) - \frac{(y_i - \mu_i(\theta))^2}{2\sigma_i^2(\theta)}$$

Now, this can’t be differentiated w.r.t. θ : b/c θ is discrete, not continuous? So, how do we solve?

Answer: we compute $l(\theta)$ for every value of θ and pick the one that’s highest!

Now, we’ve made some assumptions. First, we’ve assumed that each neuron is independent. Second, we’ve assumed Gaussian noise. We could still assume that neurons are independent, and instead use a Poisson noise model, where the mean firing rate λ depends on the target θ :

$$P(y; \lambda_i(\theta)) = \frac{\lambda_i^y(\theta) e^{-\lambda_i(\theta)}}{y!}$$

$$P(\{y\}; \{\lambda(\theta)\}) = \prod_{i=1}^N \frac{\lambda_i^y(\theta) e^{-\lambda_i(\theta)}}{y!}$$

$$l(\theta) = \sum_{i=1}^N y_i \log(\lambda_i(\theta)) - \log(y_i!) - \lambda_i(\theta)$$

Note the $\log(y_i!)$ term doesn't depend on θ , so it can be ignored. This leaves you with:

$$l(\theta) = \sum_{i=1}^N y_i \log(\lambda_i(\theta)) - \lambda_i(\theta)$$

Which again can be solved for each value of θ directly. Choose the θ that gives the maximum!

Finally, you can use the Gaussian noise model but relax the independence assumption between neurons. Instead, you can assume that the spike count vector follows a multivariate Gaussian distribution for each target:

$$P(\vec{y}; \vec{\mu}(\theta), \Sigma(\theta)) = (2\pi)^{-\frac{N}{2}} \det(\Sigma(\theta))^{-\frac{1}{2}} e^{-\frac{1}{2}(\vec{y} - \vec{\mu}(\theta))^T \Sigma^{-1}(\theta)(\vec{y} - \vec{\mu}(\theta))}$$

Here, you not only need a mean and variance of each neuron for each target, you need the covariance between every pair of neurons for each target. Takes a lot more data to fit, but certainly do-able.

$$l(\theta) = \log \left(\det(\Sigma(\theta))^{-\frac{1}{2}} \right) - \frac{1}{2} (\vec{y} - \vec{\mu}(\theta))^T \Sigma^{-1}(\theta) (\vec{y} - \vec{\mu}(\theta))$$

And again plug all values in and find the θ that maximizes. Shenoy and co. did this for the video. (I don't know if they ever actually relaxed the independence assumption or not.)

(Go back to slides.)