

Advanced Machine Learning Subsidiary Notes

Lecture 18: Probabilistic Inference

Adam Prügel-Bennett

May 14, 2020

1 Keywords

- Hierarchical Models, Mixture of Gaussians, Expectation Maximisation

2 Main Points

2.1 Laws of Probability

- We quickly review probabilities (you should know all this)
- Probabilities and events
 - We typically associate probabilities with events
 - The probability of an event lie between 0 and 1
 - If the set of events \mathcal{E} are exhaustive and mutually exclusive then

$$\sum_{A \in \mathcal{E}} \mathbb{P}[A] = 1$$

- Often we associate numbers to events
 - These are known as *random variables*
 - Conventionally random variables are denoted by capitals, while we use lower-case letters to represent the value the random variable takes
 - We associate probability distributions to random variables
 - For discrete random variables these are known as *probability mass functions* and are denoted $\mathbb{P}[X = x]$
 - When our events are continuous we often associate the outcome to a continuous random variable
 - In this case the probability of a continuous random variable taking a particular value is typically 0
 - We then look at probability densities

$$f_X(x) = \lim_{\delta x \rightarrow 0} \frac{\mathbb{P}[x \leq X \leq x + \delta x]}{\delta x}$$

- * Densities are not probabilities (they can be greater than 1 although

$$\mathbb{P}[a \leq X \leq b] = \int_a^b f_X(x) \, dx$$

is a probability and is always less than or equal to 1)

- Probabilities become interesting when we have multiple events
 - The *joint probability* of both event A and B occurring is denoted $\mathbb{P}[A, B]$
 - The probability of random variables X and Y taking values x and y is denoted by
 - * $\mathbb{P}[X = x, Y = y]$ for discrete random variables or
 - * $f_{X,Y}(x, y)$ for continuous random variables (where this is now a probability density)
 - * sometimes we write $\mathbb{P}[X, Y]$ or $f(x, y)$ when the context is clear
 - The *conditional probability* of an event A happening given and event B has happened is denoted by $\mathbb{P}[A|B]$ for discrete random variables or $f_{X|Y}(x|y)$ for continuous random variables
 - * sometimes we write $\mathbb{P}[X|Y]$ or $f(x|y)$ when the context is clear
 - * conditional probability doesn't imply any causation
 - * Note that $\mathbb{P}[X|Y]$ or $f(x|y)$ are probabilities or densities of X

$$\sum_x \mathbb{P}[X = x|Y = y] = 1 \quad \int f(x|y) dx = 1$$

- * But they are not probabilities or densities of Y
- One of the most important rules in probabilities is
 - * $\mathbb{P}[X, Y] = \mathbb{P}[X|Y] \mathbb{P}[Y] = \mathbb{P}[Y|X] \mathbb{P}[X]$
 - * $f(x, y) = f(x|y) f(y) = f(y|x) f(x)$
 - * Clearly this is where Bayes' rule comes from
- A second rule that we use all the time is

$$\sum_y \mathbb{P}[X, Y = y] = \mathbb{P}[X] \quad \int f(x, y) dy = f(x)$$

- All this generalises to more the two random variables
 - * $\mathbb{P}[X = x, Y = y|Z = z]$ is the probability that both $X = x$ and $Y = y$ given that $Z = z$
 - * $\mathbb{P}[X = x|Y = y, Z = z]$ is the probability that $X = x$ given that $Y = y$ and $Z = z$
- Random variables are *independent* of each other if $\mathbb{P}[X, Y] = \mathbb{P}[X] \mathbb{P}[Y]$
- Random variables X and Y are conditionally independent of each other given Z if

$$\mathbb{P}[X, Y|Z] = \mathbb{P}[X|Z] \mathbb{P}[Y|Z]$$

- We often consider *random vectors* $\mathbf{X} = (X_1, X_2, \dots, X_n)$ where each component is a random variable
- I am expecting to you to know this material

2.2 Probabilistic Inference

- Most probabilistic inference involves constructing a model of the underlying data generation process and using Bayes' rule or maximum likelihood to learn unknown parameters of the model
- In modelling physical processes it is often easier to specify conditional probabilities where there is some causal relationship
- **Discriminative Models**
 - Often in machine learning our goal is to learn the probability distribution $\mathbb{P}[Y|\mathbf{X}]$ where Y is our target and \mathbf{X} is a data point

- We may parameterise this distribution with some parameters Θ and our task would be to learn these parameters based on training data

- **Generative Models**

- Surprisingly it is often easier to model the joint probability $\mathbb{P}[Y, \mathbf{X}]$
- This means that we model the process of both generating the targets and the feature vectors together
- These are known as *generative models* as they allow us to generate random examples
- We don't necessarily want to use them to generate random samples; it just makes the modelling process easier (although you need to get used to this as it feels counter-intuitive)
- We can use generative models to do discrimination since $\mathbb{P}[Y|\mathbf{X}] = \mathbb{P}[Y, \mathbf{X}] / \mathbb{P}[Y]$ where $\mathbb{P}[Y] = \sum_{\mathbf{X}} \mathbb{P}[Y, \mathbf{X}]$
- Examples of generative models include *Hidden Markov Models* and *Topic Models* (covered later)

- **Latent Variables**

- In building probabilistic models we often model quite complicated processes
- To do this we introduce intermediate processes described by random variables that we never observe
- These are known as **latent variable**
- Often our model will involve many different layers between the inputs \mathbf{X} and targets Y : this construction is sometimes known as a *hierarchical model*

- **Difficulty of Bayes**

- Bayesian inference is difficult because for most likelihoods there is no conjugate prior and the posterior is a mess
- In this case it can be very difficult to compute the *evidence* or *marginal likelihood*

$$\mathbb{P}[\mathcal{D}] = \sum_{\Theta} \mathbb{P}[\mathcal{D}|\Theta] \mathbb{P}[\Theta]$$

or

$$f(\mathcal{D}) = \int f(\mathcal{D}|\theta) f(\theta) d\theta$$

- * this is hard when Θ takes on too many values (e.g. it might be a high dimensional vector or a continuous variable)
- One solution to this is to obtain samples from the posterior distribution (this approach uses Monte Carlo methods which are very powerful, but can be slow)
- Another approach is to seek the **maximum a-posteriori** or **MAP** solution

$$\theta_{\text{MAP}} = \underset{\theta}{\operatorname{argmax}} f(\mathcal{D}|\theta) f(\theta) = \underset{\theta}{\operatorname{argmax}} \log(\mathcal{D}|\theta) + \log(f(\theta))$$

- * this is much easier than the full Bayesian approach as we don't need to compute the marginal likelihood $f(\mathcal{D})$
- * but it isn't really Bayesian (although some users will claim it is)
- * it throws away the posterior and replaces this with its mode
- * we have lost a measure of uncertainty
- We can go one step further and assume a uniform prior
 - * This leads to the **maximum likelihood estimate**
 - * This was first proposed by Ronald Fisher in the time when Bayesian inference was considered taboo
 - * Despite its strong connection to Bayesian inference it was accepted by the statistical community

2.3 Mixtures of Gaussians

- To illustrate latent variables and a simple hierarchical model we consider a classic probabilistic model known as *mixture of Gaussians*
- We consider a concrete scenario
- We suppose we are observing the decay of two types (A and B) of short-lived particles ¹
- We can measure their half lives, X_i , but we don't know the type of particle
- We have a measurement error of the half-life
- Let $Z_i \in \{0, 1\}$ equal 1 if particle i is of type A and 0 if it is of type B
- The probability distribution of the half-life measurement is therefore

$$f(X_i|Z_i, \Theta) = Z_i \mathcal{N}(X_i|\mu_A, \sigma_A^2) + (1 - Z_i) \mathcal{N}(X_i|\mu_B, \sigma_B^2)$$

- where μ_A and μ_B are the expected half-lives for particles of type A and B respectively
- σ_A and σ_B are the standard deviations in the measurements
- this says that if the i^{th} particle is of type A then the probability of X_i is $\mathcal{N}(X_i|\mu_A, \sigma_A^2)$, while if it of type B , then X_i is distributed according to $\mathcal{N}(X_i|\mu_B, \sigma_B^2)$
- We show some typical data from $m = 1\,000$ observations in Figure 1

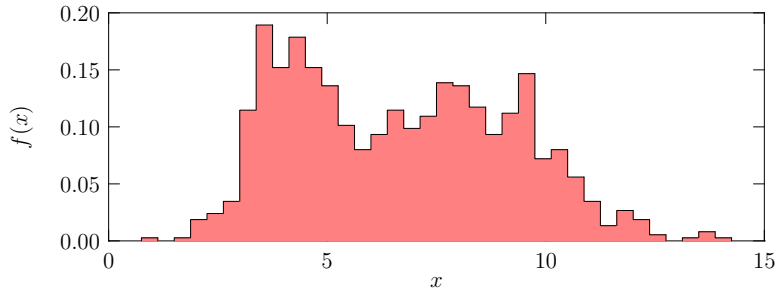


Figure 1: Example of data measuring the half-lives of two types of particles

- Our job is to infer the random variables $\Theta = (\mu_1, \mu_2, \sigma_1, \sigma_2, p)$, where $p = \mathbb{P}[Z_i = 1]$ is the probability of the particle being type A
- We can do a full Bayesian calculation, but let us just use maximum likelihood
- The maximum likelihood of the data $\mathcal{D} = \{X_i | i = 1, 2, \dots, m\}$ is

$$\begin{aligned} f(\mathcal{D}|\Theta) &\stackrel{(1)}{=} \sum_{\mathbf{Z} \in \{0,1\}^m} f(\mathcal{D}, \mathbf{Z}|\Theta) \\ &\stackrel{(2)}{=} \prod_{i=1}^m \sum_{Z_i \in \{0,1\}} f(X_i, Z_i|\Theta) \stackrel{(3)}{=} \prod_{i=1}^m \sum_{Z_i \in \{0,1\}} f(X_i|Z_i, \Theta) \mathbb{P}[Z_i] \end{aligned}$$

(1) where we marginalise out the latent variables $\mathbf{Z} = (Z_1, Z_2, \dots, Z_n)$

(2) we assume the data is independent

(3) we use the identity $f(X_i, Z_i|\Theta) = f(X_i|Z_i, \Theta) \mathbb{P}[Z_i]$

¹If you prefer, you can think of an autonomous vehicle using lidar where it detects reflections from two different, but close by, objects, A and B . We make multiple noisy measurements of the distance from the two objects.

- It is usually easier working with the log-likelihood

$$\begin{aligned}\log(f(\mathcal{D}|\Theta)) &= \sum_{i=1}^m \log(f(X_i|Z_i=1) \mathbb{P}[Z_i=1] + f(X_i|Z_i=0) \mathbb{P}[Z_i=0]) \\ &= \sum_{i=1}^m \log(p \mathcal{N}(X_i|\mu_A, \sigma_A) + (1-p) \mathcal{N}(X_i|\mu_B, \sigma_B))\end{aligned}$$

- We could do gradient descent on this, but it is an ugly expression to work with

2.4 Expectation Maximisation

- Rather than maximise the likelihood directly we can iteratively maximise the expected log-likelihood starting from some initial guess $\Theta^{(0)}$; we get an improved guess

$$\Theta^{(t+1)} = \underset{\Theta}{\operatorname{argmax}} \sum_{\mathbf{Z} \in \{0,1\}^m} \mathbb{P}[\mathbf{Z}|\mathcal{D}, \Theta^{(t)}] \log(f(\mathcal{D}, \mathbf{Z}|\Theta)) \quad (1)$$

- This is a general optimisation strategy that is regularly used when we have latent variables
- It is known as **expectation maximisation** or the **EM-algorithm**
- This looks very different to maximising the log-likelihood: it takes some effort to understand why this works
- To understand this we note

$$f(\mathcal{D}, \mathbf{Z}|\Theta) = f(\mathcal{D}|\mathbf{Z}, \Theta) \mathbb{P}[\mathbf{Z}|\Theta]$$

From which we can deduce (taking logs and rearranging)

$$\log(f(\mathcal{D}|\mathbf{Z}, \Theta)) = \log(f(\mathcal{D}, \mathbf{Z}|\Theta)) - \log(\mathbb{P}[\mathbf{Z}|\Theta])$$

- We now consider the probability distribution $\mathbb{P}[\mathbf{Z}|\mathcal{D}, \Theta^{(t)}]$, that tells us the probability that $Z_i = 1$ given X_i and the parameters $\Theta^{(t)}$ (this is different to the prior distribution $\mathbb{P}[\mathbf{Z}|\Theta^t] = p^{(t)}$)
- If we not take expectations of $\log(f(\mathcal{D}|\Theta))$ give above with respect to this distribution then

$$\begin{aligned}\log(f(\mathcal{D}|\Theta)) &= \mathbb{E}_{\mathbf{Z}|\Theta^{(t)}}[\log(f(\mathcal{D}, \mathbf{Z}|\Theta))] - \mathbb{E}_{\mathbf{Z}|\Theta^{(t)}}[\log(\mathbb{P}[\mathbf{Z}|\Theta])] \\ &= Q(\Theta|\Theta^{(t)}) + S(\Theta|\Theta^{(t)})\end{aligned}$$

- Note that the left-hand side does not involve the latent variables so when we take the expectation we get itself
- The first term on the right-hand side is

$$Q(\Theta|\Theta^{(t)}) = \mathbb{E}_{\mathbf{Z}|\Theta^{(t)}}[\log(f(\mathcal{D}, \mathbf{Z}|\Theta))] = \sum_{\mathbf{Z} \in \{0,1\}^m} \mathbb{P}[\mathbf{Z}|\mathcal{D}, \Theta^{(t)}] \log(f(\mathcal{D}|\mathbf{Z}, \Theta))$$

- This is the term we are optimising in equation (1)
- The second term is

$$S(\Theta|\Theta^{(t)}) = -\mathbb{E}_{\mathbf{Z}|\Theta^{(t)}}[\log(\mathbb{P}[\mathbf{Z}|\Theta])] = - \sum_{\mathbf{Z} \in \{0,1\}^m} \mathbb{P}[\mathbf{Z}|\mathcal{D}, \Theta^{(t)}] \log(\mathbb{P}[\mathbf{Z}|\Theta])$$

- Using the identity for the log-likelihood we can write the change in log-likelihood when we update our parameters

$$\begin{aligned}\Delta L &= \log(f(\mathcal{D}|\Theta^{(t+1)})) - \log(f(\mathcal{D}|\Theta^{(t)})) \\ &= Q(\Theta^{(t+1)}|\Theta^{(t)}) - Q(\Theta^{(t)}|\Theta^{(t)}) + S(\Theta^{(t+1)}|\Theta^{(t)}) - S(\Theta^{(t)}|\Theta^{(t)}) \\ &= Q(\Theta^{(t+1)}|\Theta^{(t)}) - Q(\Theta^{(t)}|\Theta^{(t)}) + \text{KL}\left(\mathbb{P}[Z|\Theta^{(t)}] \parallel \mathbb{P}[Z|\Theta^{(t+1)}]\right)\end{aligned}$$

– where

$$\begin{aligned}\text{KL}\left(\mathbb{P}[Z|\Theta^{(t)}] \parallel \mathbb{P}[Z|\Theta^{(t+1)}]\right) &= S(\Theta^{(t+1)}|\Theta^{(t)}) - S(\Theta^{(t)}|\Theta^{(t)}) \\ &= - \sum_{Z \in \{0,1\}^m} \mathbb{P}[Z|\mathcal{D}, \Theta^{(t)}] \log\left(\frac{\mathbb{P}[Z|\Theta^{(t+1)}]}{\mathbb{P}[Z|\Theta^{(t)}]}\right)\end{aligned}$$

– We have shown in a previous lecture that KL-divergences are non-negative

- Now in expectation maximisation we choose

$$\Theta^{(t+1)} = \underset{\Theta}{\operatorname{argmax}} Q(\Theta|\Theta^{(t)})$$

which implies $Q(\Theta^{(t+1)}|\Theta^{(t)}) \geq Q(\Theta^{(t)}|\Theta^{(t)})$

- Thus $\Delta L \geq 0$, so in each step we increase the log-likelihood
- This gives us a relative simple procedure for maximising the likelihood (we can also use this to maximise the *a posteriori* solution); we choose Θ to maximise

$$Q(\Theta|\Theta^{(t)}) = \sum_{Z \in \{0,1\}^m} \mathbb{P}[Z|\mathcal{D}, \Theta^{(t)}] \log(f(\mathcal{D}|Z, \Theta))$$

- Let us return to the problem of working out the half-life statistics of our two types of particles A and B

- Recall $f(\mathcal{D}, Z|\Theta) = \prod_{i=1}^m f(X_i|Z_i, \Theta) \mathbb{P}[Z_i]$ where

$$f(X_i, Z_i|\Theta) = p Z_i \mathcal{N}(X_i|\mu_A, \sigma_A^2) + (1-p)(1-Z_i) \mathcal{N}(X_i|\mu_B, \sigma_B^2)$$

- Let

$$\begin{aligned}p_i^{(t)} &= \mathbb{P}[Z_i = 1|X_i, \Theta^{(t)}] = \frac{p^{(t)} \mathcal{N}(X_i|\mu_A^{(t)}, \sigma_A^{2(t)})}{p^{(t)} \mathcal{N}(X_i|\mu_A^{(t)}, \sigma_A^{2(t)}) + (1-p^{(t)}) \mathcal{N}(X_i|\mu_B^{(t)}, \sigma_B^{2(t)})} \\ q_i^{(t)} &= \mathbb{P}[Z_i = 0|X_i, \Theta^{(t)}] = \frac{(1-p^{(t)}) \mathcal{N}(X_i|\mu_B^{(t)}, \sigma_B^{2(t)})}{p^{(t)} \mathcal{N}(X_i|\mu_A^{(t)}, \sigma_A^{2(t)}) + (1-p^{(t)}) \mathcal{N}(X_i|\mu_B^{(t)}, \sigma_B^{2(t)})} = 1 - p_i^{(t)}\end{aligned}$$

- Then

$$\begin{aligned}Q(\Theta|\Theta^{(t)}) &= \sum_{i=1}^m p_i^{(t)} \log(p^{(t)} \mathcal{N}(X_i|\mu_A, \sigma_A^2)) + q_i^{(t)} \log((1-p^{(t)}) \mathcal{N}(X_i|\mu_B, \sigma_B^2)) \\ &= \sum_{i=1}^m p_i^{(t)} \left(\log(p) - \frac{(X_i - \mu_A)^2}{2\sigma_A^2} - \frac{1}{2} \log(2\pi\sigma_A^2) \right) \\ &\quad + q_i^{(t)} \left(\log(1-p) - \frac{(X_i - \mu_B)^2}{2\sigma_B^2} - \frac{1}{2} \log(2\pi\sigma_B^2) \right)\end{aligned}$$

- To optimise this we just set the derivatives to 0

- Optimising with respect to p

$$\frac{\partial Q(\Theta|\Theta^{(t)})}{\partial p} = \frac{1}{p} \sum_{i=1}^m p_i^{(t)} - \frac{1}{1-p} \sum_{i=1}^m q_i^{(t)} = 0$$

solving for p

$$p^{(t+1)} = \frac{\sum_{i=1}^m p_i^{(t)}}{\sum_{i=1}^m (p_i^{(t)} + q_i^{(t)})} = \frac{1}{m} \sum_{i=1}^m p_i^{(t)}$$

- Optimising with respect to μ_A

$$\frac{\partial Q(\Theta|\Theta^{(t)})}{\partial \mu_A} = - \sum_{i=1}^m p_i^{(t)} \frac{X_i - \mu_A}{\sigma_A^2}$$

solving for μ_A (and performing a similar optimisation for μ_B)

$$\mu_A^{(t+1)} = \frac{\sum_{i=1}^m p_i^{(t)} X_i}{\sum_{i=1}^m p_i^{(t)}}, \quad \mu_B^{(t+1)} = \frac{\sum_{i=1}^m q_i^{(t)} X_i}{\sum_{i=1}^m q_i^{(t)}}$$

- Putting in the optimal value for $\mu_A^{(t)}$ and optimising with respect to σ_A^2

$$\frac{\partial Q(\Theta|\Theta^{(t)})}{\partial \sigma_A^2} = \frac{1}{2\sigma_A^4} \sum_{i=1}^m p_i^{(t)} (X_i - \mu_A^{(t)})^2 - \frac{1}{\sigma_A^2} \sum_{i=1}^m p_i^{(t)}$$

Solving for σ_A^2 (and performing a similar optimisation for σ_B^2)

$$\sigma_A^2 = \frac{\sum_{i=1}^m p_i^{(t)} (X_i - \mu_A^{(t)})^2}{\sum_{i=1}^m p_i^{(t)}}, \quad \sigma_B^2 = \frac{\sum_{i=1}^m q_i^{(t)} (X_i - \mu_B^{(t)})^2}{\sum_{i=1}^m q_i^{(t)}}$$

- These are very natural update equations
 - we make an estimate, $p_i^{(t)}$ of the probability that observation X_i is a particle of type A or B base on our current parameters
 - we then update all our parameters based on these estimates
- We are guaranteed that our EM-algorithm never decreases the likelihood (although it could reach a local rather than global optimum)
- For the data set we showed earlier (which was a random sample of size 1000 generated using $p = 0.3$, $\mu_A = 4$, $\sigma_A = 0.8$, $\mu_B = 8$ and $\sigma_B = 2$) we get the results shown in Figure 2
- The EM algorithm often leads to very natural update equations, but its convergence is often rather slow

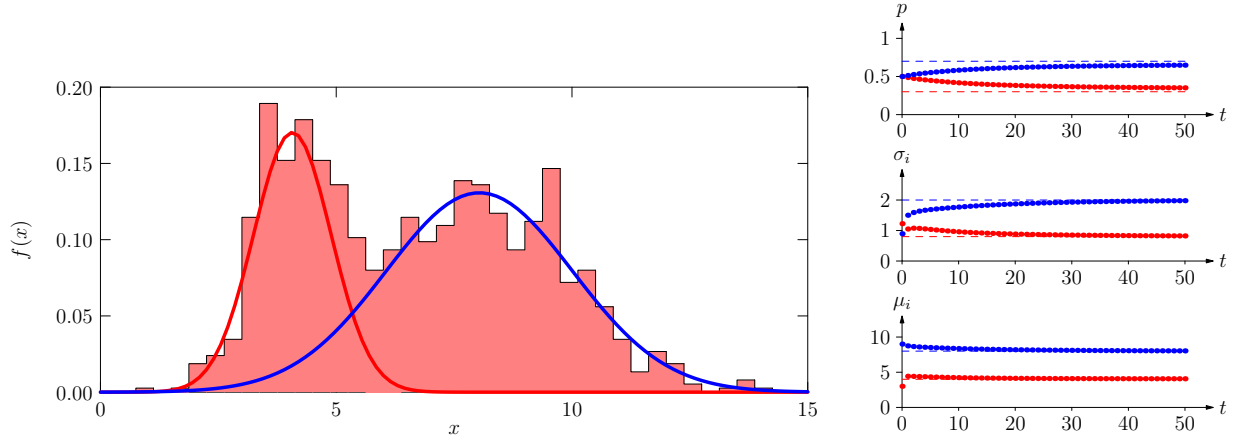


Figure 2: Example of EM algorithm to compute the statistics for the half-lives of our two particles

3 Exercises

3.1 Mysterious Disease

- We assume that we are tracking some disease
- Let $Z(t)$ be the number of people that catch the disease on day t , but this is unknown (a latent variable)
- We assume the rate of growth of the disease is

$$\mathbb{P}[Z(t+1)] = \text{Poi}\left(Z(t+1) \middle| \frac{r_0}{3} (Z(t) + Z(t-1) + Z(t-2))\right)$$

- We are assuming that someone is virulent for the first three days after catching the disease
- We assume $Z(1) = 1$ and $Z(0) = Z(-1) = 0$
- In expectation everyone with the disease will infect r_0 new people
- We observe $X(t)$ which is some proportion of new patients such that

$$\mathbb{P}[X(t) = k] = \text{Binom}(k|Z(t), p) = \binom{Z(t)}{k} p^k (1-p)^{Z(t)-k}$$

- We assume that each patient will be tested with a probability p
- We are given a time series $(X(1), X(2), \dots, X(T))$
- Build a probabilistic model to estimate p and k_0
- See answers

3.2 Experiments

3.3 Mysterious Disease

- Build a simulator of your models (assume you know p and r_0)
- Choose any language you are comfortable with
- If you are feeling very adventurous you could try to solve your model to predict p and r_0 , but be warned this is hard (you probably need to use MCMC, but you could try an EM algorithm)


```

r0 = 2
r = r0/3
p = 0.2
T = 20
Z(1) = 1;
Z(2) = poissrnd(r*Z(1));
Z(3) = poissrnd(r*(Z(1)+Z(2)));
for t = 4:T
    Z(t) = poissrnd(r*(Z(t-1)+Z(t-2)+Z(t-3)));
endfor

for t= 1:T
    X(t) = binornd(Z(t),p);
endfor

```

4 Answers

4.1 Mysterious Disease

- We want to compute $f(p, k_0 | \mathcal{D})$ where $\mathcal{D} = (X(1), X(2), \dots, X(T))$
- We have a likelihood of

$$\mathbb{P}[X(t) | Z(t), p] = \text{Binom}(X(t) | Z(t), p)$$

where

$$\mathbb{P}[Z(t+1) | r_0, Z(t), Z(t-1), Z(t-2)] = \text{Poi}\left(Z(t+1) \middle| \frac{r_0}{3} (Z(t) + Z(t-1) + Z(t-2))\right)$$

- To perform a Bayesian calculation we would have to put priors, $f(p)$ and $f(r_0)$, on p and r_0
 - A reasonable prior to use for p is $f(p) = \text{Beta}(p | a, b)$ —you could use $a = b = 0$ as an uninformative priors, but you might have some prior knowledge, e.g. $a = b = 2$ say which says $f(p) = 6p(1-p)$
 - A reasonable prior for r_0 would be $\text{Gamma}(r_0 | a, b)$ —you could use $a = b = 0$ as an uninformative prior but again you might have some prior belief, e.g. $a = 2, b = 1$
- Bayes rule tells us

$$f(p, r, \mathbf{Z} | \mathbf{X}) = \frac{\mathbb{P}[X(t) | Z(t), p] \prod_{t=1}^T \mathbb{P}[Z(t) | Z(t-1), Z(t-2), Z(t-3), r_0] f(r_0) f(p)}{\mathbb{P}[\mathbf{X}]}$$

- To get estimates of p and r_0 we have to marginalise out \mathbf{Z}
- Now the problem is this is rather horrible to compute (your priors are not conjugate priors in this problem and the posterior is very complicated)
- Probably the best way to do this is to use Markov Chain Monte Carlo (MCMC), but you will have to wait before I get there