86-631/42-631 Neural Data Analysis
Lecture 11: Estimation and Classification II

1) Recap of MLE
2) Priors
3) MAP
4) Naïve Bayes classifier.
5) Relationship between ML, MAP, and Naïve Bayes
6) Sequential Estimation
7) **Slides: Bayesian integration in sensorimotor learning (Kording and Wolpert)**

Recap: Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) is a method for estimating the parameter(s) of a probability distribution, when you are given data drawn from that distribution. Here are the steps:

Let Y be a random variable that follows some distribution with some unknown parameter θ. For example, if Y were a Poisson random variable, θ would be λ. Let P(Y=y; θ) be the probability that the random variable Y takes on value y with parameter θ.

1) Write down the likelihood function L(θ) = P(Y=y;θ).
2) To get $\hat{\theta}_{MLE}$, the MLE estimate of θ, solve

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}}\, L(\theta)$$

3) Note that the solution will be the same as solving for the maximum of the log of L. That is, if $\ell(\theta) = \log\big(L(\theta)\big)$, then

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}}\, \ell(\theta)$$

Interpretation: we are searching for the value of θ that makes the data we observed most probable. This is, in some sense, the *likeliest* value of θ.

Example:

Suppose you are observing the inter-arrival times of the action potentials from a neuron you are recording. You believe (1) that your measurements are independent from one another, and (2) that each inter-arrival time is drawn from an Exponential(λ) distribution (the same λ for each measurement). Find the MLE of λ.

Recall that the pdf of the exponential distribution is

$$f(t;\lambda) = \lambda e^{-\lambda t_i}$$

Because all of the measurements are independent, the joint pdf is just the product of the marginals.

$$L(\lambda) = f(t_1, \dots, t_N; \lambda) = \prod_{i=1}^{N} \lambda e^{-\lambda t_i}$$

Taking the log yields:

$$\ell(\lambda) = \sum_{i=1}^{N} \log\big(\lambda e^{-\lambda t_i}\big) = \sum_{i=1}^{N} \log(\lambda) - \lambda t_i = N\log(\lambda) - N\lambda\bar{t}$$

where

$$\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i$$

Taking the derivative (wrt λ) and setting equal to 0 yields:

$$\frac{d\ell(\lambda)}{d\lambda} = 0 \rightarrow \frac{N}{\lambda} - N\bar{t} = 0 \rightarrow \lambda = \frac{1}{\bar{t}}$$

Priors: How do we do better than MLE?

(Start with slides, giving an example of how you could do better than MLE by considering the prior probability of each category.)

Bayes' Rule:

$$P(C|X) = \frac{P(C)P(X|C)}{P(X)}$$

Or posterior = (prior x likelihood)/evidence

Although this holds in general for any definition of C and X, here we specifically take C to be some categorical variable, and we take X to be some kind of data: a neural spike count, for example. The difference between MLE and this formulation is that the categorical variable is not just some unknown parameter, it is a *random variable*. As a random variable, it can have a probability in its own right. For example, in the English language we know that the letter 'e' occurs more frequently than any other letter. If you were reading English, and couldn't decipher a letter, what letter would you guess? 'e'.

Now, of course, we can extend Bayes' Rule to multiple pieces of data:

$$P(C|X_1,...,X_N) = \frac{P(C)P(X_1,...,X_N|C)}{P(X_1,...,X_N)}$$

What we are interested in calculating is the C that has the maximum probability after incorporating the data. This estimate of C is known as the *Maximum A Posteriori,* or *MAP* estimate.

Naïve Bayes' Classifier

Consider the numerator for a moment. This is essentially (switched notation from X to F, oops)

$$P(C)P(F_1,...,F_N|C) = P(C,F_1,...,F_N)$$

By the Chain Rule of Probability, we can write this as:

$$P(C,F_1,...,F_N) = P(F_N|C,F_1,...,F_{N-1})P(C,F_1,...,F_{N-1})$$

$$P(F_N|C,F_1,...,F_{N-1})P(C,F_1,...,F_{N-1}) = P(F_N|C,F_1,...,F_{N-1})P(F_{N-1}|C,F_1,...,F_{N-2})P(C,F_1,...,F_{N-2})$$

$$= P(F_N|C,F_1,...,F_{N-1})P(F_{N-1}|C,F_1,...,F_{N-2})P(F_{N-2}|C,F_1,...,F_{N-3})P(C,F_1,...,F_{N-3})$$

…

$$= P(F_N|C,F_1,...,F_{N-1})P(F_{N-1}|C,F_1,...,F_{N-2})P(F_{N-2}|C,F_1,...,F_{N-3})...P(F_1|C)P(C)$$

Or, in reverse order

$$P(C,F_1,...,F_N) = P(C)P(F_1|C)P(F_2|C,F_1)...P(F_N|C,F_1,...,F_{N-1})$$

The Naïve Bayes classifier makes the very strong (naïve) assumption that all of the features are conditionally independent.

Thus, for all i≠j:

$$P(F_i|C,F_j) = P(F_i|C)$$

And we are left with:

$$P(C,F_1,...,F_N) = P(C)P(F_1|C)P(F_2|C)...P(F_N|C)$$

or

$$P(C,F_1,...,F_N) = P(C)\prod_{i=1}^{N} P(F_i|C)$$

Why make this assumption? Because you've dramatically reduced the number of parameters you need to fit. Instead of trying to estimate all the values $P(F_N|C,F_1,\ldots,F_{N-1})$ can take on, for each and every value of X_2, X_3, etc - you can instead just estimate $P(X_i|C)$ for every X_i.

As an example, we talked last class about using MLE (Maximum Likelihood Estimation) to guess at which target was presented in the BCI decoding task. We said that we could assume every neuron would emit a certain number of spikes for a particular target as draws from a Normal distribution:

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

If all the neurons are independent, then

$$P(\vec{x}) = \prod_{i=1}^{N} \frac{1}{\sigma_{i,C}\sqrt{2\pi}} e^{-\frac{(x-\mu_{i,C})^2}{2\sigma_{i,C}^2}}$$

And all we have to do is estimate a μ and a σ for each neuron, for each target.

If the neurons aren't independent, however, then we have to write

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

Here, we'd have to also estimate the covariances between all the neurons *for every single target!*

Note: when C is a random variable, these equations become:

$$P(X|C) = \frac{1}{\sigma_i(C)\sqrt{2\pi}} e^{-\frac{(x-\mu_i(C))^2}{2\sigma_i^2(C)}}$$

That is, this is the *conditional* distribution, NOT the joint distribution. (What would the joint probability distribution be?) This is a ***CRUCIAL*** distinction. Similarly, when C is random, and the neurons are not independent, we have

$$P(X|C) = (2\pi)^{-\frac{N}{2}} \det(\boldsymbol{\Sigma}(C))^{-\frac{1}{2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}(C))^T \boldsymbol{\Sigma}^{-1}(C)(\vec{x}-\vec{\mu}(C))}$$

Okay, back to the main point. We have the equation:

$$P(C, X_1, ..., X_N) = P(C)\prod_{i=1}^{N} P(X_i|C)$$

And therefore, from Bayes' rule, we have:

$$P(C|X_1, ..., X_N) = \frac{P(C)\prod_{i=1}^{N} P(X_i|C)}{P(X_1, ..., X_N)}$$

Let's consider this equation. First, note that the denominator $P(X_1,\ldots X_N)$, is a constant for a given set of data. We don't need to worry about it, because it won't affect our computed value of C! In fact, these equations are often written as

$$P(C|X_1, ..., X_N) = \frac{1}{Z} P(C)\prod_{i=1}^{N} P(X_i|C)$$

So, the Naïve Bayes' Classifier is:

$$\hat{C} = \arg\max_{C} = P(C)\prod_{i=1}^{N} P(X_i|C)$$

Where P(C) is the prior probability (for example, the relative frequencies of the English letters), and $P(X_i|C)$ are the conditional probabilities of observing that particular firing rate when presented with that target.

Summary

Last class we learned about maximum likelihood estimation (MLE):

$$\hat{C}_{MLE} = \arg\max_{C} \left\{ P\left(\vec{X} \middle| C\right) \right\}$$

(Here, we write P(X|C) even if C is a parameter instead of a random variable. This is a fairly common thing to do.)

This class we learned about maximum a posteriori estimation (MAP estimation):

$$\hat{C}_{MAP} = \arg\max_{C} \left\{ P(C) P\left(\vec{X} \middle| C\right) \right\}$$

Note that ML and MAP estimation are the same when the prior is uninformative (uniform)!

Finally, we learned about the Naïve Bayes' Classifier, which assumes that these conditional probabilities are all independent. This is like saying that the neurons firing rates (given target direction) are all independent. This classifier is written as:

$$\hat{C}_{NB} = \arg\max_{C} \left\{ P(C) \prod_{i=1}^{N} P\left(X_i \middle| C\right) \right\}$$

Some weaknesses of MAP. It gets the *mode* of the distribution – not its mean or median. For a Normal distribution these are the same, but not for distributions in general.

Often hard to compute numerically.

Sequential estimation

Sequential estimation is a particularly nice application of the Naïve Bayes' estimator. Imagine a situation where your data flows in over time. (So, you look at spike rates over a 30ms window, then another non-overlapping window…)

If you imagine that spike rates at each time are conditionally independent (given C), then they satisfy the Naïve Bayes' assumption!

That is

$$P(C, F_1, ..., F_N) = P(C)P(F_1|C)P(F_2|C, F_1)...P(F_N|C, F_1, ..., F_{N-1})$$

Reduces to:

$$P(C, F_1, ..., F_N) = P(C)P(F_1|C)P(F_2|C)...P(F_N|C) = P(C)\prod_{i=1}^{N} P(X_i|C)$$

This means you could actually watch your estimate evolve over time…

$$\underbrace{\underbrace{P(C)P(F_1|C)}_{\hat{C}_1}P(F_2|C)...P(F_N|C)}_{\hat{C}_2}$$

Another way of saying this:

When the first piece of data comes in, you compute

$$P(C|X_{t1}) = \frac{P(C)P(X_{t1}|C)}{P(X)}$$

Then, when the second piece of data comes in, you use the posterior *as the new prior!* This is known as *sequential estimation*.

$$P(C|X_{t2}) = \frac{P(C_{t1})P(X_{t2}|C)}{P(X)}$$

It works especially well if (1) you have Gaussian distributions for both your prior and your likelihood terms. Then the posterior is also Gaussian. (When the prior and the posterior have the same form (though with potentially different parameters) it is said that you have the *conjugate prior* to your likelihood. The Gaussian is *self-conjugate.*

Furthermore, Bayes' rule doesn't just give you the class estimate, it gives you the whole probability distribution for C! So, for example, you could allow data to flow in until the probability for a particular C exceeds some threshold.