Advanced Computational Neuroscience - Week 1
—
Drift Diffusion

Sam Suidman

February 14, 2022

1 Introduction

The drift diffusion model is important, because neurons often fire when the voltage ν gets larger than a threshold ν_{thr}. The firing of neurons can be modelled by a diffusion model. The time it then takes to fire is the first passage time (FPT), where the voltage ν for first time becomes larger than the threshold ν_{thr}.

2 Theory

There is a theoretical probability distribution for the FPT given by Formula 1.

$$\rho(t) = \frac{\nu_{thr}}{\sqrt{2\pi}\sigma t^{\frac{3}{2}}} e^{-\frac{(\nu_{thr}-\mu t)^2}{2\sigma^2 t}} \tag{1}$$

In the model the neuron starts at $\nu(0) = 0$ and will drift away from there. The input it receives each time step dt in this model has a mean drift μdt. Also it has some random Gaussian noise $d\xi$ with mean 0 and variance $\sigma\sqrt{dt}$. The experiment is done N trials. Hence, the voltage evolves according to:

$$\nu(t+dt) = \nu(t) + \mu dt + d\xi \tag{2}$$

It is also interesting to see what happens for different values of μ and σ. We then assume that the log likelihood is given by:

$$L = \sum_{i=1}^{n} \log \rho(t_i|\mu,\sigma) = n \log\frac{1}{\sigma} - \frac{3}{2}\sum_{i=1}^{n}\log t_i - \sum_{i=1}^{n}\frac{(1-\mu t_i)^2}{2\sigma^2 t_i} \tag{3}$$

From which we can derive the maximum likelihood for μ with:

$$\frac{\partial L}{\partial \mu} = -\sum_{i=1}^{n}\frac{2(1-\mu t_i)(-t_i)}{2\sigma^2 t_i} = \frac{1}{\sigma^2}(n - \mu\sum_{i=1}^{n} t_i) = 0 \tag{4}$$

$$\mu = \frac{1}{\bar{t}} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} t_i} \tag{5}$$

And for σ^2 with:

$$\frac{\partial L}{\partial \sigma^2} = \frac{\partial}{\partial \sigma^2}(-\frac{1}{2}n\log(\sigma^2) - \sum_{i=1}^{n}\frac{(1-\mu t_i)^2}{2\sigma^2 t_i}) = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(\frac{1}{t_i} - \mu) = 0 \tag{6}$$

$$\sigma^2 = f - \frac{1}{\bar{t}} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{t_i} - \mu \tag{7}$$

3 Results

Running the model I first used the following values: $N = 10.000$, $dt = 0.001$, $\mu = 0.1$, $\sigma = 0.1$, $\nu_{thr} = 1$, $T = 20$. To be able to run many trials N with small time steps dt I first created Gaussian noise of size N by $\frac{T}{dt}$. I created a cumulative sum afterwards together with the mean drift μ and then looked at the times where ν for first time passed ν_{thr}. Some example trials are shown in Figure 1 on the left. The FTP histogram is shown on the right. The histogram follows very much the theoretical description given by formula 1.

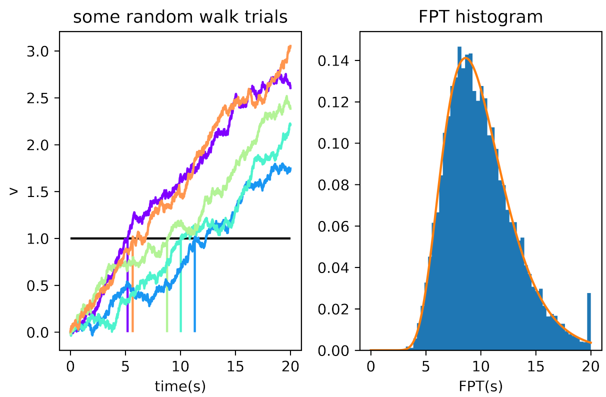

Figure 1: **Left** Some random walks with vercitcal lines representing the FPT's. **Right** FPT histogram with the theoretical distribution of Formula 1 shown as an orange line.

The second time I ran the model I used different values of μ and σ and compared the empirical values with the model values by looking at the fractions $\frac{\mu}{\mu_{exact}}$ and $\frac{\sigma}{\sigma_{exact}}$. This time I changed two parameters to $N = 1000$ and $dt = 0.01$, because otherwise the program takes too long to run for respectively 20 and 25 values for μ and σ both between 0.1 and 2. The results are shown in Figure 2. As you probably expected for the left Figure, the more noise (bigger σ) the more the fraction differs from 1. This is because the noise term $d\xi$ overshadows the μdt term. Furthermore this effect is bigger for smaller μ, because then it takes longer to reach the ν_{thr}. For the right Figure the fraction becomes smaller for bigger σ^2. This is because the system becomes more unpredictable for bigger σ^2.

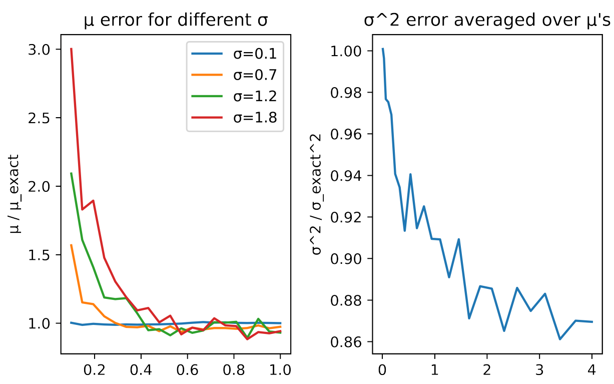

Figure 2: Empirical data of μ and σ compared with the model (exact) values. This comparison is done with the fractions $\frac{\mu}{\mu_{exact}}$ and $\frac{\sigma}{\sigma_{exact}}$. **Left** For different values of σ^2 the fraction is plotted. **Right** The fraction for σ^2 is averaged over μ.

4 Conclusion

To summarize, I found that the theoretical FPT distribution of formula 1 compares very good with the empirical data. In addition the error of μ and σ become bigger for a model with more noise represented by larger values of σ.

5 Appendix

The code to run this can be found in the attached Jupyter Notebook.