CS 698S – Homework Assignment 1

Jaivardhan Kapoor
Roll: 150300

Question 1

Given x is scalar r.v. , $x \sim N(x|0,\eta)$
and , $\eta \sim Exp(\eta|r^2/2)$; $r > 0$
where $Exp(x|\lambda) = \lambda \exp(-\lambda x)$

We have to derive marginal distribution of x, i.e;

$$p(x|r) = \int p(x|\eta)\, p(\eta|r)\, d\eta$$

Since this is a hard integral, we use moment generating function to calculate this.

$$p(x|r) = \int_{0}^{\infty} \frac{r^2}{2} \exp\left\{\frac{r^2\eta}{2}\right\} \cdot \frac{1}{\sqrt{2\pi\eta}} \exp\left\{\frac{-x^2}{2\eta}\right\} d\eta$$

[limit is from 0 to ∞ because η takes values >0]

$$\text{mgf}\,(p(x|r)) = \int e^{tx}\left(\int \frac{r^2}{2\sqrt{2\pi\eta}} \exp\left\{\frac{-x^2}{2\eta}\right\} \exp\left\{-\frac{r^2\eta}{2}\right\} d\eta\right) dx$$

$$= \int_{-\infty}^{\infty}\int_{0}^{\infty} \left(\frac{r^2}{2\sqrt{2\pi\eta}} \exp\left\{\frac{-x^2}{2\eta} + tx - \frac{r^2\eta}{2}\right\} d\eta\, dx.\right.$$

$$= \int_{0}^{\infty}\int_{-\infty}^{\infty} \frac{r^2}{2\sqrt{2\pi\eta}} \exp\left\{\frac{-x^2}{2\eta} + tx - \frac{r^2\eta}{2}\right\} dx\, d\eta$$

$$= \int_{0}^{\infty}\int_{-\infty}^{\infty} \frac{r^2}{2\sqrt{2\pi\eta}} \exp\left\{\frac{-x^2 + 2tx\eta - r^2\eta^2}{2\eta}\right\} dx\, d\eta$$

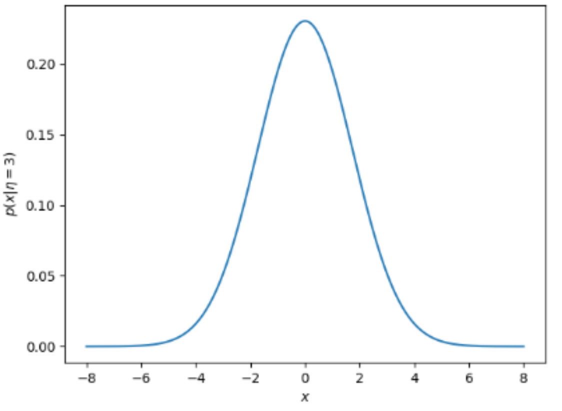
$$= \int_{0}^{\infty}\int_{-\infty}^{\infty} \underbrace{\frac{r^2}{2\sqrt{2\pi\eta}} \exp\left\{\frac{-(x-t\eta)^2}{2\eta}\right\}}_{\text{this gets integrated to } 1} \exp\left\{\frac{(r^2-t^2)\eta^2}{2\eta}\right\} dx\, d\eta$$

$$= \frac{r^2}{2}\int_{0}^{\infty} \exp\left\{\frac{(r^2-t^2)}{2}\eta\right\} d\eta = \frac{2}{-(t^2-r^2)} \times \frac{r^2}{2} = \frac{1}{1-\frac{t^2}{r^2}} = \frac{e^{\mu t}}{1-t^2b^2}$$

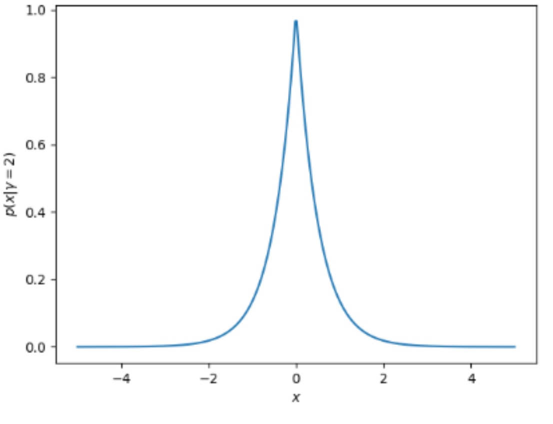
We observe that this is the mgf of the Laplace distribution where $p(x|r) = L(0, \frac{1}{r})$

the marginalised distribution $p(x|r)$ is a laplace distribution, with

$$L(\mu, b) = \frac{1}{b}\exp\left\{-\frac{|x-\mu|}{b}\right\},$$ non-differentiable at its mean (in this case, 0)



P(x|eta=3)



p(x|gamma=2)

Question 2

For Bayesian Linear Regression model, with likelihood $p(y|x,\omega) = \mathcal{N}(\omega^T x, \beta^{-1})$ and prior $p(\omega) =$ Normal $(0, \lambda^{-1} I)$

The predictive posterior is $p(y_*|x_*) = N(\mu_N^T x_*, \beta^{-1} + x_*^T \Sigma_N x_*)$
$= \mathcal{N}(\mu_N^T x_*, \sigma_N^2(x_*))$

where $\mu_N(x_*) = \Sigma(\beta \sum_{n=1}^N y_n x_n)$.
and $\sigma_N^2(x_*) = \beta^{-1} + x_*^T \Sigma_N x_*$; $\Sigma_N = (\beta \sum_{n=1}^N x_n x_n^T + \lambda I)^{-1}$

We have to prove that $\sigma_{N+1}^2(x_*) \leq \sigma_N^2(x_*)$

Observe that $\sigma_N^2 - \sigma_{N+1}^2 = x_*^T \left[(\beta \sum_{n=1}^N x_n x_n^T + \lambda I)^{-1} - (\beta \sum_{n=1}^{N+1} x_n x_n^T + \lambda I)^{-1} \right] x_*$

Take $\beta \sum_{n=1}^N x_n x_n^T + \lambda I = M$.

Then, $\sigma_N^2 - \sigma_{N+1}^2 = x_*^T \left[(M^{-1}) - (M + \beta x_{N+1} x_{N+1}^T)^{-1} \right] x_*$

We use the matrix identity $(M + vv^T)^{-1} = M^{-1} - \frac{(M^{-1}v)(v^T M^{-1})}{1 + v^T M^{-1} v}$

take $v = \sqrt{\beta} x_{N+1}$

Then, $\sigma_N^2 - \sigma_{N+1}^2 = x_*^T \left[M^{-1} - (M + vv^T)^{-1} \right] x_*$
$= x_*^T \left[\frac{(M^{-1}v)(v^T M^{-1})}{1 + v^T M^{-1} v} \right] x_*$. Since we see that M is a symmetric matrix (Since I is symmetric, $\sum x_n x_n^T$ is symmetric so their inverse is symmetric),

$(M^{-1}v)^T = v^T M^{-T} = v^T M^{-1}$

and $v^T M^{-1} v > 0$ [psd matrix M^{-1}].

$\sigma_N^2 - \sigma_{N+1}^2 = \frac{(x_*^T (M^{-1}v)).((v^T M^{-1}) x_*)}{1 + v^T M^{-1} v} = \frac{\text{non-negative } \cancel{\text{positive}} \text{ value}}{\text{positive value}} \geq 0.$

Thus, $\sigma_N^2 \geq \sigma_{N+1}^2$.

Hence, proved

Question 3.

Given N observations $\vec{x}_1, \vec{x}_2, \dots \vec{x}_N$ with each $\vec{x}_n \in R^D$,

Consider observations

$$\vec{x}_n = A\vec{z}_n + \vec{\epsilon}_n \quad \text{, with } \vec{\epsilon}_n \sim \mathcal{N}(0, \psi)$$

and $A = [\vec{a}_1, \vec{a}_2, \dots \vec{a}_K]$ D*K matrix,

and $z_n = [z_{n1}, z_{n2}, \dots z_{nK}]^T$, s.t. $Z = \{z_1, z_2 \dots z_N\}$.

1\. Suppose Z is ~~given~~ known, and assuming $\vec{a}_k$ prior, $a_k \sim \mathcal{N}(0, D^{-1}I_D)$.

We have to derive $p(\vec{a}_k | X, Z)$.

$$X = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{N1} \\ x_{12} & x_{22} & & x_{N2} \\ x_{13} & x_{23} & & \vdots \\ \vdots & \vdots & & \vdots \\ x_{1D} & x_{2D} & & x_{ND} \end{bmatrix} = [\vec{a}_1 \vec{a}_2 \vec{a}_3 \dots \vec{a}_K] Z + [\vec{\epsilon}_1 \vec{\epsilon}_2 \dots \vec{\epsilon}_N].$$

$$\Rightarrow [x_{1k}\ x_{2k} \dots x_{Nk}] = \vec{a}_k^T Z + [\epsilon_{1k}\ \epsilon_{2k} \dots \epsilon_{Nk}].$$

Let $\vec{X}_k$ be $[x_{1k}\ x_{2k}\ x_{3k} \dots x_{Nk}]^T$.

and $E_k = [\epsilon_{1k}\ \epsilon_{2k} \dots \epsilon_{Nk}]^T$.

This gives $\vec{X}_k = Z^T\vec{a}_k + \vec{E}_k$

Notice that each $\vec{a}_k$ depends upon only the k^{th} component of $\vec{x}_i$, that is, all other components have no effect on $\vec{a}_k$. Similarly, only the k^{th} component of ϵ has an effect on $\vec{a}_k$.

for this, marginalise components $i = 1:D$, $i \neq k$ of ϵ, such that the resulting probability distribution of ϵ_{-k} is $\mathcal{N}(0, \sigma_k^2)$, σ_k^2 is the marginalised variance of k^{th} component of ϵ.

Then, using the properties of probability distributions, we get:

$$p(a_k) = \mathcal{N}(a_k | 0, D^{-1}I_D), \quad p(X | \vec{a}_k, Z, \psi) = \mathcal{N}(Z^T\vec{a}_k + \vec{E}_k, \sigma_k^{-2} I_N)$$

$$\Rightarrow p(\vec{a}_k | X, Z) = \mathcal{N}\left(\vec{a}_k \middle| (DI_D + Z\sigma_k^{-2}I_N Z^T)^{-1} Z\sigma_k^{-2} I_N \vec{X}_k,\ (DI_D + Z\sigma_k^{-2} I_N Z^T)^{-1}\right)$$

2. Now, we know $A = \{\vec{a}_1, \vec{a}_2, \ldots \vec{a}_K\}$. This gives a linear transformation of $\vec{x}_n = A\vec{z}_n + \vec{\varepsilon}_n$; $\varepsilon_n \sim \mathcal{N}(0, \psi)$

with a prior of $p(\vec{z}_n) = \mathcal{N}(0, \gamma^{-1} I_K)$.

and likelihood $p(\vec{x}_n \mid \vec{z}_n, A) = \mathcal{N}(x_n \mid A\vec{z}_n + b, \psi)$

Using Baye's Theorem for gaussians,

$$p(\vec{z}_n) = \mathcal{N}(z_n \mid 0, \Lambda^{-1}) \text{ where } \Lambda = \gamma I_K$$

$$\text{and } p(\vec{x}_n \mid \vec{z}_n, A) = \mathcal{N}(\vec{x}_n \mid A\vec{z}_n, L^{-1}) \text{ where } L = \psi^{-1}$$

we know that

$$p(\vec{z}_n \mid \vec{x}_n, A) = \mathcal{N}(\vec{z}_n \mid \Sigma\{A^T L \vec{x}_n + \Lambda \cdot 0\}, \Sigma) \; ; \; \Sigma = (\Lambda + A^T L A)^{-1}$$

$$= \mathcal{N}(\vec{z}_n \mid (\gamma I_K + A^T \psi^{-1} A)^{-1} A^T \psi^{-1} \vec{x}_n, (\gamma I_K + A^T \psi^{-1} A)^{-1})$$

since z_n only depends on $\vec{x}_n$ and <u>not</u> $\vec{x}_K$ for $K \neq n$, therefore,

$$p(\vec{z}_n \mid \cancel{x_n}, A) = \mathcal{N}(\vec{z}_n \mid ((\gamma I_K + A^T \psi^{-1} A)^{-1} A^T \psi^{-1} x_n), (\gamma I_K + A^T \psi^{-1} A)^{-1})$$

Question 4

Given spike and slab prior $p(w|b, \sigma_{sp}^2, \sigma_{sl}^2) = \begin{cases} \mathcal{N}(w|0, \sigma_{sp}^2) ; & b=0 \\ \mathcal{N}(w|0, \sigma_{sl}^2) ; & b=1. \end{cases}$

This is modelled by bernoulli b; s.t. $b=1$ with probability $\pi = \frac{1}{2}$ and $b=0$ with probability $(1-\pi) = \frac{1}{2}$

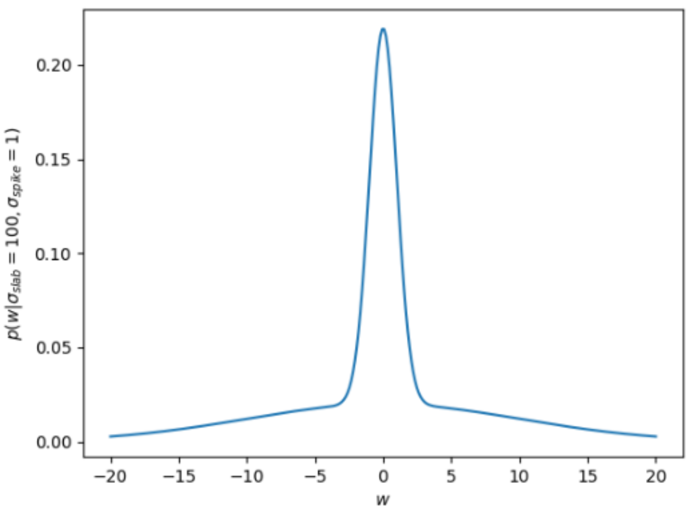
Then, $p(w|b, \sigma_{sp}^2, \sigma_{sl}^2) = \mathcal{N}(w|0, (\sigma_{sp}^2)^{1-b}(\sigma_{sl}^2)^b)$

Marginal prior $p(w|\sigma_{sl}^2, \sigma_{sp}^2) = \sum_{b\in\{0,1\}} p(w|b, \sigma_{sl}^2, \sigma_{sp}^2)\, p(b)$

$$= \frac{1}{2}\mathcal{N}(w|0, \sigma_{sp}^2) + \frac{1}{2}\mathcal{N}(w|0, \sigma_{sl}^2)$$

$$= \frac{1}{2}\left[\frac{1}{\sqrt{2\pi\sigma_{sp}^2}} \exp\left\{\frac{-w^2}{2\sigma_{sp}^2}\right\} + \frac{1}{\sqrt{2\pi\sigma_{sl}^2}} \exp\left\{\frac{-w^2}{2\sigma_{sl}^2}\right\}\right]$$

A mixture of 2 gaussians



plot for $p(w|\sigma_{sl}^2 = 100, \sigma_{sp}^2 = 1)$

Now, considering a noisy version of w, modelled by $x = w + \varepsilon$, $\varepsilon \sim \mathcal{N}(\varepsilon|0, \rho^2)$

$\Rightarrow$ since w and ε are independent variables,

and $w \sim \mathcal{N}(w|0, (\sigma_{sp}^2)^{1-b}(\sigma_{sl}^2)^b)$, $\varepsilon \sim \mathcal{N}(\varepsilon|0, \rho^2)$

we can write x as $x \sim \mathcal{N}(x|0, \rho^2 + (\sigma_{sp}^2)^{1-b}(\sigma_{sl}^2)^b)$

[using the results that $x = w + \varepsilon \Rightarrow \mu_x = \mu_w + \mu_\varepsilon$ and $\Sigma_x = \Sigma_w + \Sigma_\varepsilon$]

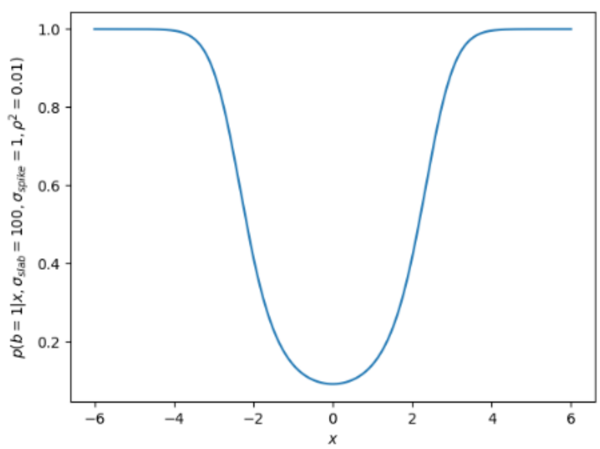
Thus,

$$p(b, x|\rho^2, \sigma_{sp}^2, \sigma_{sl}^2) = \mathcal{N}(x|0, \rho^2 + (\sigma_{sp}^2)^{1-b}(\sigma_{sl}^2)^b)\, p(b)$$

Also, $p(b|x,\rho^2,\sigma^2_{sl},\sigma^2_{sp}) = \frac{p(b,x|\rho^2,\sigma^2_{sl},\sigma^2_{sp})}{p(x|\rho^2,\sigma^2_{sl},\sigma^2_{sp})}$

$$= \frac{\mathcal{N}(x|0,\rho^2+(\sigma^2_{sp})^{1-b}(\sigma^2_{sl})^{b})\times\frac{1}{2}}{\mathcal{N}(x|0,\rho^2+\sigma^2_{sp})\times\frac{1}{2}+\mathcal{N}(x|0,\rho^2+\sigma^2_{sl})\times\frac{1}{2}}$$

$$p(b=1|x,\rho^2,\sigma^2_{sp},\sigma^2_{sl}) = \frac{\mathcal{N}(x|0,\rho^2+\sigma^2_{sl})}{\mathcal{N}(x|0,\rho^2+\sigma^2_{sl})+\mathcal{N}(x|0,\rho^2+\sigma^2_{sp})}$$



Plot of $p(b=1|x,\rho^2,\sigma^2_{sp},\sigma^2_{sl})$ with $\rho^2=0.01$, $\sigma^2_{sp}=1$, $\sigma^2_{sl}=100$

posterior distribution, $p(w|x,\rho^2,\sigma^2_{sp},\sigma^2_{sl}) = \frac{p(x|w)p(w)}{p(x)}$ [Ignoring $\rho^2,\sigma^2_{sp},\sigma^2_{sl}$]

observe that since $x=w+\epsilon$, $\epsilon\sim\mathcal{N}(\epsilon|0,\rho^2)$,

~~it follows that $w = x+\epsilon$, $\epsilon\sim\mathcal{N}(\epsilon|-0,\rho^2)=\mathcal{N}(\epsilon|0,\rho^2)$.~~

$\Rightarrow p(x|w) = \mathcal{N}(x|w,\rho^2)$

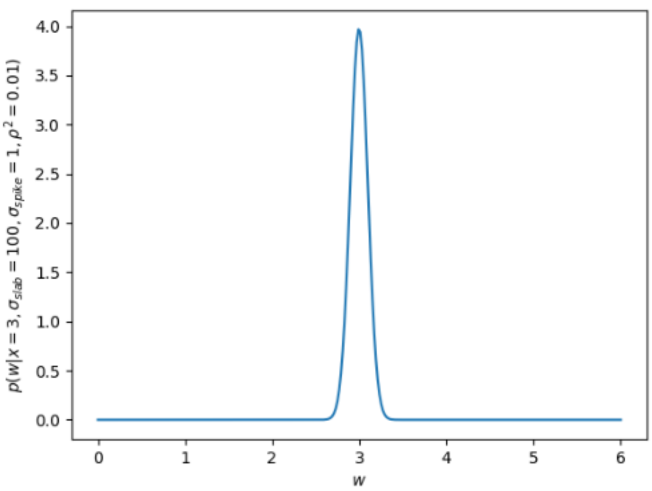
also, $p(w)=\frac{1}{2}[\mathcal{N}(w|0,\sigma_{sp}^2)+\mathcal{N}(w|0,\sigma_{sl}^2)]$

and $p(x)=\frac{1}{2}[\mathcal{N}(x|0,\rho^2+\sigma_{sp}^2)+\mathcal{N}(x|0,\rho^2+\sigma_{sl}^2)]$

$$\Rightarrow p(w|x,\rho^2,\sigma_{sp}^2,\sigma_{sl}^2) = \frac{\mathcal{N}(x|w,\rho^2)[\mathcal{N}(w|0,\sigma_{sp}^2)+\mathcal{N}(w|0,\sigma_{sl}^2)]}{\mathcal{N}(x|0,\rho^2+\sigma_{sp}^2)+\mathcal{N}(x|0,\rho^2+\sigma_{sl}^2)}$$

and, $p(x|w,\rho^2)\triangleq p(w|x,\rho^2)$

$$\Rightarrow p(w|x,\rho^2,\sigma_{sp}^2,\sigma_{sl}^2) = \frac{\mathcal{N}(w|x,\rho^2)[\mathcal{N}(w|0,\sigma_{sp}^2)+\mathcal{N}(w|0,\sigma_{sl}^2)]}{\mathcal{N}(x|0,\rho^2+\sigma_{sp}^2)+\mathcal{N}(x|0,\rho^2+\sigma_{sl}^2)}$$



$p(w|x=3, \sigma_{slab}=100, \sigma_{spike}=1, \rho^2=0.01)$
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6
w

Question 5

Edward is a probabilistic modeling library built on top of TensorFlow and written in python. Edward's design reflects an iterative process pioneered by Edward Box:

- Build a model of a phenomenon. The model consists of probability distributions, observed data, and graphical models.
  The data can be preloaded, fed into the program during runtime using TensorFlow placeholders, or directly read from files in case of large size.
  A probabilistic model is a joint distribution $p(\boldsymbol{x}, \boldsymbol{z})$ of data $\boldsymbol{x}$ and latent variables $\boldsymbol{z}$. Random variables are generated on the fly when the graph is initiated.

- Make inferences about the model given the data. Edward has many inference algorithms, including variational and black-box inference, Markov Chain Monte Carlo(MCMC), and a symbolic library in production, which will help in exact inference with symbolic probability distributions and closed form solutions.

- Criticize the model's fit to the data.
  Edward explores model criticism using point estimates of latent variables, and posterior predictive checks.

In the following section, we use Edward to model data using end-to-end Bayesian linear regression, using weight $\boldsymbol{w}$ and intercept $\boldsymbol{b}$

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \sigma_w^2 \boldsymbol{I})$$
$$p(b) = \mathcal{N}(b|0, \sigma_b^2)$$
$$p(\boldsymbol{y}|\boldsymbol{w}, b, \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|\boldsymbol{x}_n^T \boldsymbol{w} + b, \sigma_y^2)$$

Fixing $\sigma_w, \sigma_b, \sigma_y = 1$,

```
import numpy as np
import edward as ed
import tensorflow as tf

#Import the Normal distribution
from edward.models import Normal

N = 40 #Number of data points
D = 10 #Number of features

#Initializing variables using TensorFlow variables and placeholders, defining models
X = tf.placeholder(tf.float32, [N, D])
w = Normal(mu=tf.zeros(D), sigma=tf.ones(D))
b = Normal(mu=tf.zeros(1), sigma=tf.ones(1))
y = Normal(mu=ed.dot(X, w) + b, sigma=tf.ones(N))

#Building a toy dataset
def build_toy_dataset(N, w, noise_std=0.1):
  D = len(w)
  x = np.random.randn(N, D).astype(np.float32)
  y = np.dot(x, w) + np.random.normal(0, noise_std, size=N)
  return x, y

w_true = np.random.randn(D)
X_train, y_train = build_toy_dataset(N, w_true)
X_test, y_test = build_toy_dataset(N, w_true)

#Inference part - using Variational Inference
```

```
qw = Normal(mu=tf.Variable(tf.random_normal([D])),
            sigma=tf.nn.softplus(tf.Variable(tf.random_normal([D])))) #Initializing required w with
            #a random mean and diagonal covariance matrix.
qb = Normal(mu=tf.Variable(tf.random_normal([1])),
            sigma=tf.nn.softplus(tf.Variable(tf.random_normal([1])))) #Initializes the bias
            #with random mean and variance

#Run Variational Inference using Kullback-Leibler divergence, using a default of 500 iterations.
inference = ed.KLqp({w: qw, b: qb}, data={X: X_train, y: y_train}) #Data fed into dicts into the
                                                                    #KLqp inference method of Edward
inference.run() #Initializes the back-end TensorFlow graph

#Criticism part - We test our model by point based evaluations on test data.
#Form the Posterior predictive distribution
y_post = Normal(mu=ed.dot(X, qw.mean()) + qb.mean(), sigma=tf.ones(N))
#We can evaluate various point based quantities using PPD.
print("Mean squared error on test data:")
print(ed.evaluate('mean_squared_error', data={X: X_test, y_post: y_test}))#Prints a value of 0.012
```

As we see, Edward is primarily a library for black-box inference, and other variational inference algorithm. The computational backend of TensorFlow provides Edward massive speed boost of 35X compared to Stan or PyMC3.