Comp 0085 Assignment 4

Alexandra Maria Proca (SN: 20047328)

December 9, 2020

Question 1.

a. For graph 2, the conditional independence relationships that hold for variable C are
$C \perp\!\!\!\perp A|B$
$C \perp\!\!\!\perp A|\{B, D\}$
For graph 4, the conditional independence relationships that hold for variable C are
$C \perp\!\!\!\perp A|B$
$C \perp\!\!\!\perp A|\{B, D\}$
For graph 6, the conditional independence relationships that hold for variable C are
$C \perp\!\!\!\perp A|B$
$C \perp\!\!\!\perp A|\{B, D\}$
For graph 8, the conditional independence relationships that hold for variable C are
$C \perp\!\!\!\perp A|B$
$C \perp\!\!\!\perp A|\{B, D\}$

b. Graphs 1 and 2 are equivalent and belong to the same set S_1, as they express all the same marginal and conditional independence relationships between their variables. Namely,
$A \perp\!\!\!\perp C|B$
$A \perp\!\!\!\perp C|\{B, D\}$
$A \perp\!\!\!\perp D|\emptyset$
$A \perp\!\!\!\perp D|B$
$A \perp\!\!\!\perp D|\{B, C\}$
$B \perp\!\!\!\perp D|\emptyset$
$B \perp\!\!\!\perp D|A$

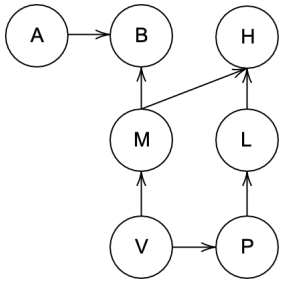
Graphs 3 and 5 are equivalent and belong to the same set S_2, as they express all the same marginal and conditional independence relationships between their variables. Namely,
$A \perp\!\!\!\perp C|B$
$A \perp\!\!\!\perp C|\{B, D\}$
$A \perp\!\!\!\perp D|B$
$A \perp\!\!\!\perp D|C$
$A \perp\!\!\!\perp D|\{B, C\}$
$B \perp\!\!\!\perp D|C$
$B \perp\!\!\!\perp D|\{A, C\}$

Graphs 4,6,7, and 8 are equivalent and belong to the same set S_3, as they express all the same marginal and conditional independence relationships between their variables. Namely,
$A \perp\!\!\!\perp C|B$
$A \perp\!\!\!\perp C|\{B, D\}$
$A \perp\!\!\!\perp D|B$
$A \perp\!\!\!\perp D|\{B, C\}$

Because all conditional independence relationships in S_3 are exhibited in both S_1 and S_2, S_1 and S_2 are subsumed by S_3.

Question 2.

a. The directed graph below represents the relationships between the given variables. Namely, vulcans have higher probability of getting Microsoftus than humans ($V \to M$), most vulcans like pizza and some humans like pizza ($V \to P$), microsoftus usually causes high temperature and blue spots on the face ($M \to H$, $M \to B$), Linuxitis always causes high temperature ($L \to H$), Applosis sometimes causes blue spots on the face ($A \to B$), and a recent study suggests that excess pizza consumption increases risk of Linuxitis ($P \to L$).



b. There are about four times as many humans as vulcans on the ship. Thus, the probability of being a vulcan ($V = 1$) is

$P(V = 1)$	$P(V = 0)$
0.20	0.80

Applosis is a very rare disease. Thus,

$P(A = 1)$	$P(A = 0)$
0.02	0.98

Vulcans have a higher probability of getting Microsoftus than humans and Microsoftus is a rare disease. Thus,

V	$P(M = 1)$	$P(M = 0)$
0	0.06	0.94
1	0.15	0.85

Most vulcans like pizza and some humans like pizza. Thus,

V	$P(P = 1)$	$P(P = 0)$
0	0.30	0.70
1	0.80	0.20

A recent study suggests that excess pizza consumption increases risk of Linuxitis. Thus,

P	$P(L = 1)$	$P(L = 0)$
0	0.02	0.98
1	0.15	0.85

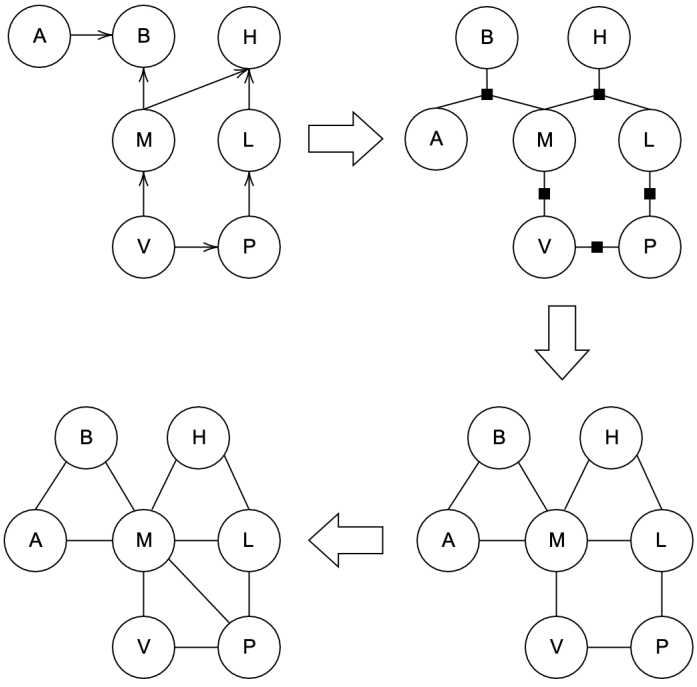
Microsoftus usually causes high temperature and Linuxitis always causes high temperature. Thus,

M	L	$P(H=1)$	$P(H=0)$
0	0	0.05	0.95
1	0	0.70	0.30
0	1	1.0	0.0
1	1	1.0	0.0

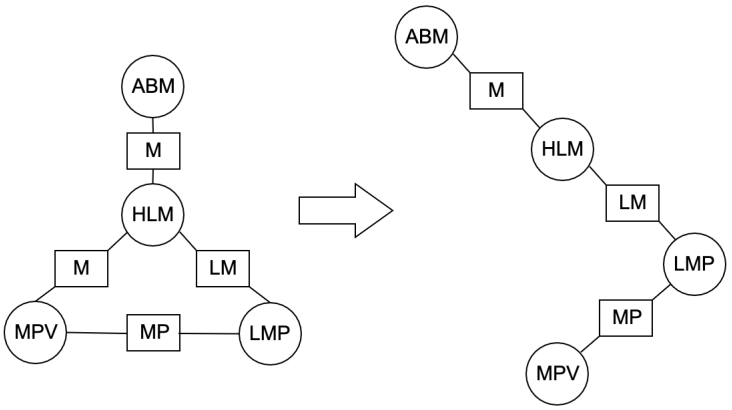
Microsoftus usually causes blue spots on the face and Applosis sometimes causes blue spots on the face. Thus,

M	A	$P(B=1)$	$P(B=0)$
0	0	0.01	0.99
1	0	0.70	0.30
0	1	0.20	0.80
1	1	0.90	0.10

c. The directed graph is transformed into an intermediate factor graph, an undirected graph, and a chordal graph, as shown below. Because the original undirected graph contains a loop of four nodes, it must be triangulated in order to produce a chordal graph. Using minimum deficiency search variable elimination order, a possible elimination order is: A (0 new edges), B (0), H (0), V (1 new edge between M and P), P (0), and M (0). Thus, an edge is placed between M and P. An equally plausible elimination order is A, B, H, P (1), V, and M, which would yield an edge between L and V.



A weighted graph is constructed with the maximal cliques ($C_{ABM}, C_{HLM}, C_{MPV}, C_{LMP}$) of the chordal undirected graph as nodes and the intersecting clique variables as separators (S_M, S_M, S_{ML}, S_{MP}). Defining the weight of edges as the size of the separators, the maximum-weight spanning tree is created as the final junction tree, as shown below.



Each clique factor can be represented in terms of the conditional probabilities.

Clique Factor	Conditional Probability
f_{ABM}	$P(B\|A,M)P(A)$
f_{HLM}	$P(H\|L,M)$
f_{LMP}	$P(L\|P)$
f_{MPV}	$P(M\|V)P(P\|V)P(V)$

d. Shafer-Shenoy propagation can be used to recursively calculate the messages of the junction tree in order to find the message passed from clique C_{HLM} to C_{ABM}, $M_{HLM\to ABM}(X_{S_M})$, which can then be used to calculate the marginal probability $P(X_{C_{ABM}})$ and find $P(A=1|H=1,B=1)$. This can be done in two ways.

The marginal probability of C_{ABM} can be written as
$P(X_{C_{ABM}}) = f_{ABM}(X_{C_{ABM}})M_{HLM\to ABM}(X_{S_M})$
Solving for $M_{HLM\to ABM}(X_{S_M})$ recursively,

$$M_{HLM\to ABM}(X_{S_M}) = \sum_{X_{C_{HL}}} f_{HLM}(X_{C_{HLM}}) \left(\sum_{X_{C_P}} f_{LMP}(X_{C_{LMP}}) \left(\sum_{X_{C_V}} f_{MPV}(X_{C_{MPV}}) \right) \right)$$
$$= \sum_{H,L} P(H|L,M) \left(\sum_{P} P(L|P) \left(\sum_{V} P(M|V)P(P|V)P(V) \right) \right)$$
$$= \sum_{H,L} P(H|L,M) \left(\sum_{P} P(L|P)P(M)P(P) \right)$$
$$= \sum_{H,L} P(H|L,M)P(L)P(M)$$
$$= \sum_{H} P(H|M)P(M)$$
$$= \sum_{H} P(M|H)P(H)$$
$$= P(M)$$

Substituting the computed message into the marginal clique probability,
$P(X_{C_{ABM}}) = f_{ABM}(X_{C_{ABM}})P(M)$
$= P(B|A,M)P(A)P(M)$
Because B is observed,
$\implies P(B=1|A,M)P(A)P(M)$

$= P(A, B = 1, M)$
Solving for A conditioned on $B = 1, H = 1$,
$$\implies \frac{\sum_{M} P(A = 1, B = 1, M)}{\sum_{A,M} P(A, B = 1, M)P(H = 1)}$$
$= P(A = 1|B = 1, H = 1)$

Alternatively, $M_{HLM \to ABM}(X_{S_M})$ can be solved for with the observation of $H = 1$.
$$M_{HLM \to ABM}(X_{S_M}) = \sum_{L} P(H = 1|L, M) \left(\sum_{P} P(L|P) \left(\sum_{V} P(M|V)P(P|V)P(V) \right) \right)$$
$$= \sum_{L} P(H = 1|L, M) \left(\sum_{P} P(L|P)P(M)P(P) \right)$$
$$= \sum_{L} P(H = 1|L, M)P(L)P(M)$$
$= P(H = 1|M)P(M)$
$= P(H = 1, M)$

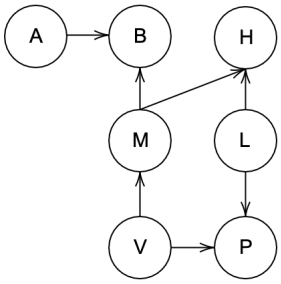
Substituting the computed message into the marginal clique probability,
$P(X_{C_{ABM}}) = f_{ABM}(X_{C_{ABM}})P(H = 1, M)$
$P(B|A, M)P(A)P(H = 1, M)$
Because B is observed,
$\implies P(B = 1|A, M)P(A)P(H = 1, M)$
$= P(A, B = 1, M, H = 1)$
Solving for A conditioned on $B = 1, H = 1$,
$$\implies \frac{\sum_{M} P(A = 1, B = 1, M, H = 1)}{\sum_{A,M} P(A, B = 1, M, H = 1)}$$
$= P(A = 1|B = 1, H = 1)$

e. The probability `P(patient has Applosis|patient has blue spots on face and high temperature) < P(patient has Applosis)` because $B = 1, H = 1$ decreases the probability of the patient having Applosis from the baseline probability $P(A = 1) = 0.02$. Applosis does not cause a high temperature and the probability of having both a high temperature ($P(H = 1) = 0.05$) and Applosis ($P(A = 1) = 0.02$) is very low. An alternative explanation for having both a high temperature and Applosis could be having Microsoftus or Linuxitis- both of which have low probabilities ($P(M = 1) = 0.06, P(L = 1) = 0.02$). Thus, it would be very unlikely that the patient had both Microsoftus/Linuxitis and Applosis. The presence of both blue spots and a high temperature increases the probability that the patient has Microsoftus, and thus decreases the probability the patient has Applosis. Therefore, $P(A = 1|B = 1, H = 1) < P(A = 1)$.

The probability `P(patient has Applosis|patient has blue spots on face and high temperature) < P(patient has Applosis|patient has blue spots on face)` because the presence of a high temperature decreases the probability of having Applosis and increases the probability of having Microsoftus. Because Microsoftus usually causes a high temperature and blue spots ($P(H = 1|M = 1) = P(B = 1|M = 1) = 0.70$), as opposed to Applosis which only sometimes causes blue spots ($P(B = 1|A = 1) = 0.20$, $P(H = 1) = 0.05$), and because Microsoftus is less rare than Applosis, the presence of both B and H increases the probability the patient has Microsoftus and decreases the probability that the patient has Applosis. Because Microsoftus is rare ($P(M = 1) = 0.06$) and Applosis is very rare ($P(A = 1) = 0.02$), the probability of having both is very low and thus it is more likely that the patient only has Microsoftus. Comparatively, the presence of only blue spots yields a higher probability

of Applosis, as Applosis sometimes causes blue spots ($P(B = 1|A = 1) = 0.2$), whereas Applosis does not cause high temperature ($P(H = 1) = 0.05$). Therefore, $P(A = 1|B = 1) > P(A = 1|B = 1, H = 1)$ because the patient having only blue spots yields a higher probability the blue spots were caused by Applosis than having both blue spots and a high temperature, which is more probable to have been caused by Microsoftus.

f. The directed graph below would apply if the supposition that Linuxitis induces a craving for pizza were true. As opposed to the previous model, $L \to P$.



Comparing the new graph to the previous, several conditional independence relationships would differ because observing P removes independencies between L and V,M,B. Namely,
$M \perp\!\!\!\perp L|P \Rightarrow M \not\perp\!\!\!\perp L|P$
$V \perp\!\!\!\perp L|P \Rightarrow V \not\perp\!\!\!\perp L|P$
$B \perp\!\!\!\perp L|P \Rightarrow B \not\perp\!\!\!\perp L|P$

g. Bayesian model selection can be used to compare the hypothesis to the one advanced by the original study by treating the data variables (P, H, V, B) as observations in each model and the variables not included in the data (A, L, M) as parameters. The probability of each model given the data can then be computed by integrating over the unobserved parameters and solving using Bayes' theorem.

Solving for the original model m,
$$P(\mathcal{D}|m) = \int_{\Theta_m} P(\mathcal{D}|\boldsymbol{\theta}_m, m)P(\boldsymbol{\theta}_m|m)\ d\boldsymbol{\theta}_m$$
Rewriting in terms of the observed data and unobserved parameters,
$$\int_{A,L,M} \prod_{i=1}^{N} P(B_i|A, M, m)P(A|m)P(H_i|L, M, m)P(L|P_i, m)P(M|V_i, m)P(P_i|V_i, m)P(V_i|m)$$
$$P(A, L, M|m)\ dA\ dL\ dM$$
$$= \prod_{i=1}^{N} P(V_i|m)P(P_i|V_i, m)P(H_i|m)P(B_i|m)$$
$$= \prod_{i=1}^{N} P(V_i, P_i, H_i, B_i|m)$$
$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})} = \frac{\prod_{i=1}^{N} P(V_i, P_i, H_i, B_i|m)P(m)}{\prod_{i=1}^{N} P(V_i, P_i, H_i, B_i)}$$

Similarly, solving for the proposed hypothesis m',

$$P(\mathcal{D}|m') = \int_{\Theta_{m'}} P(\mathcal{D}|\boldsymbol{\theta}_{m'}, m')P(\boldsymbol{\theta}_{m'}|m')\ d\boldsymbol{\theta}_{m'}$$

Rewriting in terms of the observed data and unobserved parameters,

$$\int_{A,L,M} \prod_{i=1}^{N} P(B_i|A,M,m')P(A|m')P(H_i|L,M,m')P(P_i|L,V_i,m')P(L|m')P(M|V_i,m')P(V_i|m')$$
$$P(A,L,M|m')\ dA\ dL\ dM$$
$$= \prod_{i=1}^{N} P(B_i|m')P(H_i|m')P(P_i|V_i,m')P(V_i|m')$$
$$= \prod_{i=1}^{N} P(V_i,P_i,H_i,B_i|m')$$
$$P(m'|\mathcal{D}) = \frac{P(\mathcal{D}|m')P(m')}{P(\mathcal{D})} = \frac{\prod_{i=1}^{N} P(V_i,P_i,H_i,B_i|m')P(m')}{\prod_{i=1}^{N} P(V_i,P_i,H_i,B_i)}$$

Comparing the probability of the proposed hypothesis to the original model,

$$\frac{P(m'|\mathcal{D})}{P(m|\mathcal{D})} = \frac{P(\mathcal{D}|m')P(m')}{P(\mathcal{D}|m)P(m)}$$
$$= \frac{\prod_{i=1}^{N} P(V_i,P_i,H_i,B_i|m')P(m')}{\prod_{i=1}^{N} P(V_i,P_i,H_i,B_i|m)P(m)}$$

The data would likely be adequate to distinguish between hypotheses because the states of the unobserved variables (A, L, M) are likely to be reasonably predicted given the observed data, as their conditional probabilities are relatively unique. Since the models are being compared, the probability ratio between the two is likely to be approximately proportional to one with all variables represented in the data, given the number of variables observed. It would also be important to have enough representative data to be able to generalize the results and sufficiently draw a conclusion.

Question 3.

a. The variables a and b act as the weight and bias on t. Therefore, a and b can be represented as weight $\boldsymbol{w}$ ($\begin{bmatrix} a & b \end{bmatrix}$) and values of 1 can be added as a dimension of T ($\begin{bmatrix} t \\ 1 \end{bmatrix}$), yielding $\boldsymbol{w}T = at + b$. Thus, the posterior mean and covariance over a and b is calculated in terms of $\boldsymbol{w}$ and T using

$\Sigma_{\boldsymbol{w}} = \left(\frac{TT^{\mathrm{T}}}{\sigma^2} + C^{-1}\right)^{-1}$

$\bar{\boldsymbol{w}} = \Sigma_{\boldsymbol{w}} \frac{TY^{\mathrm{T}}}{\sigma^2}$

where $T = \begin{bmatrix} t_1 & t_2 & \dots & t_N \\ 1 & 1 & \dots & 1 \end{bmatrix}$, σ^2 is the noise residual $\epsilon(t) = 1$, Y is a vector of all given CO_2 concentrations, and $C = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix} = \begin{bmatrix} 10^2 & 0 \\ 0 & 100^2 \end{bmatrix}$.

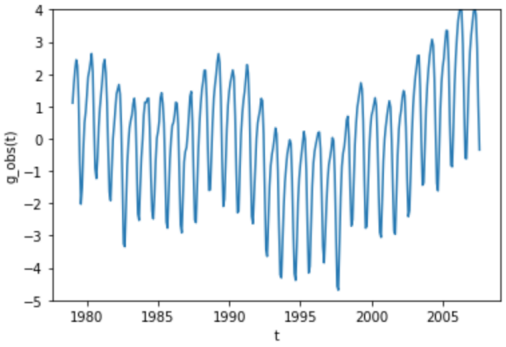
The expressions are implemented and the posterior mean and covariance are found, as shown below.

```
import numpy as np
```

```
from numpy import linalg as LA

data=np.loadtxt('co2.txt')

# Creates t,Y, and T
t=np.zeros(len(data))
Y=np.zeros((len(data)))
for i in range(0,len(data)):
    t[i]=data[i][0]+(data[i][1]-1)/12
    Y[i]=data[i][2]
T=np.ones((2,len(t)))
T[0]=t

# Creates C
C=np.zeros((2,2))
C[0,0]=10**2
C[1,1]=100**2

# Posterior covariance
postcov=LA.inv((T @ T.T)+LA.inv(C))

# Posterior mean
postmean=postcov @ (T @ Y.T)
```

Over a and b, the posterior covariance is found to be
$\Sigma_{\boldsymbol{w}} = \begin{bmatrix} 4.17454539e\text{-}5 & -8.32106418e\text{-}2 \\ -8.32106418e\text{-}2 & 1.65865938e2 \end{bmatrix}$
and the posterior mean is found to be
$\bar{\boldsymbol{w}} = \begin{bmatrix} 1.57052912 & -2.78155763e3 \end{bmatrix}$.

b. The difference is found between the observed function values and the predicted mean function values to calculate $g_{obs}(t)$, $g_{obs}(t) = f_{obs}(t) - (a_{MAP}t + b_{MAP}) = Y - \bar{\boldsymbol{w}}T$. The residuals $g_{obs}(t)$ are plotted below.



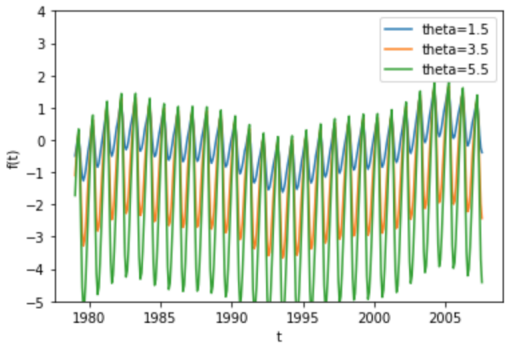
The residuals do not conform to the prior belief over $\epsilon(t)$ because the residuals found have a variance of 3.304, while the prior of $\epsilon(t)$ has a variance of 1 and thus the residuals vary more than the predicted noise. Additionally, the residuals appear to be time-dependent as periodicity and an overall curve (squared exponential) appears in the plot, which may suggest that the Gaussian prior is insufficient as it does not capture the time-dependence of the data.

c. Samples are drawn from a Gaussian process given a covariance kernel function $k(\cdot,\cdot)$ and a vector of input points $\boldsymbol{t}$ in the following function. Additionally, the implementation of the kernel function in part

(d) is included below. A small constant ϵI is added to K in order to stabilize the covariance matrix, as its eigenvalues can decay rapidly, and the Cholesky decomposition is used to find the lower-triangular matrix L of K. The mean ($\boldsymbol{\mu} = \mathbf{0}$) is then added to the product of L and N random Gaussian samples ($\boldsymbol{z} \sim \mathcal{N}(0, I)$), in order to compute $f(\boldsymbol{t})$: $f(\boldsymbol{t}) = \boldsymbol{\mu} + L\boldsymbol{z}$.

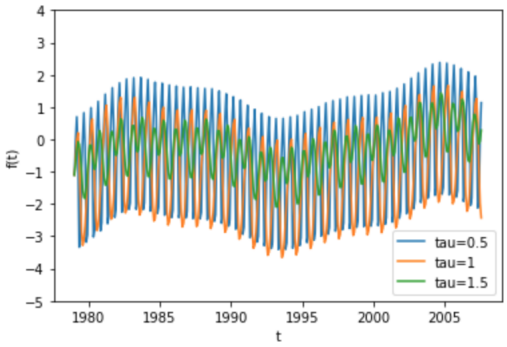
```
# Creates covariance kernel of t with given parameters
def kernel(T,theta,phi,tau,sigma,eta,zeta):
    mean=np.zeros((len(T)))
    K=np.zeros((len(T),len(T)))
    sind=0
    tind=0
    for s in T:
        tind=0
        for t in T:
            k=(theta**2)*np.exp(-1*(2*np.sin(np.pi*(s-t)/tau)**2)/sigma**2)
            k=k+(phi**2)*np.exp(-1*((s-t)**2)/(2*(eta**2)))
            k=k+(zeta**2)*(s==t)
            K[sind][tind]=k
            tind+=1
        sind+=1
    return K,mean

# Draws samples from Gaussian process given K and t
def samples(K,mean,t):
    # Epsilon*I added to kernel to stabilize it as PD matrix before taking
    # cholesky decomposition (epsilon=0.0001)
    L=LA.cholesky(K+(0.0001*np.identity(len(t))))
    f=mean+(L @ np.random.normal(0,1,len(t)))
    return f

K,mean=kernel(t,theta,phi,tau,sigma,eta,zeta)
f=samples(K,mean,t)
```

d. The random seed is set in order to easily compare characteristics of the functions dependent on the parameters. Notably, the kernel function is comprised of the periodic kernel function, the squared exponential kernel function, and noise. This can be observed in the plots below as they both display short periodic waves (periodic function) and large overall curves (squared exponential function). Parameter settings thus influence different aspects of the waves and curve. Initial parameters are chosen as $\theta = 3.5, \tau = 1, \sigma = 2.5, \phi = 0.5, \eta = 2.6$, and $\zeta = 0$.

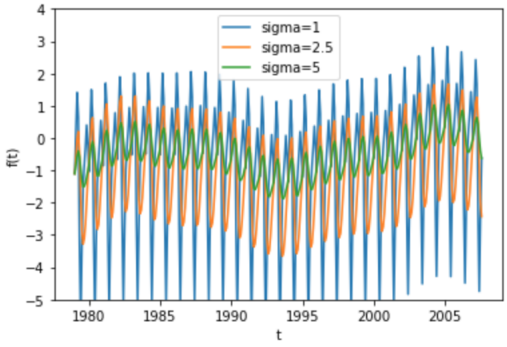


The value of $f(t)$ is plotted above for varying values of $\theta = 1.5, 3.5, 5.5$. θ is the signal variance, a scaling factor determining the variation of the periodic kernel function from its mean, specifically by

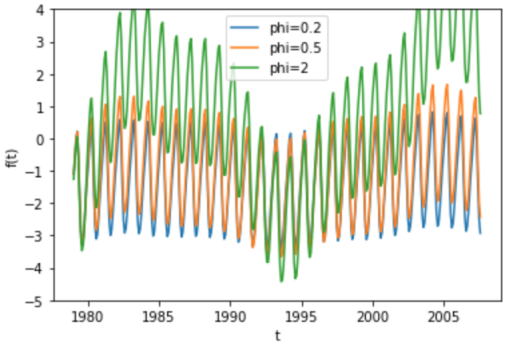
modifying the amplitude of the periodic waves. A higher value of θ yields a larger amplitude and a higher variance from the mean.



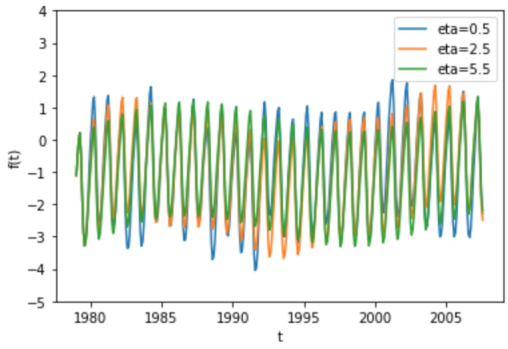
The value of $f(t)$ is plotted above for varying values of $\tau = 0.5, 1, 1.5$. τ determines the distance of the period of the wave function. A higher value of τ yields a longer period and a shorter amplitude of waves.



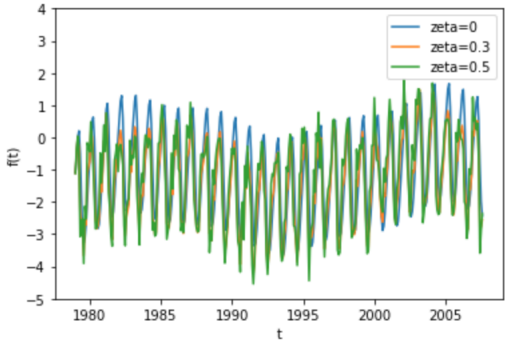
The value of $f(t)$ is plotted above for varying values of $\sigma = 1, 2.5, 5$. σ determines the lengthscale of waves and thus controls the smoothness of waves and how quickly they change values in the periodic kernel function. A higher value of σ yields smoother waves (less fluctuations) with slower changes in function values.



The value of $f(t)$ is plotted above for varying values of $\phi = 0.2, 0.5, 2$. ϕ is a scaling factor determining the variation of the function from its mean. A higher value of ϕ yields greater variation from the mean and a more pronounced shape of the squared exponential curve.

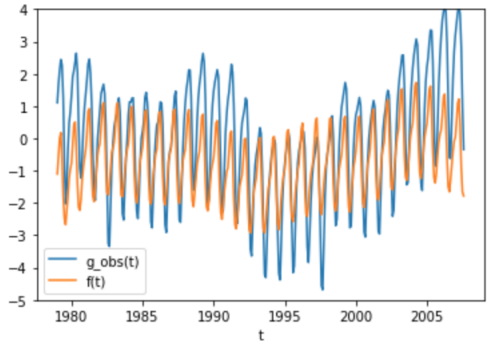


The value of $f(t)$ is plotted above for varying values of $\eta = 0.5, 2.5, 5.5$. η determines the lengthscale of the squared exponential and thus controls the smoothness of the squared exponential curve. A higher value of η yields smoother curves with slower changes in function values.

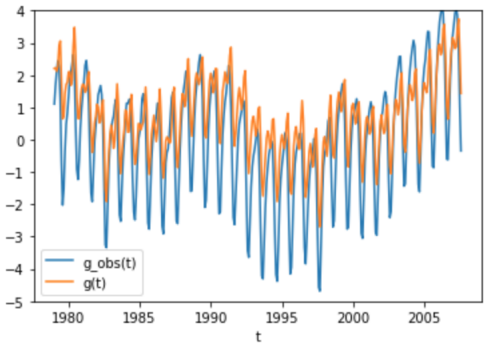


The value of $f(t)$ is plotted above for varying values of $\zeta = 0, 0.3, 0.5$. ζ determines how much noise appears in $f(t)$. A higher values of ζ yields a plot with more noise.

e. Based on the observations in (d), suitable hyperparameters for the kernel would be: $\theta = 3.5, \tau = 1, \sigma = 2.5, \phi = 0.5, \eta = 2.6$, and $\zeta = 0.01$. Modelling the computed $f(t)$ with the covariance kernel and the chosen hyperparameters and comparing to $g_{obs}(t)$,



The two plots overlap relatively well, as $f(t)$ models $g_{obs}(t)$'s general shape and values. Using the zero mean GP with the covariance kernel to model the residual function $g_{obs}(t)$ and find the new residual $g(t) = Y - (f(t) + \bar{\boldsymbol{w}}T)$, compared to $g_{obs}(t)$,



As displayed in the above figure, the variance of the new residual function $g(t)$ with the given parameters is much smaller (0.968) than $g_{obs}(t)$ (3.304) and much closer to $\epsilon(t) = 1$. Thus, it manages to model the residual function well, outside of expected additional noise.

f. The CO_2 concentration levels are extrapolated to 2020 using the Gaussian process with covariance kernel K over the observed and future values of t and the chosen parameter values from (e). The values of t are computed from 1979 to 2020, $t = \begin{bmatrix} t_1 & \dots & t_{344} & t'_{345} & \dots & t'_{493} \end{bmatrix}$. The covariance kernel K' is then computed over t from 1979-2020 and partitioned by the covariances between values of t that have been observed (A) and values of t' that are being predicted (B),
$K' = \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix}$

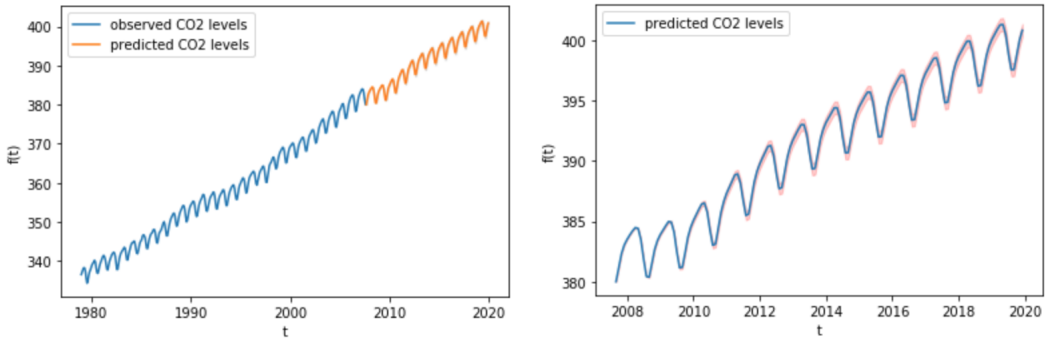
The predictive mean and variance of the residual $g(t')$ can then be computed using the partitions,
$\boldsymbol{\mu}_{pred} = K_{BA}K_{AA}^{-1}g_{obs}(t)$
$\Sigma_{pred} = K_{BB} - K_{BA}K_{AA}^{-1}K_{AB}$

Samples can be drawn from the Gaussian process using the predictive mean and covariance, which are then summed with the product of the posterior mean and T' values being predicted,
$T' = \begin{bmatrix} t'_{345} & \dots & t'_{493} \\ 1 & \dots & 1 \end{bmatrix}$
$g(t') \sim GP(\mu_{pred}, \Sigma_{pred})$
$f_{pred}(\boldsymbol{t'}) = a_{MAP}\boldsymbol{t'} + b_{MAP} + g(\boldsymbol{t'}) = \bar{\boldsymbol{w}}T' + g(\boldsymbol{t'})$

```
# Creates t and T
tpred=np.zeros(492)
i=0
for y in range(1979,2020):
    for m in range(1,13):
        tpred[i]=y+(m-1)/12
        i+=1
Tpred=np.ones((2,len(tpred)))
Tpred[0]=tpred

theta= 3.5
phi= 0.5
tau=1
sigma=2.5
eta=2.6
zeta=0.01

# Calculates kernel and predicted mean and covariance
```

```
K,mean=kernel(tpred,theta,phi,tau,sigma,eta,zeta)
gobs=Y-(postmean @ T)
meanpred=K[344:492,0:344] @ (LA.inv(K[0:344,0:344]) @ gobs)
covpred=K[344:492,344:492]-K[344:492,0:344] @ (LA.inv(K[0:344,0:344]) @ K[0:344,344:492])

# Samples g and calculates f(t) and standard deviation
g=samples(covpred,meanpred,tpred[344:492])
fpred=g+(postmean @ Tpred[:,344:492])
stdev=np.sqrt(np.abs(np.diag(covpred)))
```

Plotting the extrapolated CO_2 concentration levels along with the observed CO_2 levels,



The behavior of the extrapolation conforms to the expectations as it follows a similar pattern in its periodicity, curve slope, and curve shape. As $g(t')$ sampled from the Gaussian process moves further from the observed values of t, the standard deviation increases. The conclusions are not very sensitive to the settings of the kernel hyperparameters because the conclusions are stabilized by the product of the posterior means and t. Hyperparameter choice gives fluctuations in the predicted residual of the posterior mean and t', but ultimately the conclusion remains relatively steady with slight differences in curvature and period.

g. The above procedure is not fully Bayesian because it attempts to find a best fit $f(t)$ of Y through modifying the hyperparameters directly from the data. There is no prior for the hyperparameters and the hyperparameters are manually chosen based on observation rather than evidence optimization. $f(t)$ could be modelled in a Bayesian framework by using evidence optimization by gradient ascent in $\log P(Y|t,\theta,\tau,\sigma,\phi,\eta,\zeta)$.