PHYS 414 Problem Set 1: Chaos, machine learning, and aliens

Problem 1: Trajectories, chaos, and the Markovian assumption

In this problem we will investigate one of the basic claims made in class: if we have sufficient randomization of a system over a timescale δt, the Markovian assumption should hold for transition probabilities between states of that system. In other words, the chance of observing a state at some time $t + \delta t$ will depend just on the state at time t, and not on the states at earlier times $t - \delta t$, $t - 2\delta t$, etc. In order to test this, we need a system that exhibits chaotic dynamics, since that is what ultimately allows for randomization. One of the simplest such chaotic systems is a special case of the so-called *logistic growth model*.

To understand the model, it is easiest to consider an ecological analogy: imagine a population of organisms whose population at the beginning of the ith time step is n_i. Each time step is a full generation, where the current population gives birth to offspring and dies off, so that at the next time step we have some population n_{i+1}. If each organism leaves behind 4 surviving offspring, we have $n_{i+1} = 4n_i$, and the population grows exponentially. More realistically, we might imagine that there are finite resources in the environment, so that it can sustain at most K organisms—this is known as the carrying capacity of the environment. The closer the population n_i is to K, the fewer resources there are for offspring to survive, which we can represent by modifying the growth equation to the following form (known as discrete-time logistic growth):

$$n_{i+1} = 4n_i \left(1 - \frac{n_i}{K}\right). \tag{1}$$

When n_i is near K, the resources are so scarce you can actually have population decrease between generations, $n_{i+1} < n_i$. For simplicity, let us focus on the variable $x_i = n_i/K$, which measures how much of the carrying capacity has been filled at the ith time step. The logistic growth equation in terms of this variable can be found by dividing Eq. (1) by K:

$$x_{i+1} = 4x_i \left(1 - x_i\right). \tag{2}$$

This equation has a fascinating diversity of dynamics depending on the value of the prefactor, which in our case is 4. (For more details, see: https://en.wikipedia.org/wiki/Logistic_map.) For our choice of prefactor, this equation represents one of the simplest examples of chaos, as we will verify below. Note that by construction the magnitude of x_i is always between 0 and 1 at every generation.

a) Write a progam that allows you to iterate Eq. (2) starting from any initial value x_0, in order to generate a trajectory $(x_0, x_1, x_2, \ldots)$. Compare two trajectories, for example one starting at $x_0 = 0.1$, and the other starting at $x_0' = 0.1 + 10^{-8}$. Plot the absolute differences $\delta_i = |x_i - x_i'|$ between the trajectories over several hundred time steps (a log scale will help with the visualization). Check that δ_i initially increases exponentially (roughly linearly on a log graph), and then eventually saturates as you approach about 50 steps. This exponentially growing divergence between trajectories that start with very similar initial conditions is the hallmark of chaos, the so-called butterfly effect. (The trajectories cannot continue diverging forever, because both x_i and x_i' have to be between 0 and 1, and hence their absolute difference cannot grow indefinitely.) Thus if we have to choose a randomization time scale, setting $\delta t = 50$ time steps should ensure that the system is maximally randomized, and has had sufficient time to “forget” about its past history.

b) Now let us define coarse-grained “macrostates” for the system, which we record every $\delta t = 50$ time steps. The macrostates are $y_i = s(x_{i\delta t})$, where

$$s(x) = \begin{cases} 1 & \text{if } 0 \leq x < 1/4 \\ 2 & \text{if } 1/4 \leq x < 1/2 \\ 3 & \text{if } 1/2 \leq x < 3/4 \\ 4 & \text{if } 3/4 \leq x \leq 1 \end{cases}. \tag{3}$$

This essentially sorts x into four equally sized bins, labeled 1 through 4. So for example if we ran a trajectory of 200 time steps starting from $x_0 = 0.1$, and recorded every 50 time steps as

$$(x_0, x_{50}, x_{100}, x_{150}, x_{200}) = (0.100, 0.560, 0.372, 0.787, 0.0874), \tag{4}$$

the corresponding trajectory of states would be

$$(y_0, y_1, y_2, y_3, y_4) = (1, 3, 2, 4, 1). \tag{5}$$

We would like to verify that the state dynamics are Markovian. To do this, let us create an ensemble of trajectories in the following way: every trajectory will start in state 1, so we choose x_0 randomly between 0 and 1/4. Then iterate Eq. (2) over two hundred steps, and record every $\delta t = 50$ steps to get a sequence $(y_0, y_1, y_2, y_3, y_4)$. Repeat this whole procedure a large number of times (for example a million), and store in an array for later processing. We will thus have an ensemble of a million trajectories all starting from state 1. To calculate any conditional probability, we just have to count entries in this array.

As an example, let us say we were interested in the conditional probability $\mathcal{P}(y_4 = j | y_3 = 1)$. This is the probability of observing state j at time $t_4 = 4\delta t$, given that the previous observation y_3 was state 1. In terms of our ensemble, this can be estimated as

$$\mathcal{P}(y_4 = j | y_3 = 1) \approx \frac{\#\ \text{of trajectories of form } (*, *, *, 1, j)}{\#\ \text{of trajectories of form } (*, *, *, 1, *)}. \tag{6}$$

Here $*$ represents any state number. Similarly if we were interested in $\mathcal{P}(y_4 = j | y_3 = 1, y_2 = k)$, the probability of observing $y_4 = j$ given that the previous two states were k and 1, this can be calculated as

$$\mathcal{P}(y_4 = j | y_3 = 1, y_2 = k) \approx \frac{\#\ \text{of trajectories of form } (*, *, k, 1, j)}{\#\ \text{of trajectories of form } (*, *, k, 1, *)}. \tag{7}$$

Verify that in this case the Markovian assumption is true:

$$\mathcal{P}(y_4 = j | y_3 = 1, y_2 = k) \approx \mathcal{P}(y_4 = j | y_3 = 1), \tag{8}$$

for any choice of j or k. In other words, to know the probability of observing some y_4, we only need to know y_3, and not any state before y_3. Of course here we only checked Markovianity for

$y_3 = 1$, but we could equally have shown it for any other value of y_3 (you do not need to do this). The main reason that Markovianity works is that $\delta t = 50$ is longer than the timescale for trajectories to diverge due to chaos. Hence the precise value of y_2 does not matter when we look at y_4: the "memory" of the system does not extend that far back.

c) The validity of the Markovian assumption depends on the proper choice of δt. Had we chosen δt smaller than the timescale of randomization, then Markovianess may not have worked. Repeat the calculation of part (b) but use a timescale $\delta t = 1$ step to record states (hence you only need to run trajectories up to x_4). Show that in this case

$$\mathcal{P}(y_4 = j | y_3 = 1, y_2 = 1) \neq \mathcal{P}(y_4 = j | y_3 = 1, y_2 = 4) \tag{9}$$

for some states j, and hence Eq. (8) is no longer true. Here the probability of observing a certain y_4 is influenced by not just the immediate previous observation y_3, but also depends on y_2. The "memory" extends at least to y_2, and hence the system is not Markovian at these short timescales. Thus in general we will need two ingredients for the Markovian assumption to work in our coarse-grained dynamical descriptions of systems: (i) we need some underlying chaotic dynamics to ensure randomization; (ii) we need to wait a sufficiently long time δt between observations to ensure the memory of the system extends no further than the previous step.

d) Return to the setup of part (b) where $\delta t = 50$ time steps. We will stick with this choice for the rest of the problem set. Using the ensemble obtained in (b), calculate the full 4×4 transition matrix $W(t_3)$, whose elements are defined as

$$W_{ij}(t_3) = \mathcal{P}(y_4 = i | y_3 = j) \approx \frac{\text{\# of trajectories of form } (*, *, *, j, i)}{\text{\# of trajectories of form } (*, *, *, j, *)}. \tag{10}$$

In fact, you can calculate this transition matrix for any of the time steps. For example at the previous time step the matrix elements are

$$W_{ij}(t_2) = \mathcal{P}(y_3 = i | y_2 = j) \approx \frac{\text{\# of trajectories of form } (*, *, j, i, *)}{\text{\# of trajectories of form } (*, *, j, *, *)}. \tag{11}$$

Verify that for this system the transition matrix remains the same at each time. In other words, check that:

$$W(t_1) \approx W(t_2) \approx W(t_3) \tag{12}$$

Note that for the very first transition matrix, $W(t_0)$, you only can get statistics for the first column, because in your ensemble you always started from state 1 at t_0. But if you look at the first column of $W(t_0)$, it is approximately the same as the first column of $W(t_n)$ for $n > 0$.

e) The state probability vector at each time $\boldsymbol{p}(t_n)$ has components $p_i(t_n) = \mathcal{P}(y_n = i)$. We know from class it should obey the discrete-time master equation

$$\boldsymbol{p}(t_{n+1}) = W(t_n)\boldsymbol{p}(t_n). \tag{13}$$

So the probability distribution of states at time t_4 can be found by iterating the above equation,

$$\boldsymbol{p}(t_4) = W(t_3)W(t_2)W(t_1)W(t_0)\boldsymbol{p}(t_0). \tag{14}$$

Using the results from part (d), calculate $\boldsymbol{p}(t_4)$ via the above equation. Note that since we always start in state 1 in our ensemble, $\boldsymbol{p}(t_0) = (1, 0, 0, 0)$. (Though you have not explicitly calculated the full matrix $W(t_0)$, it turns out you only need to know the first column to calculate the product $W(t_0)\boldsymbol{p}(t_0)$.)

To check that the master equation and your ensemble agree with each other, calculate $\boldsymbol{p}(t_4)$ directly from your ensemble. In other words,

$$p_i(t_4) = \mathcal{P}(y_4 = i) \approx \frac{\#\ \text{ of trajectories of form } (*, *, *, *, i)}{\#\ \text{ of trajectories of form } (*, *, *, *, *)}, \tag{15}$$

for $i = 1, \ldots, 4$. Check that this is equal to the result you got from Eq. (14).

f) Using a procedure analogous to Eq. (14), calculate the probability vectors at all earlier time steps: $\boldsymbol{p}(t_1)$, $\boldsymbol{p}(t_2)$, ... Check to see if the probabilities converge to constant values as time progresses: does the system reach a stationary state where the probabilties stop changing, $\boldsymbol{p}(t_{n+1}) \approx \boldsymbol{p}(t_n)$ for sufficiently large n?

Problem 2: Machine learning as Bayesian model fitting

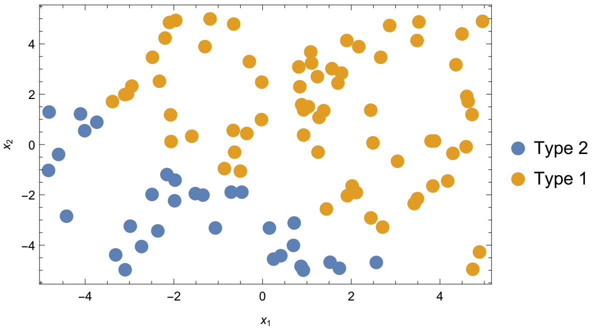

Figure 1: Training data.

In this problem we will see how fitting data to a model using machine learning can be framed in terms of Bayes's theorem. As described in class, we introduce a very simple classification problem: the input is a two-component vector $\boldsymbol{x} = (x_1, x_2)$, which has to be classified as either type 1 or type 2. We have $N = 100$ properly classified examples, shown in Fig. 1. These data points can be downloaded from the link: hinczlab.org/phys414/mldata.csv. The file has 100 rows, one per data point, with three columns separated by commas. The nth row has the format: $x_1^{(n)}$, $x_2^{(n)}$, ℓ_n. Here ℓ_n is the label (type) of the nth data point $\boldsymbol{x}^{(n)} = (x_1^{(n)}, x_2^{(n)})$. The goal is to design a network that can take a vector $\boldsymbol{x}$ it has never seen before, and tell us the probability that this vector belongs to either type 1 or type 2.

Because the classification problem is low-dimensional and straightforward, we do not need a complex network. For our purposes, the following will suffice. Consider a network whose input is $\boldsymbol{x}$ and output is $p_1(\boldsymbol{x}; \boldsymbol{w})$, the estimated probability that $\boldsymbol{x}$ belongs to type 1. This network depends on the parameters $\boldsymbol{w} = (w_1, w_2, w_3)$ in the followng way,

$$p_1(\boldsymbol{x}; \boldsymbol{w}) = \sigma(w_1 x_1 + w_2 x_2 + w_3), \tag{16}$$

where the nonlinear function $\sigma(y)$ is defined as

$$\sigma(y) = \frac{1}{2}(1 + \tanh y). \tag{17}$$

By construction $\sigma(y)$ can take any real y (positive or negative), and outputs a number between 0 and 1. Thus it is perfect for our case, where we want p_1 to correspond to a probability.

Let us now set up the Bayesian framework. The probability that we assign to a single labeled data point $(\boldsymbol{x}^{(n)}, \ell_n)$ is related to the network output as follows:

$$\mathcal{P}((\boldsymbol{x}^{(n)}, \ell_n)|\boldsymbol{w}) = \begin{cases} p_1(\boldsymbol{x}^{(n)}; \boldsymbol{w}) & \text{if } \ell_n = 1 \\ 1 - p_1(\boldsymbol{x}^{(n)}; \boldsymbol{w}) & \text{if } \ell_n = 2 \end{cases}. \tag{18}$$

Note that the second line reflects the fact that if we want to know the probability of something labeled as type 2, it is just one minus the type 1 probability. Since each data point is independent from the rest, the probability the network assigns to the whole data set $\mathcal{D}$ is just

$$\mathcal{P}(\mathcal{D}|\boldsymbol{w}) = \prod_{n=1}^{N} \mathcal{P}((\boldsymbol{x}^{(n)}, \ell_n)|\boldsymbol{w}). \tag{19}$$

Prior to any training, we have to make some assumptions about the network parameters, in order to keep them in reasonable ranges which prevent numerical errors. The most commonly used assumption is that the parameters are drawn from independent Gaussian distributions with zero mean and standard deviation s,

$$\mathcal{P}(\boldsymbol{w}) = \frac{1}{(2\pi s^2)^{3/2}} \exp\left(-\frac{\boldsymbol{w}^2}{2s^2}\right). \tag{20}$$

The parameter s is known as a regularization parameter, and controls the width of this prior distribution. Unless otherwise noted, we will set $s = 1$. As we will see below in part (d), if you make your prior too restrictive (s too small) you limit the range of networks the fitting can explore, and you will not be able to find a network that describes the data well.

The posterior distribution $\mathcal{P}(\boldsymbol{w}|\mathcal{D})$ is given by Bayes's theorem:

$$\mathcal{P}(\boldsymbol{w}|\mathcal{D}) = \frac{\mathcal{P}(\mathcal{D}|\boldsymbol{w})\mathcal{P}(\boldsymbol{w})}{\mathcal{P}(\mathcal{D})}. \tag{21}$$

The goal in this problem set will be to find a set of network parameters $\boldsymbol{w}$ that maximizes $\mathcal{P}(\boldsymbol{w}|\mathcal{D})$, and hence gives you a network that is most likely to describe the data. In a later problem set we will revisit this problem and figure out a way to estimate the whole posterior distribution $\mathcal{P}(\boldsymbol{w}|\mathcal{D})$. Knowing the full distribution is at the heart of *Bayesian machine learning*: it gives you a sense of how many alternative network parameter sets could have described the data, and hence a measure of how confident you can be in the network's prediction. For now, it is sufficient to find at least one network that solves our problem.

a) Write down (and simplify as much as possible) an expression for the loss function $\mathcal{L}(\boldsymbol{w}|\mathcal{D}) = -\log \mathcal{P}(\boldsymbol{w}|\mathcal{D})$. Minimizing the loss $\mathcal{L}$ will be equivalent to maximizing the posterior distribution. For simplicity, throw out any terms that do not directly depend on $\boldsymbol{w}$, for example $-\log \mathcal{P}(\mathcal{D})$. These will not be relevant to our minimization problem. Write a program to evaluate this loss function for our training data set $\mathcal{D}$. Set your regularization parameter to $s = 1$. Plug in a few different choices of the parameters $\boldsymbol{w}$ and see how the loss function behaves. See if you can tweak the parameter values to make the loss function smaller.

b) Trying to find the minimum of $\mathcal{L}$ by hand is not an efficient strategy, and would be impossible if the number of parameters were larger (in real-world applications the dimension of the vector $\boldsymbol{w}$ might be in the hundreds of millions). There are many methods to numerically minimize a function like $\mathcal{L}$, and here we will explore the simplest procedure, known as *gradient descent.* The idea is to imagine $\mathcal{L}(\boldsymbol{w}|\mathcal{D})$ as a landscape dependent on the three parameters $\boldsymbol{w} = (w_1, w_2, w_3)$. At any point $\boldsymbol{w}$ on this landscape, the gradient vector

$$\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}|\mathcal{D}) = (\partial_{w_1}\mathcal{L}(\boldsymbol{w}|\mathcal{D}), \partial_{w_2}\mathcal{L}(\boldsymbol{w}|\mathcal{D}), \partial_{w_3}\mathcal{L}(\boldsymbol{w}|\mathcal{D})) \tag{22}$$

tells us the direction in which the landscape is increasing most steeply. If we want to minimize $\mathcal{L}$, we should travel *opposite* to this direction, like gradually descending a mountain. Imagine we are making consecutive guesses as to where the minimum lies, $\boldsymbol{w}^{(m)}$, for $m = 0, 1, 2, \ldots$. Then the algorithm tells us the next guess should be related to the gradient at the current guess in the following manner:

$$\boldsymbol{w}^{(m+1)} = \boldsymbol{w}^{(m)} - \eta \nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}^{(m)}|\mathcal{D}). \tag{23}$$

The parameter η is known as the *learning rate*, and controls how quickly we descend the mountain. Typically η is kept small to prevent overshooting the minimum, but if we make η too small it will take forever to reach the minimum. For this problem $\eta = 0.01$ should work well. Write a program that implements gradient descent: starting from some initial guess, for example $\boldsymbol{w}^{(0)} = (1, 1, 1)$, the program should iterate Eq. (23) for several thousand steps, until you have converged on a parameter set $\boldsymbol{w}^{(m)}$ at large m that approximately minimizes the loss. Call this best guess parameter set $\boldsymbol{w}^*$. Check that this parameter set performs at least equal to (or hopefully better) than what you found by hand in part (a). *Hint:* before writing the program, write down an analytical expression for the gradient in Eq. (22). Be careful to note that the mathematical form of each term in the gradient corresponding to a labeled data point depends on the value of ℓ_n for that point.

c) To see how well your network performs, here are three new vectors that were not in your training data: $\boldsymbol{x}^A = (-1, -4)$, $\boldsymbol{x}^B = (-1, -1)$, and $\boldsymbol{x}^C = (-1, 3)$. These will be our *testing data*. Evaluate the network output $p_1(\boldsymbol{x}; \boldsymbol{w}^*)$ for each of these $\boldsymbol{x}$ using your best-guess parameter set $\boldsymbol{w}^*$ from part (b). If you have found a decent solution, the probability of being of type 1 should be close to zero for $\boldsymbol{x}^A$, somewhere in the middle for $\boldsymbol{x}^B$, and close to 1 for $\boldsymbol{x}^C$. If this is the case, congratulations, you have "taught" your machine well!

d) Of course the precise network parameters $\boldsymbol{w}^*$ you find from part (b) depend on your choice of prior distribution, controlled by the parameter s. Repeat parts (b) and (c) but using a much narrower prior distribution, with $s = 0.1$. Because you have constrained the range of network parameters to be much smaller (increasing the penalty for larger values of w_i) , the results should be worse: your loss function at $\boldsymbol{w}^*$ should be larger, and when you test the data in part (c) you should find inaccurate assignments of probabilities. Now repeat (b) and (c) with a much looser prior, $s = 10.0$. Here you should find decent results. The network parameters may be different from those found with $s = 1$, the algorithm still manages to find a good solution. Often the goal of machine learning is not necessarily to find a unique best solution, but something that "works" well for the task at hand. Parameters that influence the training, like s and η, are known as *hyperparameters* and have to be chosen carefully for the whole procedure to work.

Problem 3: Are we alone in the universe?

In this problem we will see how Bayesian analysis can help us estimate model parameters even in the extreme case of a single datapoint: life had to arise on Earth earlier than 3.5 Gyr (gigayears) ago (see Fig. 1 for the oldest fossilized evidence currently known). As of now we have no other datapoints of life existing anywhere in the universe (though according to a study published in January 2015 there are tantalizing indications that the Curiosity rover on Mars may be on the verge of adding another datapoint; see part (e) of this problem for an actual calculation of what this would imply). In general, can we say anything about the likelihood of life arising from non-living matter, a process known as *abiogenesis*? Life began early in the Earth's history: the Earth is 4.5 Gyr old, and life arose within the first 1 Gyr of its existence, though almost certainly not within the first 0.5 Gyr because conditions on the very early Earth were inhospitable. This fact seems to support the idea that abiogenesis is a typical occurrence in the universe, fueling optimism about life existing on many Earth-like exoplanets in habitable zones around Sun-like stars. The current estimate based on data from the Kepler spacecraft is that there could be roughly $\approx 10^{10}$ such planets in the Milky Way alone [Petigura *et al.*, Proc. Natl. Acad. Sci. **110**, 19273 (2013)]. If they are of comparable age to the Earth, what fraction of them harbor life? Is the optimism justified?

A more careful evaluation using Bayesian analysis was performed by David Spiegel and Edwin Turner [Proc. Natl. Acad. Sci. **109**, 395 (2012); posted on the course website]. We will derive (in simplified form) a version of their main results. The goal is to determine the conditional probability $\mathcal{P}(x|\mathcal{D})$. Here x are the parameters in a theoretical model for abiogenesis (in our case a single parameter). $\mathcal{D}$ is the data, which consists of humans having "measured" that life arose on earth by a time $t_{\text{emerge}} \approx 1$ Gyr after the planet's formation. The conditional probability $\mathcal{P}(x|\mathcal{D})$ encapsulates what we can say about x given the existing data. To evaluate it, we use Bayes's rule:

Figure 2: Datapoint #1: fossilized evidence of microbial communities dating back to 3.5 billion years ago, discovered in western Australia [Noffke *et al.*, Astrobiology **13**, 1103 (2013)].

$$\mathcal{P}(x|\mathcal{D}) = \frac{\mathcal{P}(\mathcal{D}|x)\mathcal{P}(x)}{\mathcal{P}(\mathcal{D})} \tag{24}$$

The denominator $\mathcal{P}(\mathcal{D})$ is a independent of x, so we can treat it as a normalization constant ensuring that $\int dx\, \mathcal{P}(x|\mathcal{D}) = 1$. To complete the analysis, we need expressions for $\mathcal{P}(\mathcal{D}|x)$ and $\mathcal{P}(x)$. The latter represents our prior knowledge (rough guess-work!) about x. Let us find each of these expressions in turn.

a) The first ingredient is a model for abiogenesis. We start with the assumption that conditions on a planet right after its formation will not allow life, up until some minimum time t_{min} has passed. If $t = 0$ is the time of planetary formation, we will fix $t_{\text{min}} \approx 0.5$ Gyr, assuming it is comparable for all Earth-like planets. Though abiogenesis is a complex series of chemical events, we can combine them all into a single overall "reaction", which happens at an unknown constant rate λ (a Poisson process) for all times $t \geq t_{\text{min}}$. More precisely, λ is the probability per unit time of abiogenesis, so that the probability of life arising in some short interval dt is λdt (or equivalently, $1 - \lambda dt$ is the probability that life did not arise in this interval). The probabilities in each consecutive interval (i.e. t to $t + dt$ and $t + dt$ to $t + 2dt$) are independent of each other. This model does not preclude life arising independently multiple times, but we are only interested in the first instance. Given the above assumptions, use the laws of probability (and the limit $dt \to 0$) to show that the probability that no life has arisen up to time t after a planet's formation is:

$$P_{\text{no-life}}(\lambda, t) = \begin{cases} 1 & 0 \leq t < t_{\text{min}} \\ e^{-\lambda(t - t_{\text{min}})} & t \geq t_{\text{min}} \end{cases} \tag{25}$$

Hence the probability that life has arisen (at least once) before time t is $P_{\text{life}}(\lambda, t) = 1 - P_{\text{no-life}}(\lambda, t)$. This will be our main model, governed by a single parameter λ which we would like to pinpoint. (As we will see in part (b), we will do this by estimating $x \equiv \log_{10} \lambda$, the overall order of magnitude.)

b) If you assume λ is set by fundamental chemistry and is the same throughout the universe, let us get a feel for the consequences of its scale. Find the different numerical values of λ (in units of Gyr^{-1}) that would imply the following facts are true for Earth-like planets of comparable age to ours ($t_0 = 4.5$ Gyr):

- λ_1: on average, we are the only such planet at the present time in the entire observable universe where life has emerged (out of $\approx 10^{20}$ Earth-like planets of similar age in the universe). In other words, $P_{\text{life}}(\lambda_1, t_0) = 10^{-20}$.
- λ_2: on average, we are the only such planet at the present time in the Milky Way where life has emerged (out of $\approx 10^{10}$ Earth-like planets of similar age in our galaxy). In other words, $P_{\text{life}}(\lambda_2, t_0) = 10^{-10}$.
- λ_3: 99.9999% of Earth-like planets of similar age have life. In other words, $P_{\text{life}}(\lambda_3, t_0) = 0.999999$.

From top to bottom, these give you a sense of the immense breadth of possible λ values. Since we do not even have a grasp of its order of magnitude, our prior probability distribution should reflect this. Let us define $x = \log_{10} \lambda$ and say that all orders of magnitude between $x_{\text{min}} = \log_{10} \lambda_1$ and $x_{\text{max}} = \log_{10} \lambda_3$ are equally probable. In terms of this parameter x we will choose our prior probability distribution to be:

$$\mathcal{P}(x) = \begin{cases} \frac{1}{x_{\text{max}} - x_{\text{min}}} & \text{if} \quad x_{\text{min}} \leq x \leq x_{\text{max}} \\ 0 & x < x_{\text{min}} \quad \text{or} \quad x > x_{\text{max}} \end{cases} \tag{26}$$

c) The implications of our single datapoint $\mathcal{D}$ are more complicated than just specifying an upper bound on Earth's abiogenesis. What $\mathcal{D}$ really states is that: "an intelligent life form on Earth was able to gather evidence at the present time ($t_0 = 4.5$ Gyr) showing that life started before a time $t_{\text{emerge}} = 1$ Gyr in the Earth's history." This presupposes that enough time has passed between t_{emerge} and the t_0 for evolution to produce a scientifically-advanced species capable of investigating fossil evidence of abiogenesis. If life on Earth emerged at $t = 4.0$ Gyr, there almost certainly would not be enough time for evolution to produce a species to collect the datapoint $\mathcal{D}$ at t_0. Let us specify a minimum time delay δt_{evolve} for the evolution of an intelligent species after abiogenesis. Then only abiogenesis events where $t_{\text{emerge}} < t_0 - \delta t_{\text{evolve}} \equiv t_{\text{required}}$ could have any possibility of being measured. Let us choose $\delta t_{\text{evolve}} = 1$ Gyr to set a rough time scale (probably on the short side) for the evolution of intelligence, so $t_{\text{required}} = 3.5$ Gyr is the cutoff for measurable abiogenesis required by evolutionary constraints. Let E be the statement "abiogenesis occurred between t_{min} and t_{emerge}", and R be the statement "abiogenesis occurred between t_{min} and t_{required}". Then we will take $\mathcal{P}(\mathcal{D}|x)$ to mean $\mathcal{P}(E|R, x)$, or the probability that E is true given that R is true and the model parameter value is x. Using the laws of probability and the result of part (a), argue that for any measured value of t_{emerge},

$$\mathcal{P}(\mathcal{D}|x) = \begin{cases} \frac{P_{\text{life}}(10^x, t_{\text{emerge}})}{P_{\text{life}}(10^x, t_{\text{required}})} & \text{if } t_{\text{min}} \leq t_{\text{emerge}} \leq t_{\text{required}} \\ 0 & \text{if } t_{\text{emerge}} < t_{\text{min}} \text{ or } t_{\text{emerge}} > t_{\text{required}} \end{cases} \tag{27}$$

Hint: Think about the definition of conditional probability. Also note that if $t_{\text{min}} \leq t_{\text{emerge}} \leq t_{\text{required}}$, then R is definitely true if E is true.

d) Putting the result of parts (b) and (c) together, use Bayes's rule to determine the posterior probability $\mathcal{P}(x|\mathcal{D})$. Make sure to normalize by choosing some appropriate numerical value for $\mathcal{P}(\mathcal{D})$. Plot $\mathcal{P}(x|\mathcal{D})$ versus x to see how the probability behaves. Using numerical integration, figure out the probability that x is between x_{min} and $x_{\text{mid}} = \log_{10} \lambda_2$. Let us call this probability p_{L}, where L represents extreme loneliness (we are surely alone in our galaxy, and possibly the observable universe). On the other extreme, figure out the probability p_{M} that 99% or more of Earth-like planets of comparable age to ours have seen life emerge. M represents "the more the merrier." How do you like these odds? *Hint:* you may find your numerical integrator (Mathematica!?!) gives nonsense when you try to extend the integration range down to x_{min}. To solve these numerical issues, use the following trick: if you need to integrate a function $f(x)$ from x_{min} to x_{max}, you can break the integral into two pieces: $\int_{x_{\text{min}}}^{x_{\text{max}}} dx\, f(x) = \int_{x_{\text{min}}}^{x_{\text{mid}}} dx\, f(x) + \int_{x_{\text{mid}}}^{x_{\text{max}}} dx\, f(x)$. The first piece you can evaluate analytically by using the $x \to -\infty$ limit of $f(x)$. It goes to a simple constant f_0 (which you can find by Taylor expansion in terms of small 10^x), so $\int_{x_{\text{min}}}^{x_{\text{mid}}} dx\, f(x) \approx f_0(x_{\text{mid}} - x_{\text{min}})$. The second piece you can evaluate numerically.

Figure 3: Datapoint #2 (hypothetical): the Gillespie lake outcrop on Mars exhibiting potential signs of microbial structures.

e) Nora Noffke, the geobiologist responsible for discovering the oldest fossils on Earth (Fig. 1) published an article recently analyzing photos taken by the Curiosity rover on Mars (Fig. 2; see the write-up at: http://shar.es/1bNqS7). She makes a case that Mars exhibits structures remarkably similar to fossilized microbial mats seen on Earth. If these speculations are proven to be true, we would have a second datapoint. What would be the consequences? The Gillespie lake outcrop on Mars where these photos were taken is 3.7 Gyr old, so $t_{\text{emerge}}^{\text{Mars}} = 0.8$ Gyr (Mars has the same age as Earth). Assuming t_{min} is unchanged for Mars, and that life arose there independently of Earth, how would $\mathcal{P}(\mathcal{D}|x)$ change with two datapoints? Recalculate p_{L} and p_{M} from part (d) (be careful to find the new normalization constant of the distribution first). That's a big pretty big difference, no? Stay tuned: searching for fossilized microbial mats is a major target for the upcoming Mars 2020 rover.

Note: a more complete Bayesian analysis would have allowed the other parameters like t_{min} and δt_{evolve} to vary, with appropriately chosen prior probabilities. This would be significantly more complex, beyond the scope of the problem set. If you are overly bothered by these limitations, feel free to do the analysis and write a research article!