Chris Piech
CS 109

Section #9
Dec 3rd, 2025

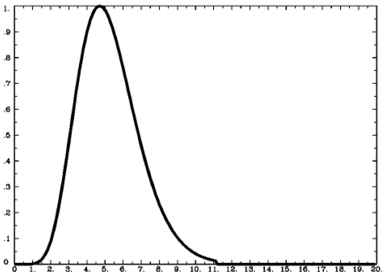
Section 9 Solutions

1 Decoding the brain (with MLE!)

What do people mean when they say, the brain lights up? In most cases, it is not possible to directly observe neurons firing in the human brain. However, it is possible to obtain a good proxy for neuronal activity with MRI scanners by measuring oxygen levels. When subjects undergo functional MRI experiments, they are exposed to some stimulus (such as images, videos, or words) while the scanner measures the oxygen levels in voxels, or tiny 1 mm cubes throughout their brain.

(Aside: the measurements from MRI scanners do not correspond to any standard unit, so MRI data is often given in terms of a.u., or arbitrary units. In this problem, dont worry about units!)

When a voxel responds to a stimulus, its oxygen levels follow the Hemodynamic Response Function (HRF). An example HRF is shown below:



1. The measurements from MRI scanners are subject to noise. This noise is Gaussian, i.e., the measured signal at any point in time can be modeled as a Normal distribution, where the mean is the true signal, and assume the standard deviation $\sigma = 2$.

   If, at some point in time, the true signal is 5, what is the probability density that the measured signal is 4.8? Simplify your answer as much as possible.

Let S be the measured signal so, $S \sim N(\mu = 5, \sigma = 2)$. Then, using the PDF for a Normal random variable:

$$
\begin{aligned}
f(4.8) &= \frac{1}{\sqrt{2\pi} \cdot 2} e^{-\frac{(4.8-5)^2}{2 \cdot 2^2}} \\
&= \frac{1}{2\sqrt{2\pi}} e^{-\frac{0.01}{2}}
\end{aligned}
$$

2. Suppose we showed a subject a stimulus three separate times, yielding three separate hemodynamic responses. For each response, we take a measurement at the peak of the signal. We get the following measurements: $x_1 = 5.0$, $x_2 = 5.1$, and $x_3 = 5.5$.

   If the true measurement at the peak of the response is μ, what is the probability density of observing these three measurements? Assume $\sigma = 2$, and that each measurement is I.I.D.

Again, let S be the measured signal where $S \sim N(\mu, \sigma = 2)$.

$$\begin{aligned} f(x_1, x_2, x_3) &= \Pi_{i=1}^3 f(x_i) \\ &= \Pi_{i=1}^3 \frac{1}{2\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\cdot 2^2}} \\ &= \Pi_{i=1}^3 \frac{1}{2\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{8}} \\ &= \frac{1}{2\sqrt{2\pi}} \Pi_{i=1}^3 e^{-\frac{(x_i-\mu)^2}{8}} \end{aligned}$$

3. Find the maximum likelihood estimation of μ.

Taking the log of the likelihood we calculated in part (2), we have:

$$\begin{aligned} LL(\mu) &= \log(\frac{1}{2\sqrt{2\pi}}) + \Sigma_{i=1}^3 (-\frac{(x_i - \mu)^2}{8}) \\ &= \log(\frac{1}{2\sqrt{2\pi}}) - \frac{1}{8}\Sigma_{i=1}^3 (x_i - \mu)^2 \end{aligned}$$

Now we need to take the partial derivative of the log likelihood w.r.t. μ:

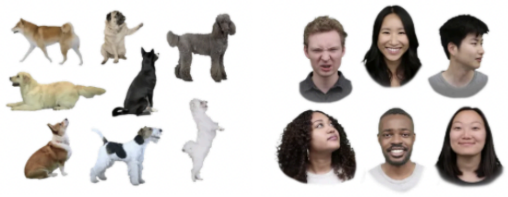
$$\begin{aligned} \frac{\partial LL(\mu)}{\partial \mu} &= -\frac{1}{8}\Sigma_{i=1}^3 \big(-2(x_i - \mu)\big) \\ &= \frac{1}{4}\Sigma_{i=1}^3 (x_i - \mu) \end{aligned}$$

Setting, the partial derivative to 0, we can find the MLE of μ.

$$\begin{aligned} \frac{1}{4}\Sigma_{i=1}^3 (x_i - \mu) &= 0 \\ \Sigma_{i=1}^3 (x_i - \mu) &= 0 \\ 3\mu &= \Sigma_{i=1}^3 x_i \\ \mu = \frac{1}{3}(5.0 + 5.1 + 5.5) &= 5.2 \end{aligned}$$

4. Researchers are often interested in what types of stimuli different brain regions will respond to. They can answer this question by studying the HRF; when a voxel responds more strongly to a certain stimulus, it will exhibit a heightened hemodynamic response.

   Suppose we have shown a subject two types of stimuli—faces and dogs—and we are interested in whether a certain voxel responds more strongly to faces than dogs. Assume we have measured the peak of the voxels hemodynamic response to 10 different face pictures $(f_1, f_2, \ldots, f_{10})$ and 10 different dog pictures $(d_1, d_2, \ldots, d_{10})$.



   Using bootstrapping, write a function in pseudocode to determine whether the voxels response to faces is higher than its response to dogs.

While solutions to this subproblem may vary, they should include the following key steps:

a. Combine $f_1, f_2, \ldots, f_{10}$ and $d_1, d_2, \ldots, d_{10}$ into one list.

b. Repeatedly draw pairs of samples, with replacement, from the list.

c. Record the number of instances where the difference between the means of sample A and sample B are at least as large as the difference in the means of the original samples.

d. If the percentage of hits is lower than some significance threshold (e.g. $\alpha = 0.05$), then we can reject the null hypothesis, i.e. that there is no difference in means of the two groups.

2 Timing Attack (23 points)

In this problem we will see how to crack a password in linear time by measuring how long the password check takes to execute (see code below).

```
# An insecure string comparison
def does_password_match(guess, password):
    n_guess = len(guess)
    n_password = len(password)
    if n_guess != n_password:
        return False                    # 4 lines executed to get here
    for i in range(n_guess):
        if guess[i] != password[i]:
```

```
        return False                # 6 + 2i lines executed to get here
    return True                     # 5 + 2n lines executed to get here
```

Assume that our server takes T ms to execute any line in the code where $T \sim N(\mu = 5, \sigma^2 = 0.5)$ milliseconds. The amount of time taken to execute a line is always independent of other lines.

On our site, all passwords only use lower case letters and are between 5 and 10 letters long, inclusive. A hacker is trying to crack the root password which is “gobayes” by carefully measuring how long the code takes to tell her that her guesses are incorrect.

a. (5 points) What is the distribution for the time it takes to execute k lines of code?

Let T_k be the amount of time to execute k lines. $T_k = \sum_{i=1}^{k} X_i$ where X_i is the amount of time to execute line i. $X_i \sim N(\mu = 5, \sigma^2 = 0.5)$. Since T_k is the sum of k independent normals:

$$T_k \sim N(\mu = 5k, \sigma^2 = 0.5k)$$

b. (7 points) First, the hacker needs to find the length of the password. What is the probability that the time taken to check a guess of correct length (when the server executes 6 lines) is longer than the time taken to check a guess of an incorrect length (when the server only executes 4 lines)? Assume the first letter of the guess does not match the password’s first letter. Hint: $P(A > B) = P(A - B > 0)$.

Time to run 6 lines of code: $T_6 \sim N(\mu = 30, \sigma^2 = 3)$
Time to run 4 lines of code: $T_4 \sim N(\mu = 20, \sigma^2 = 2)$ Then we apply a linear transform to T_4 so we can subtract it from T_6 by “adding” two Normal RVs.

$$\begin{aligned}
-T_4 &\sim N(\mu = -20, \sigma^2 = 2) \\
T_6 - T_4 &\sim N(\mu = 10, \sigma^2 = 5) \\
P(T_6 > T_4) &= P(T_6 - T_4 > 0) \\
&= 1 - F_{T_6 - T_4}(0) \\
&= 1 - \Phi\left(\frac{0 - 10}{\sqrt{5}}\right) \approx 1.0
\end{aligned}$$

c. (8 points) Now that our hacker knows the length of the password, to get the actual string, she will try to determine one letter at a time, starting with the first letter. To start, the hacker tries the string “aaaaaaa” and sees that it takes 27ms. Based on this timing, how much more probable is it that first character did not match (server executes 6 lines) than the first character did match (server executes 8 lines)? Assume that all letters in the alphabet are equally likely to be the first letter.

Let M be the event that the first letter matched. The problem's wording indicates that we are looking for a ratio between the probability of M^C to the probability of M, conditioned on observing a code execution time of 27ms. Let $T_?$ be the time it took to check the password, with the number of lines executed being unknown. To calculate this ratio, we can apply Bayes' Theorem:

$$\begin{aligned}\frac{P(M^C|T_? = 27)}{P(M|T_? = 27)} &= \frac{f(T_? = 27|M^C)P(M^C)}{f(T_? = 27|M)P(M)} \\ &= \frac{f(T_? = 27|M^C)\frac{25}{26}}{f(T_? = 27|M)\frac{1}{26}} \\ &= 25 \cdot \frac{f(T_? = 27|M^C)}{f(T_? = 27|M)}\end{aligned}$$

From here, we can reason about how $T_?$ is distributed if we know M or M^C. If M happened, then $T_? = T_8$; otherwise, $T_? = T_6$. So we can re-write the above as:

$$\begin{aligned}&= 25 \cdot \frac{f(T_6 = 27)}{f(T_8 = 27)} \\ &= 25 \cdot \frac{\frac{1}{\sqrt{6\pi}}e^{-\frac{(27-30)^2}{6}}}{\frac{1}{\sqrt{8\pi}}e^{-\frac{(27-40)^2}{8}}} \\ &= 25 \cdot \frac{\sqrt{8}}{\sqrt{6}} \cdot \frac{e^{-\frac{9}{6}}}{e^{-\frac{169}{8}}} \\ &\approx 9.6 \text{ million}\end{aligned}$$

d. (3 points) If it takes the hacker 6 guesses to find the length of the password, and 26 guesses per letter to crack the password string, how many attempts does she need to crack our password, "gobayes"? Yikes!

Since the password is length 7: $6 + 7 \cdot 26 = 188$.