Assignment 3

Callum Lau, 19102521

Unsupervised Learning

November 25, 2019

Question 1

Part (a)

We use the following algorithm to determine if a path is blocked by a node V. If all such paths are blocked then we can say $X \perp\!\!\!\perp Y \mid Z$. We say that a path is blocked if:

(i) $V \in Z$ and the arrows at V meet tail to tail or head to tail at the node.

(ii) $V \notin Z$ and none of its descendants are in Z and the arrows meet head to head at the node (i.e. the node is a collider).

We now consider graphs 2, 4, 6, 8 using this scheme. Note that the node C is connected to B and D in all graphs and therefore we only have to consider the conditional independency between C and A:

Graph 2 $C \perp\!\!\!\perp A \mid B$ since on the only path $A \to B \to C$, the node B is in the conditioning set and the arrows meet tail to tail. Therefore by (i) B blocks this path and hence the result. It follows easily we also have $C \perp\!\!\!\perp A \mid \{B, D\}$.

Graph 4 $C \perp\!\!\!\perp A \mid B$ since on the path $A \to B \to C$ the arrows meet head to tail and B is in the conditioning set. Similarly on the path $A \to B \to D \to C$, B is in the conditioning set and the arrows meet head to tail again. Therefore node B blocks all paths and hence the result. It follows easily we also have $C \perp\!\!\!\perp A \mid \{B, D\}$

Graph 6 $C \perp\!\!\!\perp A \mid B$ can be read off immediately from the graph since the graph is undirected and every path from A to C contains B. It follows easily we also have $C \perp\!\!\!\perp A \mid \{B, D\}$

Graph 8 $C \perp\!\!\!\perp A \mid B$ since every path from A to C again contains B. It follows easily we also have $C \perp\!\!\!\perp A \mid \{B, D\}$

Part (b) Denote C_I the collection of graphs $I \subset \{1, 2, 3, 4, 5, 6, 7, 8\}$ such that they all exhibit the same marginal and conditional independencies.

1. We have C_{12} is valid since the arrows $A \to B$ and $A \leftarrow B$ do not affect the conditional independence relationship $C \perp\!\!\!\perp A \mid B$, and all other arrows remain the same, meaning we still have $B \perp\!\!\!\perp D \mid \varnothing$ and $B \not\perp\!\!\!\perp D \mid C$

2. We also have C_{35}, since in 3 there are no colliders, and the arrows all point in the same direction which implies all nodes are conditionally independent of all others given their neighbour. This set of conditionals can be read of directly from 5. (And there are no marginal independencies).

3. Also note that $6, 7$ and 8 are all equivalent since 7 and 8 are simply factor graphs that represent the same conditional independencies displayed in the undirected graph 4. Finally we note that 4 is equivalent to 6. In graph 4 there are no marginal independencies - the parent nodes of B (C and D) are connected - and the remaining conditional independencies are $C \perp\!\!\!\perp A \mid B$ and $D \perp\!\!\!\perp A \mid B$, which are in 6 (and there are no more). Therefore we have C_{4678}.

We know that adding edges can only reduce the number of conditional independencies contained in the graph, and therefore removing edges reduces the class of dependencies the graph can represent. We also consider just one member of each collection compared to another, since if we show one graph is subsumed by another, all members of that collection must also be subsumed.

1. Comparing 5 and 6, we see that 5 contains all the conditional independencies of 6 since it does not have the edge between B and D (and so it also has $B \perp\!\!\!\perp D \mid C$). Therefore C_{35} is subsumed by C_{4678}.

2. Comparing 1 and 8 we see that they express all the same conditional independencies apart from in 1 we have that $B \perp\!\!\!\perp D \mid \varnothing$. Hence C_{12} is subsumed by C_{4678}.

3. Finally note that neither C_{12} and C_{35} are subsumed by eachother since C_{12} expresses $B \perp\!\!\!\perp D \mid \varnothing$ (which is not in C_{35}) and C_{35} expresses $B \perp\!\!\!\perp D \mid C$ (which is not in C_{12}).

Question 2

Let $X = \{M, L, A, V, H, P, B\}$ and $X_j^i = i$ for $i \in \{0, 1\}$ and $j \in X$. We quantify the background knowledge as follows:

Part (a) We can write the joint distribution specified by the DAG as

$$p(X) = p(P|V)p(M|V)p(L|P)p(H|M, L)p(B|M, A)p(A)p(V)$$

We quantify the background knowledge using the heuristic 'rare' ≈ 0.01, 'very rare' 0.001, 'sometimes' ≈ 0.2, 'usually' ≈ 0.9

$$p(M^1) = 0.01 \tag{1}$$
$$p(L^1) = 0.001 \tag{2}$$
$$p(A^1) = 0.001 \tag{3}$$
$$p(P^1|V^1) = 0.8 \tag{4}$$

$$p(P^1|V^0) = 0.2 \tag{5}$$
$$p(B^1|M^1) = 0.9 \tag{6}$$
$$p(H^1|M^1) = 0.9 \tag{7}$$
$$p(H^1|L^1) = 1 \tag{8}$$
$$p(B^1|A^1) = 0.2 \tag{9}$$
$$p(V^1) = 0.2 \tag{10}$$

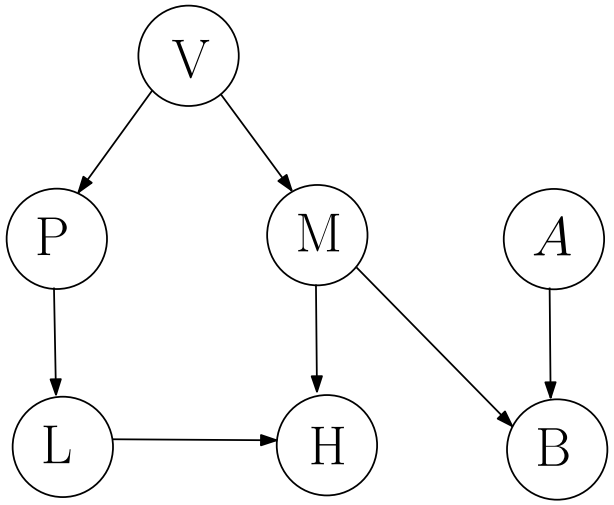


Figure 1: DAG

Part (b) In this part we assume that the expression "more likely" and "higher probability" corresponds to twice as likely. Using statements (1) to (9) and the total law of probability we infer the following conditional probability tables:

$p(P\|V)$	P^0	P^1
V^0	0.8	0.2
V^1	0.2	0.8

This table follows directly from expressions (4) and (5)
Using expression (10) and the fact that "Vulcans have a higher probability of getting Microsoftus than humans" and the total law of probability:

$$\begin{aligned} p(M^1) &= p(M^1|V^0)p(V^0) + p(M^1|V^1)p(V^1) \\ 0.01 &= c \times 0.8 + 2c \times 0.2 \\ \implies P(M^1|V^1) = c = \frac{1}{120} &\implies P(M^1|V^0) = \frac{1}{60} \end{aligned}$$

Hence we have the table

$p(M\|V)$	M^0	M^1
V^0	$\frac{119}{120} \approx 0.9917$	$\frac{1}{120} \approx 0.0083$
V^1	$\frac{59}{60} \approx 0.983$	$\frac{1}{60} \approx 0.0167$

Now consider:

$$p(P^1) = p(P^1|V^0)p(V^0) + p(P^1|V^1)p(V^1)$$
$$p(P^1) = 0.2 \times 0.8 + 0.8 \times 0.2 = 0.32$$

Then using (4) and (10), and the expression that the probability of having Linuxitis is twice as likely given that you eat pizza:

$$p(L^1) = p(L^1|P^0)p(P^0) + p(L^1|P^1)p(P^1)$$
$$0.001 = c \times 0.68 + 2c \times 0.32$$
$$\implies p(L^1|P^0) = c = \frac{1}{1320} \implies p(L^1|P^1) = \frac{1}{660}$$

Then we have the table:

$p(L\|P)$	L^0	L^1
P^0	$\frac{1319}{1320} \approx 0.99925$	$\frac{1}{1320} \approx 0.00075$
P^1	$\frac{659}{660} \approx 0.9985$	$\frac{1}{660} \approx 0.0015$

Next, using expressions (2), (7) and (8) we consider:

$$p(H^1|M^1) = p(H^1|M^1, L^0)p(L^0) + p(H^1|M^1, L^1)p(L^1)$$
$$0.9 = p(H^1|M^1, L^0) \times 0.999 + 1 \times 0.001$$
$$\implies p(H^1|M^1, L^0) = \frac{899}{999}$$

And give a somewhat arbitrary probability of having a high temperature given L^0 and M^0 of $P(H^1|M^0, L^0) = 0.001$ (possibly other illnesses). Hence we get the probability table:

$p(H\|M, L)$	H^0	H^1
M^0, L^0	$\frac{999}{1000} = 0.999$	$\frac{1}{1000} = 0.001$
M^0, L^1	0	1
M^1, L^0	$\frac{100}{999} \approx 0.1002$	$\frac{899}{999} \approx 0.8998$
M^1, L^1	0	1

Finally note that we have $p(B^1|M^1) = 0.9$ and $p(B^1|A^1) = 0.2$. Hence we guess that $P(B^1|M^1, A^1) = 0.95$, i.e. having both diseases will increase the risk of getting blue spots.

We also guess that the probability of having blue spots given that the patient does not have Microsoftus or Applosis is very low, i.e. $p(B^1|M^0, A^0) = 0.001$ Then we can find using (1), (3), (6) and (9):

$$\begin{aligned} p(B^1|M^1) &= p(B^1|M^1, A^1)p(A^1) + p(B^1|M^1, A^0)P(A^0) \\ 0.9 &= 0.95 \times 0.001 + p(B^1|M^1, A^0) \times 0.999 \\ \implies p(B^1|M^1, A^0) &= \frac{89905}{99900} \\ p(B^1|A^1) &= p(B^1|M^1, A^1)p(M^1) + p(B^1|M^0, A^1)P(M^0) \\ 0.2 &= 0.95 \times 0.01 + p(B^1|M^0, A^1) \times 0.99 \implies p(B^1|M^0, A^1) = \frac{127}{660} \end{aligned}$$

$p(B\|M, A)$	B^0	B^1
M^0, A^0	$\frac{999}{1000} = 0.999$	$\frac{1}{1000} = 0.001$
M^0, A^1	$\frac{533}{660} \approx 0.80758$	$\frac{127}{660} \approx 0.19242$
M^1, A^0	$\frac{89905}{99900} \approx 0.10006$	$\frac{89905}{99900} \approx 0.89994$
M^1, A^1	0.05	0.95

Part (c)

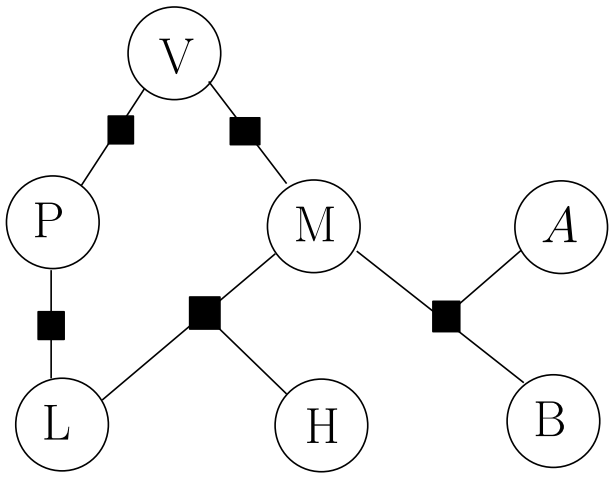


Figure 2: Factor Graph

We produce an undirected graph via moralisation of the parents of H and B.

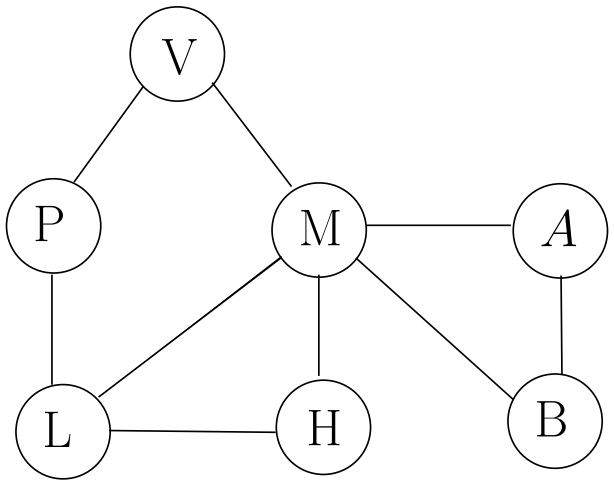


Figure 3: Undirected Graph

We now make the $LPVM$ sector into a clique via variable elimination and using minimum deficiency search. Note that removing both P and V induce one extra edge on the graph. Therefore we make the arbitrary choice of eliminating V.

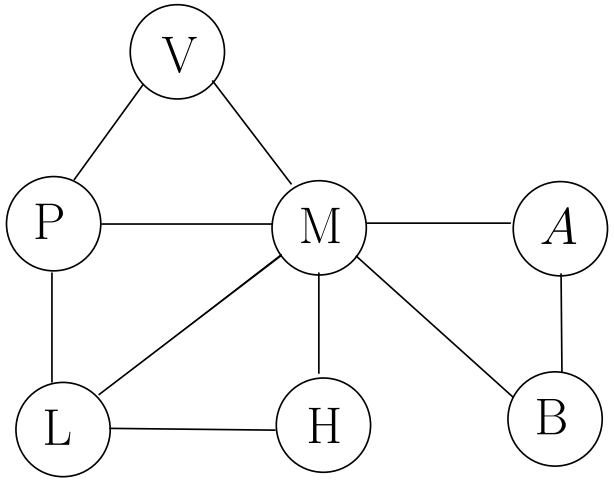


Figure 4: Triangulated Graph

We construct the junction tree by finding the maximal cliques. Note that all cliques are of size three and hence we produce the following:

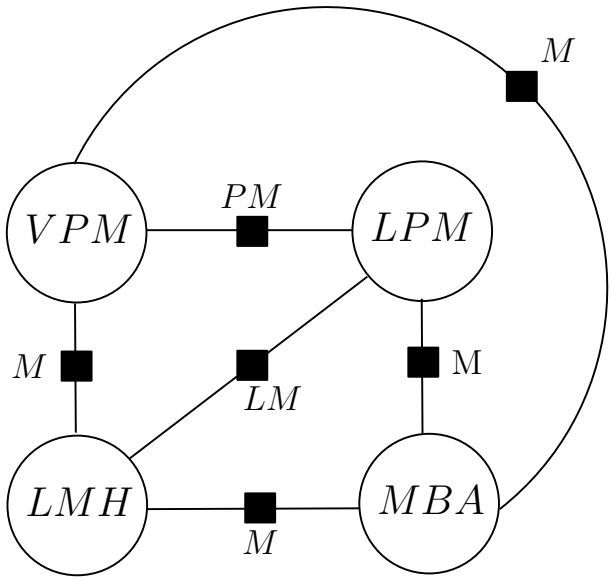


Figure 5: Junction Tree

Finally we find the maximum weight spanning tree (with weights given by the size of the separators. There are several options but we choose the following:

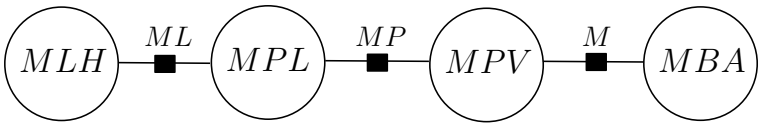


Figure 6: Efficient Junction Tree

We specify the overall potential (after setting the separator potentials to unity) as:

$$p(X) = \frac{1}{Z} f_4(M, L, H) f_3(M, L, P) f_2(M, P, V) f_1(M, B, A)$$

From here we choose the factor potentials according to the conditional probabilities specified in the DAG:

$$\begin{aligned} f_4(M, L, H) &= p(H|M, L) \\ f_3(M, L, P) &= p(L|P) \\ f_2(M, P, V) &= p(P|V)p(M|V)p(V) \\ f_1(M, B, A) &= p(B|M, A)p(A) \end{aligned}$$

Part (d) We wish to calculate:

$$p(A^1|B^1, H^1) = \frac{p(A^1, B^1, H^1)}{p(B^1, H^1)}$$

$$= \frac{\sum_M p(A^1, B^1, H^1, M)}{\sum_A \sum_M p(A, B^1, H^1, M)} = \frac{\sum_M p(A^1, B^1, H^1, M)}{\sum_A \sum_M p(A, B^1, H^1, M)}$$
$$= \frac{\sum_M p(A^1, B^1, M|H^1)p(H^1)}{\sum_A \sum_M p(A, B^1, M|H^1)p(H^1)}$$
$$= \frac{\sum_M p(A^1, B^1, M|H^1)}{\sum_A \sum_M p(A, B^1, M|H^1)}$$

Therefore if we calculate $p(A, B^1, M|H^1)$ and carry out the appropriate marginalisation then we are done.

Note that by the Shafer-Shenoy and since the junction tree is just a chain, we have:

$$\begin{aligned} p(A, B^1, M) &= f_1(A, B^1, M)M_{2\to 1} \\ &= f_1(A, B^1, M) \sum_{MPV/M} f_2(M, P, V)M_{3\to 2} \\ &= f_1(A, B^1, M) \sum_{MPV/M} f_2(M, P, V) \sum_{MLP/MP} f_3(M, L, P)M_{4\to 3} \\ &= f_1(A, B^1, M) \sum_{MPV/M} f_2(M, P, V) \sum_{MLP/MP} f_3(M, L, P) \sum_{MLH/ML} f_4(M, L, H) \end{aligned}$$

We then substitute in the conditional probabilities that we previously computed in **(b)**, as well as condition on H^1:

$$p(A, B^1, M|H^1) = p(B^1|M, A)p(A) \sum_{P,V} p(P|V)p(M|V)p(V) \sum_L p(L|P)p(H^1|M, L)$$

Note that all that we have all these conditional probabilities as the tables specified in **(b)**. We are therefore able to evaluate this probability. If we carry out the marginalisation of A and M of this quantity then we recover the denominator of the term $p(A^1|B^1, H^1)$, and if we set A^1 and marginalise over M then we recover the numerator. Hence we are able to compute the required conditional probability.

Part (e)

We believe that $p(A^1|B^1, H^1) > p(A^1)$. This is because observing blue spots B^1, which is sometimes a symptom of A^1, gives evidence that the patient has Applosis. We believe this is true because Applosis is said to be very rare, and hence upon observing any symptom of A^1, the possibility of the patient having Applosis should at least be considered.

We believe that $p(A^1|B^1) > p(A^1|B^1, H^1)$. This is because upon observing H^1, we are much more likely to think the patient has Microsoftus since H^1 and B^1 are both symptoms of M^1 that happen *usually*. In order to keep the consistency of the joint probability after conditioning on H^1 after conditioning on B^1, in some sense the probability $P(A^1|B^1) \to P(A^1|B^1, H^1)$. Mathematically we can provide a stronger justification: Note we have:

$$p(A^1|B^1) = p(A^1|B^1, H^1)p(H^1) + p(A^1|B^1, H^0)p(H^0)$$
$$\text{now assume } p(A^1|B^1) > p(A^1|B^1, H^1)$$

$$\begin{aligned}
&\iff p(A^1|B^1,H^1)p(H^1) + p(A^1|B^1,H^0)p(H^0) = p(A^1|B^1,H^1) \\
&\iff p(A^1|B^1,H^1)(1-p(H^1)) < p(A^1|B^1,H^0)p(H^0) \\
&\iff p(A^1|B^1|H^1) < p(A^1|B^1,H^0)\frac{p(H^0)}{(1-p(H^1))}
\end{aligned}$$

Now assuming that $p(H^0) \approx 1$ we can approximate:

$$p(A^1|B^1,H^1) < p(A^1|B^1,H^0)$$

So if we are confident in saying that the probability of having Applosis, given blue spots and a high temperature is lower than the probabilility of having Applosis given blue spots and **not** having a high temperature, then we can conclude $p(A^1|B^1) > p(A^1|B^1,H^1)$. This seems like a reasonable statement to conclude since, as stated earlier, observing a high temperature means the disease is much more likely to be M^1. In summary we conclude:

$$p(A^1) < p(A^1|B^1,H^1) < p(A^1|B^1)$$

Part (f) The new DAG representing the hypothesis that L induces cravings of P is:

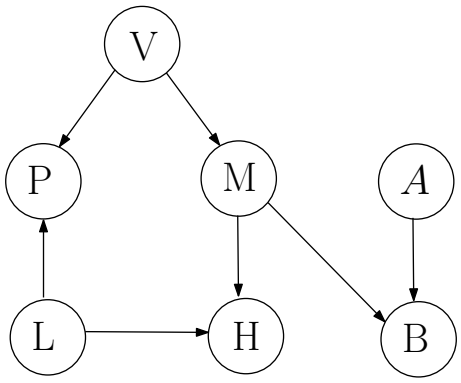


Figure 7: New DAG

Then the fact that P is now a collider we have different conditional probabilities:

Old DAG	New DAG
$L \perp\!\!\!\perp V \mid P$	$L \not\perp\!\!\!\perp V \mid P$
$L \not\perp\!\!\!\perp V \mid \varnothing$	$L \perp\!\!\!\perp V \mid \varnothing$
$H \perp\!\!\!\perp V \mid \{P, M\}$	$H \not\perp\!\!\!\perp V \mid \{P, M\}$

Part (g)

Let original model M_1 have $X_i \sim Bern(\theta_i)$ where $\theta_i \sim Beta(\alpha_i, \beta_i)$, and the new model M_2 have parameters $X_i^{'} \sim Bern(\theta_i^{'})$ where $\theta_i^{'} \sim Beta(\alpha_i^{'}, \beta_i^{'})$. Note that we can factorise the joint probability of the new DAG as:

$$P(X) = p(P|V,L)p(M|V)p(H|M,L)p(B|M,A)p(A)p(V)p(L)$$

We also write (assuming independence)

$$\begin{aligned} p(\theta_{m_i}|m_i) &= p(\theta_A, \theta_M, \theta_L, \theta_B, \theta_V, \theta_H, \theta_P|m_i) \\ &= p(\theta_A|m_i)p(\theta_M|m_i)p(\theta_L|m_i)p(\theta_B|m_i)p(\theta_V|m_i)p(\theta_H|m_i)p(\theta_P|m_i) \end{aligned}$$

Then we evaluate the Bayesian model selection criterion, using we have n data points:

$$\begin{aligned} \frac{P(M_1|D)}{P(M_2|D} &= \frac{P(D|M_1)P(M_1)}{P(D|M_2)P(M_2)} = \frac{P(D|M_1)}{P(D|M_2)} \\ &\text{(assuming both models are equally likely a priori)} \\ &= \frac{\int_{\theta_{m_1}} \prod_i p(V_i, P_i, H_i, B_i|\theta_{m_2})p(\theta_{m_1}|m_1)d\theta_{m_1}}{\int_{\theta_{m_2}} \prod_i p(V_i, P_i, H_i, B_i|\theta_{m_2})p(\theta_{m_2}|m_2)d\theta_{m_2}} \\ &= \frac{\int_{\theta_{m_1}} \prod_i \sum_{A,M,L} p(V_i, P_i, H_i, B_i, A, M, L|\theta_{m_2})p(\theta_{m_1}|m_1)d\theta_{m_1}}{\int_{\theta_{m_2}} \prod_i \sum_{A,M,L} p(V_i, P_i, H_i, B_i, A, M, L|\theta_{m_2})p(\theta_{m_2}|m_2)d\theta_{m_2}} \\ &= \frac{\int_{\theta_{m_1}} \prod_i \sum_{A,M,L} p(P_i|V_i)p(M|V_i)p(L|P_i)p(H_i|M, L)p(B_i|M, A)p(A)p(V_i)p(\theta_{m_1}|m_1)d\theta_{m_1}}{\int_{\theta_{m_2}} \prod_i \sum_{A,M,L} p(P_i|V_i, L)p(M|V_i)p(H_i|M, L)p(B_i|M, A)p(A)p(V_i)p(L)p(\theta_{m_2}|m_2)d\theta_{m_2}} \end{aligned}$$

where we note that the model parameters specify the factorisation of the DAG in terms of the conditional probabilities. We could then set the α and β values for each prior in θ_{m_1} and θ_{m_2} by centering the prior distribution on each value given by the conditional probability tables in **(b)**. We are then equipped to try and perform these integrals using the data. We could then distinguish between these models by determining whether the fraction was less than or greater than one. If the fraction was less than one, we would conclude that the probability of the data given the new model was higher than given the old model, and so we could say that the hypothesis Linuxitis induces cravings for Pizza would be more likely than Pizza causing Linuxitis. The data would therefore be adequate to distinguish between the hypothesis.

Question 3

Part (a) Given $\mathbf{w} \sim \mathcal{N}(\mathbf{m_0}, \mathbf{C}) = \mathcal{N}(\begin{pmatrix} b \\ a \end{pmatrix}, \begin{pmatrix} 100^2 & 0 \\ 0 & 10^2 \end{pmatrix})$ and $y_i \sim \mathcal{N}(\mathbf{w}^T\mathbf{x}_i, \sigma^2)$, then we know that the posterior for $\mathbf{w}$ given $\mathbf{y}$ has covariance and mean given by

$$\begin{aligned} \Sigma_W &= \left(\frac{XX^T}{\sigma^2} + C^{-1}\right)^{-1} \\ \bar{\mathbf{w}} &= \Sigma_W \left(C^{-1}\mathbf{m}_0 + \frac{XY^T}{\sigma^2}\right) \end{aligned}$$

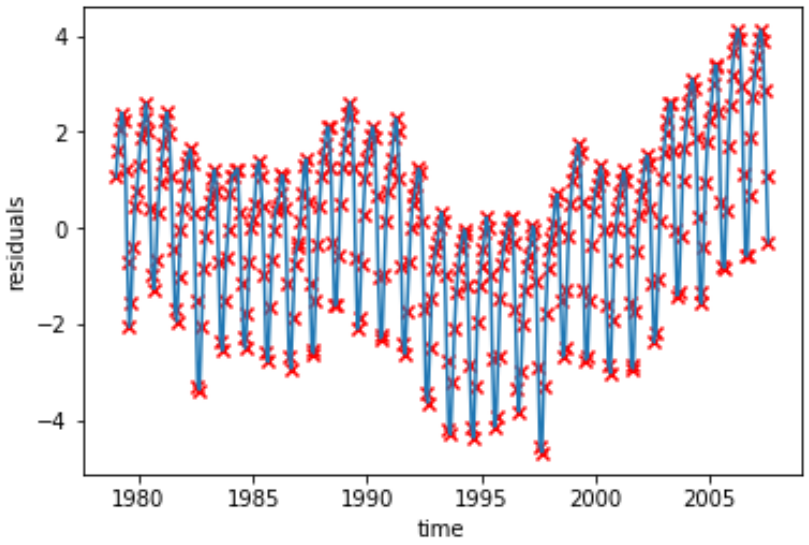
Therefore using our data we find that (approximately written down):

$$\mathbf{w} \mid X, Y, \mathbf{m}_0, \mathbf{C}, \sigma^2 \sim \mathcal{N}(\begin{pmatrix} -2776 \\ 1.572 \end{pmatrix}, \begin{pmatrix} 165.8 & -0.083 \\ -0.083 & 0.00004 \end{pmatrix})$$

The code used to accomplish this is given below:

```
import numpy as np

d_temp = np.loadtxt("co2.txt")
nData, nDim = d_temp.shape
d = np.zeros(shape=(nData, nDim-1))

for cIter in range(nData):
    d[cIter,:] = [d_temp[cIter,0] + (d_temp[cIter,1] - 1)/12, d_temp[cIter,2]]

Y = d[:,1].reshape((1,nData))
X = np.array([np.ones(nData),d[:,0]])
C = np.array([[100**2, 0],[0, 10**2]])
C_inv = np.linalg.inv(C)

sigma_w = np.linalg.inv(X @ X.T + C_inv)
mean_w = sigma_w @ (C_inv @ np.array([[360], [0]]) + X @ Y.T)

a_map = mean_w[1]
b_map = mean_w[0]
```

Part (b) Below is a plot of $g_{obs}(t)$:



The residuals gave reported mean ≈ -0.0009 and variance ≈ 3.3141. Therefore these residuals largely do not conform to our expectations over $\epsilon(t) \sim \mathcal{N}(0, 1)$. The mean conforms to our expectations as it is approximately 0, however the variance of 3.31 does not since it is much greater than 1. If the residuals followed the i.i.d assumption we would also expect to see the residuals randomly scattered about the $y = 0$ line, however we also clearly see that the residuals take a periodic shape, and so they are not i.i.d.

Part (c)

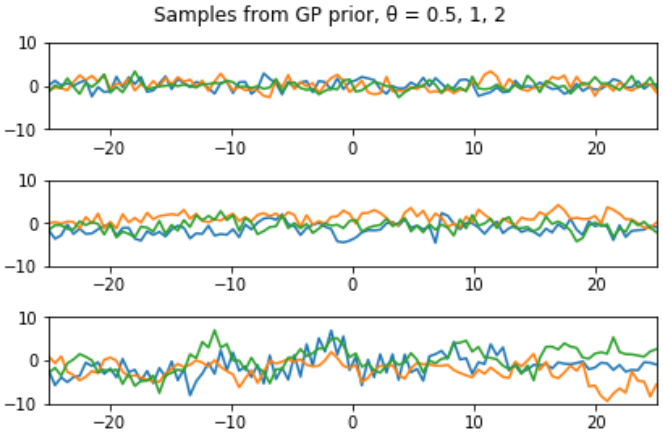
The following code is what was used to generate the GP samples, given a kernel k.

```
def kernel(k, h, x_1, x_2):
    """
    Returns the kernel covariance given by input points x_1, x_2.
    -- k is the kernel function
    -- h is a list of hyperparameters
    -- x_1 and x_2 are the input points
    """
    n_1 = x_1.shape[0]
    n_2 = x_2.shape[0]
    K = np.zeros(shape=(n_1, n_2))

    for i in range(n_1):
        for j in range(n_2):
            K[i][j] = k(x_1[i,:], x_2[j,:], h)
    return K

def k_1(s, t, h):
    """
    Definition of the kernel given in the question
    -- h is a list containing the hyperparameters
    """
    theta, tau, sigma, phi, eta, ups = h[0], h[1], h[2], h[3], h[4], h[5]
    term_1 = np.exp(-2*np.sin(np.pi*(s-t)/tau)**2/sigma**2)
    term_2 = phi**2*np.exp(-(s-t)**2/(2*eta**2))
    term_3 = 0
    if(s == t):
        term_3 = ups**2
    return theta**2*(term_1 + term_2) + term_3

def GP_sample(k, h, x, s):
    """
    Generates s samples from drawn from a GP with zero mean.
    -- x is an (n by 1) set of input points
    -- k is the kernel function
    -- h is the list of hyperparameters
    -- s is the number of samples

    -- K is the (n by n) kernel matrix evaluated at the points x.
    -- L is the Cholesky decomposition
    -- u is a set of s random normal vectors length n
    -- f is the function evaluated at input points x
    """
    n, d = x.shape

    K = kernel(k, h, x, x)
    L = np.linalg.cholesky(K)
    u = np.random.normal(size=(n,s))
    f = L @ u

    return f
```

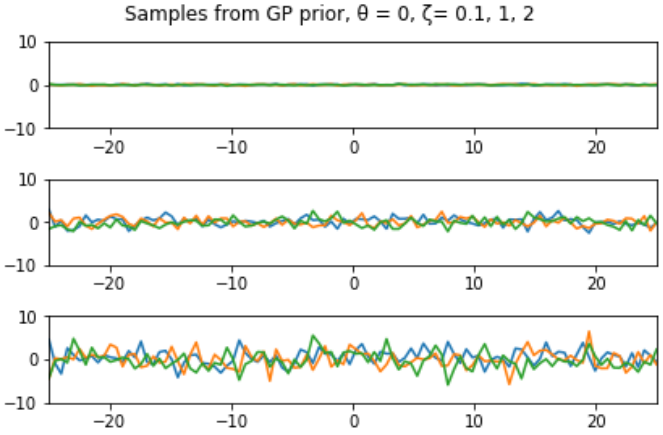
Part (d)

In the following section we plot GP samples using various settings of the hyper parame-

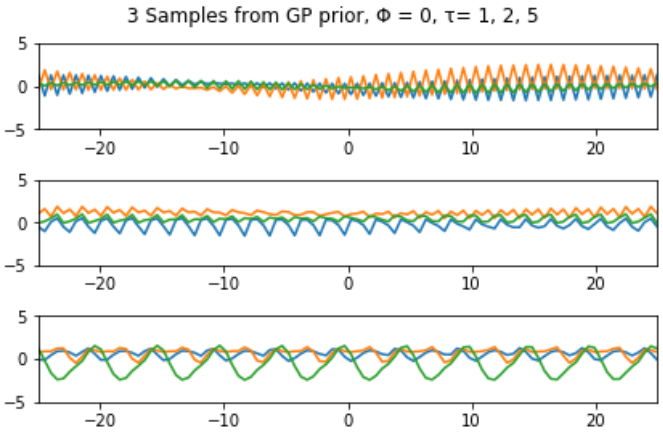
ters. In general, when testing how the samples vary with respect to a single parameter, we set all other parameters to one (and $\zeta = 0.01$) and vary the parameter we are interested in.
Testing θ:



From this plot we see clearly that θ is an overall **amplitude** parameter. This makes sense since θ is a multiplicative constant over both the exponential periodic and exponential squared terms in the kernel.
Testing ζ:
In this part we set $\theta = 0$, so that the only parameter in the kernel is ζ



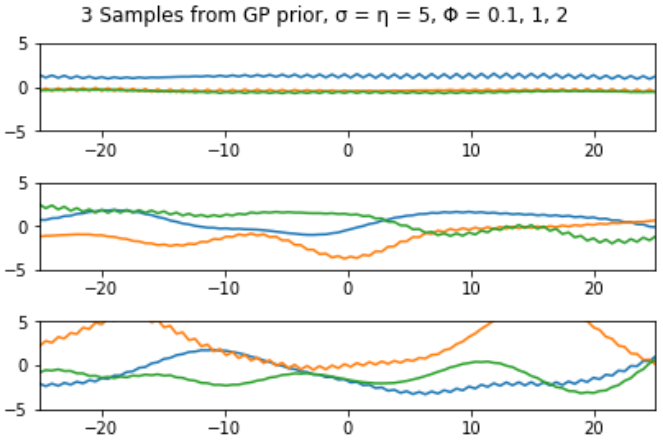
From the kernel formula we see that ζ is an **independent noise** parameter, i.e. since it exists only on the diagonal of the kernel matrix, it describes the independent noise at any one point. This is seen from the plot - for larger ζ, the amplitude of the noise increases.
Testing τ:
In this part we set $\phi = 0$, so that we can observe how the samples vary with respect to just the exponential sin term.



From the kernel formula we see that τ controls the **periodicity** of the exponential sin term. Here we have varied $\tau = 1, 2, 5$, and we can see clearly from the plot, that for larger τ the periodicity decreases.

Testing ϕ:

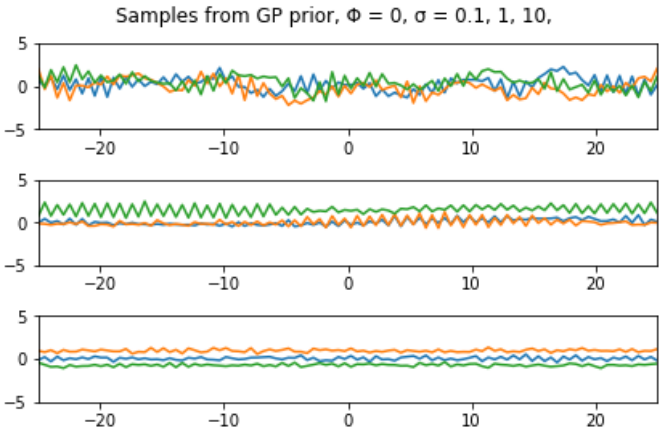
In this part we set $\sigma = \eta = 5$ in order to see more clearly how the hyperparameter ϕ controls the shape of the samples:



From the kernel formula, we see that ϕ controls the **relative importance** between the exponential sin and exponential quadratic term. This is seen clearly from this plot - the larger ϕ is, the more overall trend we see in the shape of the samples (since the exponential squared term describes the overall trend).

Testing σ:

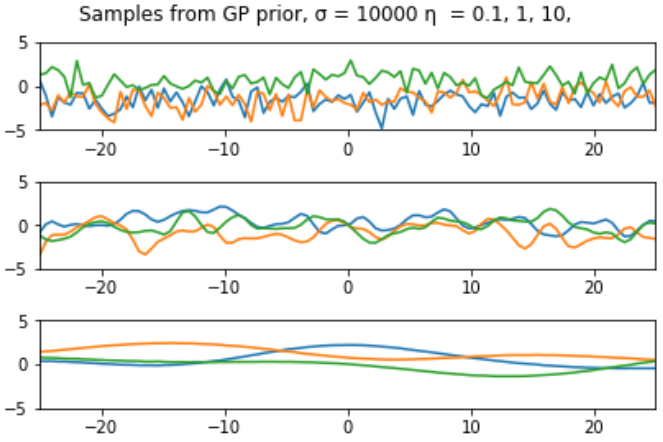
In this testing phase we set $\phi = 0$ in order to focus on the exponential sin term, and how it varies with respect to σ:



From these plots we see tat σ acts as a **smoothness** parameter - for larger σ, the smoothness of the samples increases. Note that equivalently σ can be seen as a **length-scale** - i.e. varying σ "stretches" the x-axis.

Testing η:

The same conclusions follow for η, except that η controls the scale of the exponential quadratic term. σ was set large here so that the exponential sin term became insignificant.



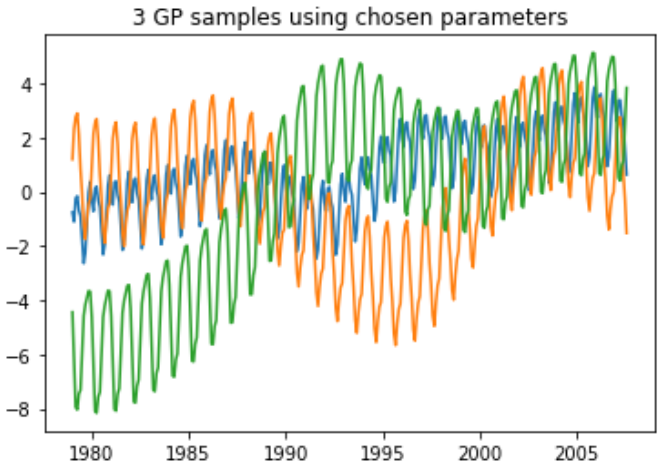
We see clearly from the plot that larger η increases the **smoothness of the trends**, and therefore also acts as a **length-scale**.

Part (e) Considering the plot of the residuals g_{obs} we note that the amplitude of each period is approximately 4. Therefore we set $\theta \approx 2$. We also note that each year consists of one period, and therefore the periodicity is one, i.e. we set $\tau \approx 1$. We should also set ζ small since the overall curve appears mostly smooth, therefore we set $\zeta = 0.01$. For ϕ we observe that in the residual plot, we see an increase of about 2, and therefore model this trend with $\phi = 1$ (since $\theta = 2$). For η we observe, that we are operating on a length scale on the training data of 28 year and hence require $2\eta^2 = 28 \implies \eta \approx 3.75$. Lastly, we set $\sigma = 1$ since each period operates on a length scale of one year. In summary, we have:

$$\theta = 2, \quad \tau = 1, \quad \sigma = 1$$

$$\phi = 1, \quad \eta = 3.75, \quad \zeta = 0.01$$

Using these parameters, and plotting GP samples using the training data as the input points, we get plots:



We see that the general structure of the samples all generally mimic the residual plot in part **(b)**, and therefore we can say that our chosen hyperparameters are suitable for modelling $g_{obs}(t)$ using a zero mean GP.

Part (f) We can write the joint over the (noisy) observed information $\mathbf{y}$, X and the predictive density $\mathbf{f}$ over test points X_* as:

$$\begin{pmatrix} y \\ f_* \end{pmatrix} \sim \mathcal{N}(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K(X,X)+\sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{pmatrix})$$

Which gives a predictive posterior density, after conditioning on $\mathbf{y}$:

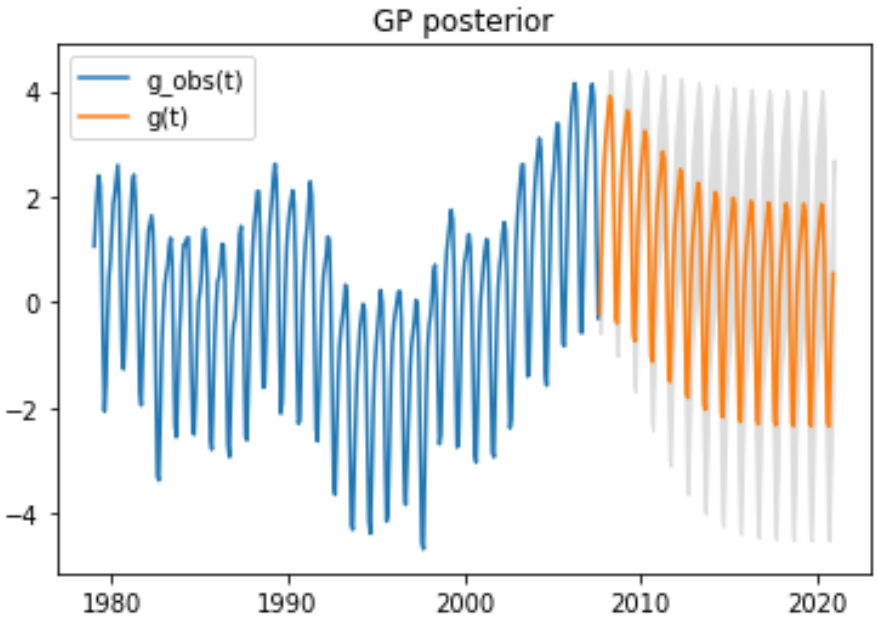
$$f \sim \mathcal{N}(E[f], cov[f]) \text{ where} \tag{11}$$
$$E[f] = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} y \tag{12}$$
$$cov[f] = K(X_*, X_*)K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \tag{13}$$

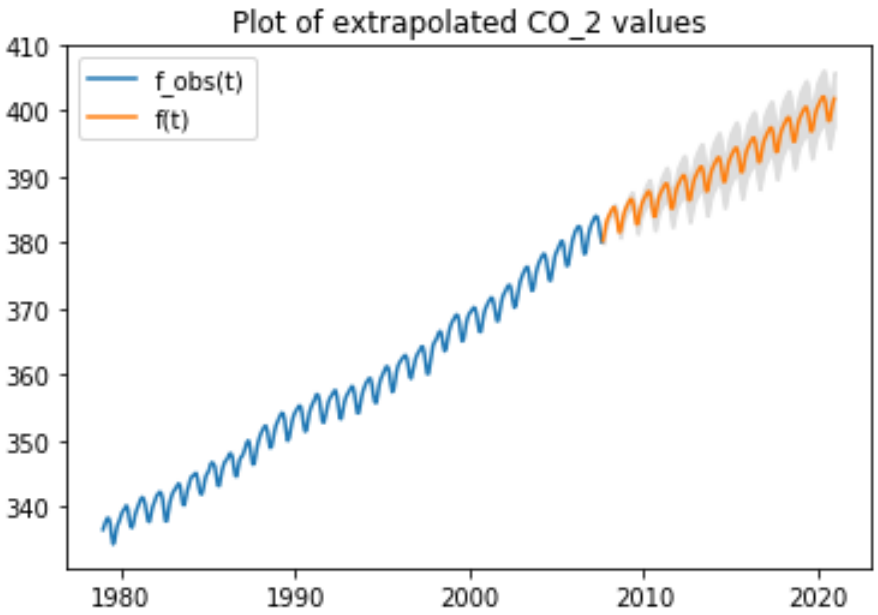
The code used to calculate the posterior mean and variance estimates as well as the extrapolated CO_2 values $f(t)$, given the training data is below:

```
def GP_posterior(x_train, x_test, y_train, k, h):
    """
    Function to compute the posterior mean and variance of the GP
    -- x_train, y_train is the training data
    -- x_test is the set of x points we wish to evaluate the GP on.
    -- k is the given kernel function
    -- h is a list of the chosen hyperparameter values
    """

    kernel_11 = kernel(k, h, x_train, x_train)
```

```
    kernel_12 = kernel(k, h, x_train, x_test)

    kernel_22 = kernel(k, h, x_test, x_test)

    n = x_train.shape[0]

    kernel_solve = kernel_12.T @ np.linalg.pinv(kernel_11 + np.eye(n))

    post_mean = kernel_solve @ y_train

    post_var = kernel_22 - kernel_solve @ kernel_12

    return post_mean, post_var

h_chosen = [2,1,1,2,3.75,0.01]
p_m, p_v = GP_posterior(x_train, x_test, y_train, k_1, h_chosen)

# One standard deviation
stdv = np.sqrt(np.diag(p_v))

# Computing the function f(t)
f_t = a_map*x_test + b_map + p_m
```

We then plotted the results, including the shaded grey area to indicate one standard deviation from the mean:





The behaviour of the extrapolation largely conforms to our expectations, this is because:

1. The trend of the CO_2 data continues in an upward fashion, with similar gradient to the data
2. The standard deviance error bars increase in magnitude as the we extrapolate further away from the test data - this conforms to the expectation that we should be less certain about our prediction as we go further into the future.
3. The shape of the curve remains the same for the extrapolation - this is to be expected since the observed values of CO_2 levels display a uniformly periodic shape for every single year, so we would not expect our mean estimates to deviate from this shape.
4. The slope of the extrapolated mean a_{MAP} is slightly tilted toward 0, i.e. lower than a linear regression line estimated from the training data. This is expected since the prior on a has zero mean, and therefore we would expect this to have an impact on the slope of the regression line given by a_{MAP}

The sensitivity of the conclusions to the settings of the hyperparameter is different for different hyperparameters:

θ Varying θ has almost no impact on the posterior mean estimates (as we would expect since it is just a scaling parameter), however as it increases we see that the standard deviation error bars increase.

τ Has a little impact on the final posterior means.

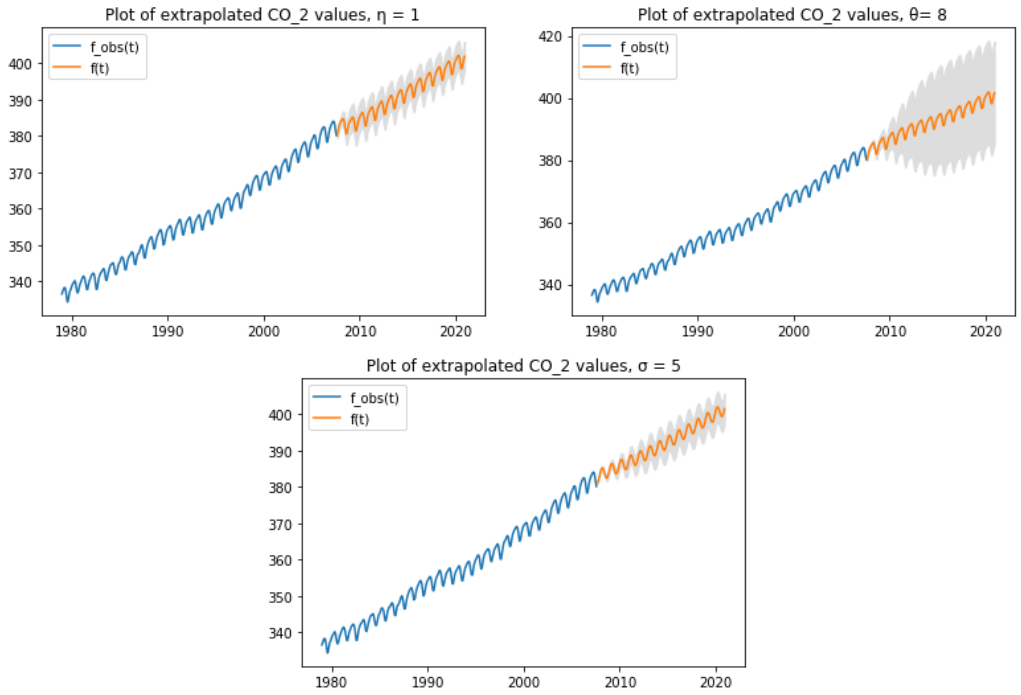
σ Varying σ has almost no impact on either the posterior mean or variances (except for especially large values).

ϕ Varying ϕ has little impact on the posterior mean, and a large impact on the posterior variances.

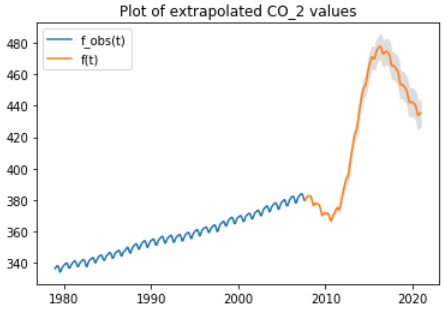
η We see that η has little impact on the posterior mean but a large impact on the posterior variances.

ζ For ζ large enough we see that this independent noise parameter dominates the variance expression and hence can have a large impact on the final posterior variances.

Some examples of varying the hyperparameters is given below (where we vary one hyperparameter independently and keep the others the same):



Overall we conclude that the posterior estimates are only slightly sensitive to the settings of the hyperparameters. Given certain settings (especially changing η and θ, the standard deviation error bars can become much larger and therefore we become a lot less confident about our predictions. However, the overall shape of the trend remains mostly the same. This is largely because of the inclusion of the noise parameter σ_n that we introduced when deriving the posterior prediction. This term greatly stabilises the matrix inversions in equations (12) - (13), since this noise parameter prevents problems from 'overfitting' to the data. If we don't include this term for noisy observations, the shape of the posterior can change wildly for even slight changes in the hyperparameters. For example, in the below plot, we set $\theta = 8$, as we did in the example above, but get an extremely different/incorrect result:



Part (f) This procedure is not fully Bayesian because we have used a MAP estimate for a and b after performing Bayesian linear regression. To make the procedure fully bayesian, we would make predictions for a particular test point by averaging out over all possible parameter values of a and b via integration. I.e the predictive distribution for a test point x_* is given by averaging out over all possible linear models w.r.t the gaussian posterior.