

SDS 383D, Exercises 2: Bayes and the Gaussian Linear Model

Jan-Michael Cabrera

February 25, 2019

A simple Gaussian location model

(A) Show that the marginal prior, $p(\theta)$ takes on the following form:

$$p(\theta) \propto \left(1 + \frac{1}{\nu} \frac{(x - m)^2}{s^2}\right)^{-\frac{\nu+1}{2}}$$

The joint prior for θ and ω can be expressed as

$$p(\theta, \omega) \propto \omega^{(d+1)/2-1} \exp\left(-\omega \frac{k(\theta - \mu)^2}{2}\right) \exp\left(-\omega \frac{\eta}{2}\right)$$

This joint normal-gamma prior can be marginalized over ω to obtain a distribution for θ . Combining terms in the exponents and letting $a = \frac{d+1}{2}$ and $b = \frac{k(\theta - \mu)^2}{2} + \frac{\eta}{2}$, we get

$$\begin{aligned} p(\theta) &\propto \int p(\theta, \omega) d\omega \\ &\propto \int \omega^{a-1} \exp(-\omega b) d\omega \end{aligned}$$

We immediately realize that the term to the right of the integral is the unnormalized form of a gamma distribution. We add these normalizing components to ensure the intergral integrates to one and reduce the result,

$$\begin{aligned} p(\theta) &\propto \frac{\Gamma(a)}{b^a} \int \frac{b^a}{\Gamma(a)} \omega^{a-1} \exp(-\omega b) d\omega \\ &\propto \Gamma\left(\frac{d+1}{2}\right) \left[\frac{\eta}{2} + \frac{k(\theta - \mu)^2}{2}\right]^{-\frac{d+1}{2}} \\ &\propto \left[\frac{\eta}{2} \left(1 + \frac{kd(\theta - \mu)^2}{2d\eta/2}\right)\right]^{-\frac{d+1}{2}} \\ &\propto \left(1 + \frac{1}{d} \frac{(\theta - \mu)^2}{(d\eta/k)}\right)^{-\frac{d+1}{2}}. \end{aligned}$$

The result can be recognized as having the form of the first equation above with the following variable relations.

$$\begin{aligned}
\nu &= d \\
x &= \theta \\
s^2 &= \frac{d\eta}{k} \\
m &= \mu
\end{aligned}$$

(B) Show that $p(\theta, \omega | \mathbf{y})$ has the form

$$p(\theta, \omega | \mathbf{y}) \propto \omega^{(d^*+1)/2-1} \exp \left[-\omega \frac{k^*(\theta - \mu^*)^2}{2} \right] \exp \left[-\omega \frac{\eta^*}{2} \right]$$

The sampling distribution is given as the following, with the term in the exponent reduced to sufficient statistics for a normal distribution,

$$p(\mathbf{y} | \theta, \omega) \propto \omega^{n/2} \exp \left[-\omega \left(\frac{S_y + n(\bar{y} - \theta)^2}{2} \right) \right]$$

We multiply our normal-gamma prior with the normal sampling distribution and collect similar terms in the exponents.

$$\begin{aligned}
p(\theta, \omega | \mathbf{y}) &\propto \omega^{(d+1)/2-1} \exp \left[-\omega \frac{(\theta - \mu)^2}{2} \right] \exp \left[-\omega \frac{\eta}{2} \right] \omega^{n/2} \exp \left[-\omega \left(\frac{S_y + n(\bar{y} - \theta)^2}{2} \right) \right] \\
&\propto \omega^{(d+n+1)/2-1} \exp \left[-\frac{\omega}{2} (k(\theta - \mu)^2 + \eta + S_y + n(\bar{y} - \theta)^2) \right]
\end{aligned}$$

We expand the term in the $e^{(\dots)}$ term and collect terms together,

$$\begin{aligned}
k(\theta - \mu)^2 + \eta + S_y + n(\bar{y} - \theta)^2 &= k\theta^2 - 2k\mu\theta + k\mu^2 + \eta + S_y + n\bar{y}^2 - 2n\theta\bar{y} + n\theta^2 \\
&= (k+n)\theta^2 + (-2k\mu - 2n\bar{y})\theta + k\mu^2 + \eta + S_y + n\bar{y}^2.
\end{aligned}$$

The result from expanding the terms can be reduced to the following form to recover the normal part of the posterior for θ ,

$$ax^2 + bx + c = a(x - h)^2 + l, \quad h = -\frac{b}{2a}, \quad l = c - ah^2$$

We carry out the process of completing the square the θ term and reduce other parts of the expression,

$$\begin{aligned}
(k+n)\theta^2 + (-2k\mu - 2n\bar{y})\theta + k\mu^2 + \eta + S_y + n\bar{y}^2 &= \\
(k+n) \left(\theta - \frac{k\mu + n\bar{y}}{k+n} \right)^2 + k\mu^2 + \eta + S_y + n\bar{y}^2 - (k+n) \left(\frac{k\mu + n\bar{y}}{k+n} \right)^2 &= \\
(k+n) \left(\theta - \frac{k\mu + n\bar{y}}{k+n} \right)^2 + \frac{nk(\mu - \bar{y})^2}{k+n} + \eta + S_y. &
\end{aligned}$$

The result is plugged back into the exponential term, and the terms are collected to obtain the form of a normal-gamma posterior.

$$\begin{aligned}
p(\mathbf{y}|\theta, \omega) &\propto \omega^{(d+n+1)/2-1} \exp \left[-\frac{\omega}{2} \left((k+n) \left(\theta - \frac{k\mu + n\bar{y}}{k+n} \right)^2 + \frac{nk(\mu - \bar{y})^2}{k+n} + \eta + S_y \right) \right] \\
&\propto \omega^{(d+n+1)/2-1} \exp \left[-\frac{\omega}{2} \left((k+n) \left(\theta - \frac{k\mu + n\bar{y}}{k+n} \right)^2 \right) \right] \exp \left[-\frac{\omega}{2} \left(\frac{nk(\mu - \bar{y})^2}{k+n} + \eta + S_y \right) \right]
\end{aligned}$$

The normal-gamma posterior has the form of the normal-gamma prior with the following variable relations.

$$\begin{aligned}
d^* &= d + n \\
k^* &= k + n \\
\mu^* &= \frac{k\mu + n\bar{y}}{k+n} \\
\eta^* &= \frac{nk(\mu - \bar{y})^2}{k+n} + \eta + S_y
\end{aligned}$$

(C) From the joint posterior, what is the conditional posterior distribution $p(\theta|\mathbf{y}, \omega)$?

$$\begin{aligned}
p(\theta|\mathbf{y}, \omega) &= \int_0^\infty p(\theta, \omega|\mathbf{y}) d\omega \\
&\propto \exp \left[-\frac{\omega}{2} \left((k+n) \left(\theta - \frac{k\mu + n\bar{y}}{k+n} \right)^2 \right) \right] \int_0^\infty \omega^{(d+n+1)/2-1} \exp \left[-\frac{\omega}{2} \left(\frac{nk(\mu - \bar{y})^2}{k+n} + \eta + S_y \right) \right] d\omega
\end{aligned}$$

The conditional posterior distribution is simply the joint posterior marginalized over ω . We also can recognize that the normal distribution can be completely characterized by its mean and variance. Therefore,

$$(\theta|\mathbf{y}, \omega) \sim \mathcal{N} \left(\frac{k\mu + n\bar{y}}{k+n}, (\omega(k+n))^{-1} \right)$$

(D) From the joint posterior, what is the marginal posterior distribution $p(\omega|\mathbf{y})$?

$$\begin{aligned}
p(\omega|\mathbf{y}) &= \int_0^\infty p(\theta, \omega|\mathbf{y}) d\theta \\
&\propto \omega^{(d+n)/2-1} \exp \left[-\frac{\omega}{2} \left(\frac{nk(\mu - \bar{y})^2}{k+n} + \eta + S_y \right) \right] \int_{-\infty}^\infty \omega^{1/2} \exp \left[-\frac{\omega}{2} \left((k+n) \left(\theta - \frac{k\mu + n\bar{y}}{k+n} \right)^2 \right) \right] d\theta
\end{aligned}$$

The marginal posterior for ω is found in a similar fashion as above, noting that part of the normal distribution to be marginalized depends on ω . Including this term in the integral ensures that it integrates to a constant not dependent on ω . The distribution then has the following form,

$$\begin{aligned}
\omega|\mathbf{y} &\sim \text{Gamma} \left(\frac{d^*}{2}, \frac{\eta^*}{2} \right) \\
d^* &= d + n \\
\eta^* &= \frac{nk(\mu - \bar{y})^2}{k+n} + \eta + S_y
\end{aligned}$$

- (E) Show that the marginal posterior $p(\theta|\mathbf{y})$ takes the form of a centered, scaled t distribution and express the parameters in terms of the four parameters of the normal-gamma posterior for (θ, ω) .

$$p(\theta) \propto \left(1 + \frac{1}{d} \frac{(\theta - \mu)^2}{(d\eta/k)}\right)^{-\frac{d+1}{2}}$$

This simply requires plugging in the respective starred values into the result above producing,

$$p(\theta|\mathbf{y}) \propto \left(1 + \frac{1}{d^*} \frac{(\theta - \mu^*)^2}{(d^*\eta^*/k^*)}\right)^{-\frac{d^*+1}{2}}$$

$$d^* = d + n$$

$$k^* = k + n$$

$$\mu^* = \frac{k\mu + n\bar{y}}{k + n}$$

$$\eta^* = \frac{nk(\mu - \bar{y})^2}{k + n} + \eta + S_y$$

- (F) True or false: in the limit as the prior parameters k , d , and η approach zero, the priors $p(\theta)$ and $p(\omega)$ are valid probability distributions.

To check this result we simply check to see if the resultant distribution integrates to one or a constant in the limit. The prior for θ is,

$$p(\theta) \propto \left[1 + \frac{1}{d} \frac{(\theta - \mu)^2}{\eta/kd}\right]^{-\frac{d+1}{2}}$$

$$\lim_{k,d,\eta \rightarrow 0} \left[1 + \frac{1}{d} \frac{(\theta - \mu)^2}{\eta/kd}\right]^{-\frac{d+1}{2}} = \lim_{k,d,\eta \rightarrow 0} \left[1 + \frac{k(\theta - \mu)^2}{\eta}\right]^{-\frac{d+1}{2}}$$

$$\left[1 + \frac{0(\theta - \mu)^2}{0}\right]^{-\frac{1}{2}}$$

In the limit as the parameters approach zero, we have an undefined function that cannot be integrated. For ω we have,

$$p(\omega) \propto \omega^{d/2-1} \exp\left(-\omega \frac{\eta}{2}\right)$$

Taking the limit for this expression,

$$\lim_{d,\eta \rightarrow 0} \omega^{d/2-1} \exp\left(-\omega \frac{\eta}{2}\right) = \omega^{-1}$$

The resultant expression has a singularity at $\omega = 0$ and will therefore integrate to infinity.

False

- (G) True or false: in the limit as the prior parameters k , d , and η approach zero, the posteriors $p(\theta|\mathbf{y})$ and $p(\omega|\mathbf{y})$ are valid probability distributions.

We first take the limit in the starred expressions,

$$\begin{aligned}
d^* = n + d &\xrightarrow{d \rightarrow 0} d^* = n \\
k^* = k + n &\xrightarrow{k \rightarrow 0} k^* = n \\
\mu^* = \frac{k\mu + n\bar{y}}{k + n} &\xrightarrow{k \rightarrow 0} \mu^* = \bar{y} \\
\eta^* = \frac{nk(\mu - \bar{y})^2}{k + n} + \eta + S_y &\xrightarrow{k, \eta \rightarrow 0} \eta^* = S_y
\end{aligned}$$

Plugging in these values into our posteriors we get,

$$\begin{aligned}
p(\theta|\mathbf{y}) &\propto \left[1 + \frac{1}{n} \frac{(\theta - \bar{y})^2}{S_y/n^2} \right]^{-\frac{n+1}{2}} \\
p(\omega|\mathbf{y}) &\propto \omega^{n/2-1} \exp\left(-\omega \frac{S_y}{2}\right)
\end{aligned}$$

Both are kernels for valid probability distributions

True

- (H) True or false: In the limit as the prior parameters k , d , and η approach zero, the Bayesian credible interval for θ becomes identical to the classical (frequentist) confidence interval for θ at the same confidence level.

$$\theta \in m \pm t^* \cdot s$$

$$\begin{aligned}
m = \mu^* &\rightarrow \bar{y} \quad \text{for } k, d, \eta \rightarrow 0 \\
s^{*2} = \frac{S_y}{n^2} &\rightarrow s = \frac{1}{n} \left[\sum_{i=1}^n (y_i - \bar{y})^2 \right]^{1/2} \\
\theta &\in \bar{y} \pm t^* \cdot \frac{1}{n} \left[\sum_{i=1}^n (y_i - \bar{y})^2 \right]^{1/2}
\end{aligned}$$

True

The Conjugate Gaussian Linear Model

Basics

(A) Derive the conditional posterior $p(\beta|\mathbf{y}, \omega)$

$$\begin{aligned}\beta|\omega &\sim \mathcal{N}(m, (\omega K)^{-1}) \\ \omega &\sim \text{Gamma}\left(\frac{d}{2}, \frac{\eta}{2}\right) \\ \mathbf{y}|\beta, \omega &\sim \mathcal{N}(X\beta, (\omega\Lambda)^{-1})\end{aligned}$$

First we find the joint posterior for β and ω given the multivariate sampling distribution.

$$\begin{aligned}p(\beta, \omega|\mathbf{y}) &\propto p(\omega)p(\beta|\omega)p(\mathbf{y}|\beta, \omega) \\ &\propto \omega^{d/2-1} \exp\left(-\omega \frac{\eta}{2}\right) \omega^{p/2} \exp\left[-\frac{\omega}{2}(\beta - m)^T K(\beta - m)\right] \omega^{n/2} \exp\left[-\frac{\omega}{2}(y - X\beta)^T \Lambda(y - X\beta)\right] \\ &\propto \omega^{\frac{d+p+n}{2}-1} \exp\left[-\frac{\omega}{2}\left((\beta - m)^T K(\beta - m) + (y - X\beta)^T \Lambda(y - X\beta) + \eta\right)\right]\end{aligned}$$

After combining exponents, we distribute the terms and collect the β terms together,

$$\begin{aligned}&(\beta - m)^T K(\beta - m) + (y - X\beta)^T \Lambda(y - X\beta) + \eta \\ &= \beta^T K\beta - \beta^T Km - m^T K\beta + m^T Km + y^T \Lambda y - y^T \Lambda X\beta - \beta^T X^T \Lambda y + \beta^T X^T \Lambda X\beta + \eta \\ &= \beta^T K\beta - 2\beta^T Km + m^T Km + y^T \Lambda y - 2\beta^T X^T \Lambda y + \beta^T X^T \Lambda X\beta + \eta \\ &= \beta^T (K + X^T \Lambda X)\beta - 2\beta^T (Km + X^T \Lambda y) + m^T Km + y^T \Lambda y + \eta\end{aligned}$$

As before, we can realize we need to get the expression in the proper form that characterizes the multivariate normal distribution. We do this by completing the square. The equation has the form $Q^T A Q + Q^T b + c = (Q - h)^T A(Q - h) + k$, with $h = -\frac{1}{2}A^{-1}b$ and $k = c - \frac{1}{4}b^T A^{-1}b$.

Let $A = (K + X^T \Lambda X) = K^*$, $b = -2(Km + X^T \Lambda y)$, and $c = m^T Km + y^T \Lambda y + \eta$. The term h reduces to the following,

$$\begin{aligned}h &= -\frac{1}{2}(K + X^T \Lambda X)^{-1} \cdot (-2(Km + X^T \Lambda y)) \\ &= (K + X^T \Lambda X)^{-1}(Km + X^T \Lambda y) = m^*\end{aligned}$$

Plugging these terms back into the equation we get,

$$\begin{aligned}&\beta^T (K + X^T \Lambda X)\beta - 2\beta^T (Km + X^T \Lambda y) + m^T Km + y^T \Lambda y + \eta \\ &= (\beta - m^*)^T K^*(\beta - m^*) + m^T Km + y^T \Lambda y + \eta - (Km + X^T \Lambda y)^T K^{*-1}(Km + X^T \Lambda y)\end{aligned}$$

These are then reintroduced into the posterior,

$$\begin{aligned}p(\beta, \omega|\mathbf{y}) &\propto \omega^{\frac{d+p+n}{2}-1} \exp\left[-\frac{\omega}{2}\left((\beta - m^*)^T K^*(\beta - m^*) + m^T Km + y^T \Lambda y + \eta - (Km + X^T \Lambda y)^T K^{*-1}(Km + X^T \Lambda y)\right)\right] \\ &\propto \omega^{\frac{d^*}{2}-1} \exp\left[-\omega \frac{\eta^*}{2}\right] \omega^{\frac{p}{2}} \exp\left[-\frac{\omega}{2}(\beta - m^*)^T K^*(\beta - m^*)\right]\end{aligned}$$

We see that the form of the posterior is also multivariate normal-gamma with the following variables

$$\begin{aligned}
d^* &= d + n \\
K^* &= (K + X^T \Lambda X) \\
\eta^* &= m^T K m + y^T \Lambda y + \eta - (K m + X^T \Lambda y)^T K^{*-1} (K m + X^T \Lambda y) \\
m^* &= (K + X^T \Lambda X)^{-1} (K m + X^T \Lambda y)
\end{aligned}$$

From this distribution, again since the multivariate normal is completely characterized by it's mean vector and covariance matrix, we get that the conditional posterior has the form,

$$\beta | \omega, \mathbf{y} \sim \mathcal{N}(m^*, (\omega K^*)^{-1}).$$

(B) Derive the marginal posterior $p(\omega | \mathbf{y})$

The marginal posterior for ω can be found by marginalizing the joint posterior over β ,

$$\begin{aligned}
p(\omega | \mathbf{y}) &\propto \int_{-\infty}^{\infty} p(\beta, \omega | \mathbf{y}) d\beta \\
&\propto \omega^{\frac{d^*}{2}-1} \exp \left[-\omega \frac{\eta^*}{2} \right] \int_{-\infty}^{\infty} \omega^{\frac{p}{2}} \exp \left[-\frac{\omega}{2} (\beta - m^*)^T K^* (\beta - m^*) \right] d\beta \\
&\propto \omega^{\frac{d^*}{2}-1} \exp \left[-\omega \frac{\eta^*}{2} \right].
\end{aligned}$$

Note that again here we must be careful of the terms included within the integral when marginalizing. From this we find that the posterior also has the form of a gamma distribution,

$$\omega | \mathbf{y} \sim \text{Gamma} \left(\frac{d^*}{2}, \frac{\eta^*}{2} \right).$$

This distribution is characterized by the following

$$\begin{aligned}
d^* &= d + n \\
K^* &= (K + X^T \Lambda X) \\
\eta^* &= m^T K m + y^T \Lambda y + \eta - (K m + X^T \Lambda y)^T K^{*-1} (K m + X^T \Lambda y)
\end{aligned}$$

(C) Putting these together, derive the marginal posterior $p(\beta | \mathbf{y})$

The process here closely matches that of part A in the previous section. We marginalize the joint posterior over ω .

$$\begin{aligned}
p(\beta | \mathbf{y}) &\propto \int_0^{\infty} p(\beta, \omega | \mathbf{y}) d\omega \\
&\propto \int_0^{\infty} \omega^{\frac{d^*}{2}-1} \exp \left[-\omega \frac{\eta^*}{2} \right] \omega^{\frac{p}{2}} \exp \left[-\frac{\omega}{2} (\beta - m^*)^T K^* (\beta - m^*) \right] d\omega \\
&\propto \int_0^{\infty} \omega^a \exp(-b\omega) d\omega
\end{aligned}$$

Letting $a = \frac{d^*+p}{2}$ and $b = \frac{1}{2} (\eta^* + (\beta - m^*)^T K^* (\beta - m^*))$ we see that the terms in the integral form an unnormalized gamma distribution. Normalizing the distribution we obtain,

$$\begin{aligned} p(\beta|\mathbf{y}) &\propto \frac{\Gamma(a)}{b^a} \int_0^\infty \frac{b^a}{\Gamma(a)} \omega^a \exp(-b\omega) d\omega \\ &\propto \Gamma(a) b^{-a} \\ &\propto [\eta^* + (\beta - m^*)^T K^* (\beta - m^*)]^{-\frac{d^*+p}{2}} \\ &\propto \left[1 + \frac{1}{d^*} \frac{(\beta - m^*)^T K^* (\beta - m^*)}{\eta^*/d^*} \right]^{-\frac{d^*+p}{2}}. \end{aligned}$$

This has the form of a multivariate t distribution.

(D) Bayesian linear model fit to data in "gdpgrowth.csv"

Important libraries for the model are first imported.

```
from __future__ import division
import numpy as np
from numpy.linalg import inv
import pandas as pd
import matplotlib.pyplot as plt
import sys
sys.path.append('../scripts/')
from linear_models import linear_model
```

The data is read into the program and variables are assigned accordingly.

```
data = pd.read_csv('../data/gdpgrowth.csv', delimiter=',') # Import data
Y = data['GR6096'] # Y
X = data['DEF60'] # X without intercept feature
```

The feature matrix is prepended with a column of ones for later determining the intercept. The precision matrix is also constructed and weighted.

```
intercept = np.ones(len(X)) # create intercept feature column
X = np.transpose(np.array((intercept, X))) # build matrix from intercept and X features

k = np.ones(X.shape[1])*0.1
K = np.diag(k) # K, precision matrix for multivariate normal prior on beta
```

The feature matrix, response vector, and prior precision matrix are passed to an informed Bayesian Linear Model which calculates the posterior mean.

```
#### Informed Bayesian Model
informed = linear_model(X, Y, K) # Pass feature matrix, response vector, and precision matrix ↔
to create linear model object
m_star_1 = informed.bayesian() # Calculate linear model intercept and slope using the bayesian↔
method
```

The feature matrix and response vector are also passed to an uninformed Bayesian Linear Model (precisions of $O(10^{-4})$) and to a least squares method.

```
#### Uninformed Bayesian Model
uninformed = linear_model(X, Y) # Pass feature matrix and response vector to create ↔
linear_model object
m_star_2 = uninformed.bayesian() # Calculate linear model intercept and slope using the ↔
bayesian method
```



```
b0, b1 = uninformed.frequentist() # Calculate linear model intercept and slope using an ←
ordinary least squares method
```

Model responses are calculated from the model outputs and plotted.

```
x = np.linspace(X[:,1].min(), X[:,1].max()) # vector for plotting purposes

#### Model responses
y1 = m_star_1[0] + x * m_star_1[1] # Informed Bayesian Linear Model
y2 = m_star_2[0] + x * m_star_2[1] # Uninformed Bayesian Linear Model
y3 = b0 + x * b1 # Ordinary Least Squared Linear Model

#### Plot models
plt.figure()
plt.plot(X[:,1], Y, '.k', label='GDP Growth Rate Vs. Defense Spending')
plt.plot(x, y1, '-b', label='Informed Bayesian Linear Model')
plt.plot(x, y2, '-r', label='Uninformed Bayesian Linear Model')
plt.plot(x[0:-1:2], y3[0:-1:2], '*g', label='Ordinary Least Squares Linear Model')
plt.xlabel('Defense Spending')
plt.ylabel('GDP Growth Rate')
plt.legend(loc=0)
plt.savefig('figures/bayesian_linear_model.pdf')
```

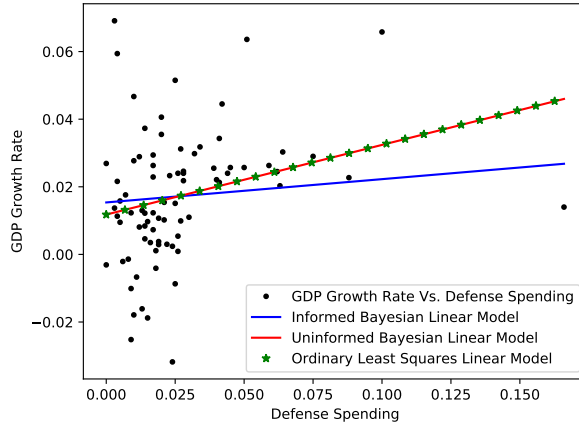


Figure 1: Linear models for GDP Growth Rate vs Defense Spending

A heavy-tailed error model

$$\begin{aligned}
\mathbf{y} | \beta, \omega, \Lambda &\sim N(X\beta, (\omega\Lambda)^{-1}) \\
\Lambda &= \text{diag}(\lambda_1, \dots, \lambda_n) \\
\lambda &\stackrel{iid}{\sim} \text{Gamma}(h/2, h/2) \\
\beta | \omega &\sim N(m, (\omega K)^{-1}) \\
\omega &\sim \text{Gamma}(d/2, \eta/2)
\end{aligned}$$

(A) Under this model, what is the implied conditional distribution $p(y_i | X, \beta, \omega)$?

$$\begin{aligned}
p(y_i|\beta, \omega) &= \int p(\lambda_i)p(\omega)p(\beta|\omega)p(y_i|\beta, \omega, \lambda_i)d\lambda_i \\
&\propto \int \lambda_i^{h/2-1} \exp\left(-\frac{h}{2}\lambda_i\right) (\omega\lambda_i)^{1/2} \exp\left[-\frac{1}{2}(\omega\lambda_i)(y_i - x_i^T\beta)^2\right] d\lambda_i \\
&\propto \int \omega^{1/2} \lambda_i^{\frac{h+1}{2}-1} \exp\left(-\frac{\lambda_i}{2} [h + \omega(y_i - x_i^T\beta)^2]\right) d\lambda_i
\end{aligned}$$

let $a = \frac{h+1}{2}$ and $b = \frac{1}{2} [h + \omega(y_i - x_i^T\beta)^2]$

$$\begin{aligned}
p(y_i|\beta, \omega) &\propto \int \lambda_i^{a-1} \exp(-b\lambda_i) d\lambda_i \\
&\propto b^{-a} \\
&\propto [h + \omega(y_i - x_i^T\beta)^2]^{-\frac{h+1}{2}} \\
&\propto \left[1 + \frac{\omega(y_i - x_i^T\beta)^2}{h}\right]^{-\frac{h+1}{2}}
\end{aligned}$$

(B) What is the conditional posterior distribution $p(\lambda_i|\mathbf{y}, \beta, \omega)$?

$$\begin{aligned}
p(\lambda_i|y_i, \beta, \omega) &\propto p(\lambda_i)p(y_i|\beta, \omega, \lambda_i) \\
&\propto \lambda_i^{h/2-1} \exp\left(-\frac{h}{2}\lambda_i\right) (\omega\lambda_i)^{1/2} \exp\left[-\frac{1}{2}(\omega\lambda_i)(y_i - x_i^T\beta)^2\right] \\
&\propto \lambda_i^{\frac{h+1}{2}-1} \exp\left(-\frac{\lambda_i}{2} [h + \omega(y_i - x_i^T\beta)^2]\right) \\
\lambda_i|y_i, \beta, \omega &\sim \text{Gamma}\left(\frac{h+1}{2}, \frac{h + \omega(y_i - x_i^T\beta)^2}{2}\right)
\end{aligned}$$

(C) Code up a Gibbs sampler that repeatedly cycles through sampling the following three sets of conditional distributions.

- $p(\beta|\mathbf{y}, \omega, \Lambda)$
- $p(\omega|\mathbf{y}, \Lambda)$
- $p(\lambda_i|\mathbf{y}, \beta, \omega)$

Imports necessary libraries for running models.

```

from __future__ import division
import numpy as np
from numpy.linalg import inv
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
import sys
sys.path.append('../scripts/')
from linear_models import linear_model
from samplers import Gibbs, Trace

```

Read data and assign to necessary vectors as well as prepend the feature matrix with a column of ones.

```
data = pd.read_csv('../data/gdpgrowth.csv', delimiter=',')

Y = data['GR6096'] # X without intercept feature
X = data['DEF60'] # Y

intercept = np.ones(len(Y)) # create intercept feature column
X = np.transpose(np.array((intercept, X))) # build matrix from intercept and X features
```

Pass feature matrix and response vector to Gibbs sampler and sample 5000 times. Runs the model, plots the omega trace and beta traces and returns the means from the beta trace.

```
#### Heavy Tailed Error Model
model = Gibbs(X, Y, samples=5000) # instantiate the model from the Gibbs class

beta_trace, omega_trace, Lambda_trace = model.heavy_tailed() # run the heavy tailed error ↵
model

omega_trace.plot('figures/') # plots omega trace
beta_trace.plot('figures/') # plots beta traces

b_0, b_1 = beta_trace.mean() # returns the means from each beta trace
```

Runs the linear model from the previous section for comparison to the Heavy Tailed Error Model.

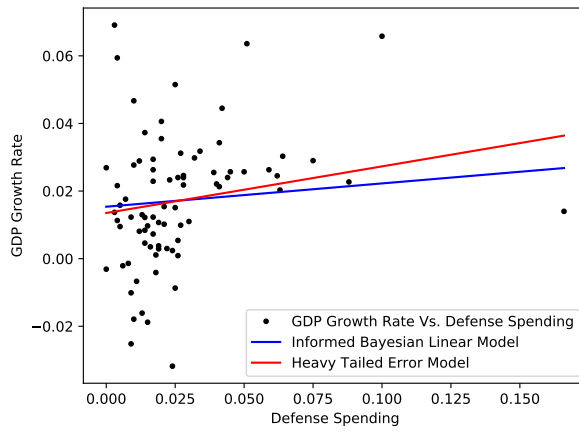
```
#### Informed Bayesian Model
k = np.ones(X.shape[1])*0.1
K = np.diag(k) # K, precision matrix for multivariate normal prior on beta

informed = linear_model(X, Y, K) # Pass feature matrix, response vector, and precision matrix ↵
to create linear model object
m_star_1 = informed.bayesian() # Calculate linear model intercept and slope using the bayesian↵
method
```

Plots model responses for comparison.

```
#### Model responses
x = np.linspace(X[:,1].min(), X[:,1].max()) # vector for plotting purposes
y1 = m_star_1[0] + x * m_star_1[1] # Informed Bayesian Linear Model
y2 = b_0 + x * b_1 # Heavy Tailed Error Model

plt.figure()
plt.plot(X[:,1], Y, '.k', label='GDP Growth Rate Vs. Defense Spending')
plt.plot(x, y1, '-b', label='Informed Bayesian Linear Model')
plt.plot(x, y2, '-r', label='Heavy Tailed Error Model')
plt.xlabel('Defense Spending')
plt.ylabel('GDP Growth Rate')
plt.legend(loc=0)
plt.savefig('figures/heavy_tailed.pdf')
```



A Bayesian Linear Model

```
from __future__ import division
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
from numpy.linalg import inv

class linear_model:
    """
    This class fits linear models using bayesian and frequentist methods to data sets
    """
    def __init__(self, X, Y, K=[]):
        """
        Parameters
        -----
            X: float
               feature matrix

            Y: float
               response vector

            K: float, optional
               precision matrix for multivariate normal prior on beta
               if not specified, sets a low precision, approximating the frequentist fit

        Raises
        -----
            ValueError
                If precision matrix shape does not match shape of feature matrix

        """
        self.X = X
        self.Y = Y
        self.K = np.array(K)

        if self.K.shape[0]==0:
            k = np.ones(self.X.shape[1])*0.0001
            self.K = np.diag(k)
        else:
            if self.K.shape[0] != self.X.shape[1]:
                raise ValueError('Precision matrix shape, %i does not match feature matrix shape, %i' % (self.K.shape[0], self.X.shape[1]))

    def bayesian(self):
        """
        Returns
        -----
            m_star: float
                Represents the slope and intercept coefficients to a linear model
                .. math:: m^* = (K + X^T \backslash \Lambda X)^{-1} (K m + X^T \backslash \Lambda Y)
                .. math:: y = m^*[0] + x m^*[1]

        """
        m = np.zeros(self.X.shape[1])

        Lambda = np.eye(len(self.X))

        m_star = inv(self.K + np.transpose(self.X) @ Lambda @ self.X) @ (self.K @ m + np.transpose(self.X) @ (Lambda @ self.Y))

        return m_star

    def frequentist(self):
        """
        Returns
        -----

```

```

        b_0: float
            intercept to an ordinary least squares linear model

        b_1: float
            slope to an ordinary least squares linear model

        .. math:: y = b_0 + x b_1

[1] Adapted from https://www.geeksforgeeks.org/linear-regression-python-implementation/

"""

n = np.size(self.X[:,1])

m_x, m_y = np.mean(self.X[:,1]), np.mean(self.Y)

SS_xy = np.sum(self.Y*self.X[:,1]) - n*m_y*m_x
SS_xx = np.sum(self.X[:,1]*self.X[:,1]) - n*m_x*m_x

b_1 = SS_xy / SS_xx
b_0 = m_y - b_1*m_x

return b_0, b_1

```

B Heavy Tailed Error Model

```

from __future__ import division
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
from numpy.linalg import inv

class Trace:
    """
    This class instantiates traces
    """
    def __init__(self, name, iterations, shape=1):
        """
        Parameters
        -----
        name: str
            name of the trace for file i/o purposes

        iterations: int
            number of iterations to be ran for a given trace

        shape: int
            sets size of trace, default=1

        """
        self.name = name
        self.trace = np.zeros((iterations, shape))

    def plot(self, figures_directory=''):
        """
        Parameters
        -----
        figures_directory: str
            sets the directory for which figures are to be saved in
            default behavior is to show the figure rather than saving

        """
        if len(figures_directory) == 0:
            plt.show()

```

```

else:
    for i in range(self.trace.shape[1]):
        plt.figure()
        plt.plot(self.trace[:,i], '-b', label=self.name+'_'+str(i))
        plt.savefig(figures_directory+self.name+'_'+str(i)+'_trace.png')

def update_trace(self, index, value):
    """
    Parameters
    -----
    index: int
        point in trace to update
    value: float
        the value passed to trace at index

    """
    self.trace[index] = value

def mean(self):
    """
    Returns
    -----
    mean: float
        calculates the mean for each column in a given trace

    """
    mean = np.zeros(self.trace.shape[1])
    for i in range(len(mean)):
        mean[i] = self.trace[:,i].mean()
    return mean

class Gibbs:
    """
    This class will use a Gibbs sampling methods to draw posterior distributions
    """
    def __init__(self, X, Y, samples=100):
        """
        Parameters
        -----
        X: float
            feature matrix

        Y: float
            response vector

        samples: int
            number of samples to be drawn from posteriors, default is 100 for debugging ←
            purposes

        """
        self.X = X
        self.Y = Y
        self.samples = samples

    def heavy_tailed(self):
        """
        Returns
        -----
        beta_trace: float
            samples from posterior of the form
            .. math:: \beta | \bar{y} \sim N(m^*, (\omega K^*)^{-1})

        omega_trace: float
            samples from the posterior of the form
            .. math:: \omega | \bar{y} \sim \text{Gamma}(d^*/2, \eta^*/2)

        Lambda_trace: float
            samples from the posterior of the form

```

```

        .. math:: \lambda_i \mid \bar{y}, \beta, \omega \sim \text{Gamma}(\frac{h+1}{2}, \frac{h + \omega(y_i + x_i^T \beta)^2}{2})
    """

    ##### Instantiate Traces
    iterations = self.samples
    beta_trace = Trace('beta', iterations = iterations, shape=self.X.shape[1])
    omega_trace = Trace('omega', iterations = iterations)
    Lambda_trace = Trace('Lambda', iterations=iterations, shape=len(self.Y))

    ##### Instantiate Prior values
    d = 1
    eta = 1

    n = self.X.shape[0]

    k = np.ones(self.X.shape[1])*0.1
    K = np.diag(k)

    h = 1
    omega = 1

    m = np.zeros(self.X.shape[1])

    lambda_diag = np.ones(self.X.shape[0])

    ##### Iteratively sample conditional distributions
    for i in range(iterations):

        Lambda = np.diag(lambda_diag)

        # m* = (K + X^T \Lambda X)^{-1} (K m + X^T \Lambda y)
        m_star = inv(K + np.transpose(self.X) @ Lambda @ self.X) @ (K @ m + np.transpose(self.X) @
            ) @ (Lambda @ self.Y))

        # K* = (K + X^T \Lambda X)
        K_star = K + np.transpose(self.X) @ Lambda @ self.X

        # d* = d + n
        d_star = d + n

        # \eta* = m^T K m + y^T \Lambda y + \eta - (K m + X^T \Lambda y)^T {K*}^{-1} (K m + X^T \Lambda y)
        eta_star = np.transpose(m) @ K @ m + np.transpose(self.Y) @ Lambda @ self.Y + eta + np.
            transpose(K @ m + np.transpose(self.X) @ (Lambda @ self.Y)) @ inv(K_star) @ (K @ m +
            + np.transpose(self.X) @ (Lambda @ self.Y))

        ##### Sample posterior values
        beta = stats.multivariate_normal.rvs(mean = m_star, cov=inv(omega* K_star))
        omega = stats.gamma.rvs(d_star/2, (2/eta_star))
        for j in range(len(lambda_diag)):
            lambda_diag[j] = stats.gamma.rvs((h+1)/2, (2/(h+omega*(self.Y[j] - np.transpose(
                self.X[j])@beta)**2)))

        ##### Update traces with posterior values
        beta_trace.update_trace(i, beta)
        omega_trace.update_trace(i, omega)
        Lambda_trace.update_trace(i, lambda_diag)

    return beta_trace, omega_trace, Lambda_trace

```