This week we started by covering parameter estimation with maximum likelihood estimation (MLE). The MLE is the value of the parameter which maximizes the likelihood L(parameter;x). The likelihood is just a notational change which is equivalent to the probability density function f(x|parameter). Thus the estimate which maximizes the likelihood/probability is the maximum likelihood estimate. In many cases to find the MLE we take the maximum log likelihood (MLL) as adding a monotonous function does not change the maximum, but the log simplifies many derivative calculations particularly with multiplications and exponentials, commonly seen throughout distributions.

We also discussed characteristics of MLEs. Reparametrizing the probability into the likelihood lets us notice that the estimator is itself a random variable dependent on the random variables of the collected data. There is bias which is the discrepancy between the estimator's mean and the parameter value used to describe the collected data's probability density function. Consistency is the guarantee that the estimator approaches the parameter value as the number of data increases. Efficiency refers to how close the MSE of the estimator is with the minimum MSE. For this the Cramer-Rao lower bound and the Fisher information matrix is used to determine the lowest value of the MSE of the estimator.

We finished up with Bayesian estimation which uses the Bayes Rule to determine the posterior density of an estimator to use either the conditional mean or the Maximum a Posterior (MAP) estimate. Bayesian estimation is a powerful estimation tool that is used in many domians especially the MAP, such as computer vision and system identification. Bayesian optimization tools are also used in hyperparameter tuning as well.

In the end probability distributions are models that we give to the (random) data around us and being able to estimate the parameters for that model is similar to the model approximation/fitting problems we have been solving in this class. With the MLE we are now able to handle parameter estimation under probability.

Akash Patel
MFML- HW9
11/9/20

2. Let $Z_1, \dots, Z_N$ be a sequence of independent Gaussian random variables with mean 0 and variance 1. You observe the random vector X in $\mathbb{R}^N$ that is generated through the autoregressive process

$$X_k = \begin{cases} Z_1, & k=1 \\ aX_{k-1} + Z_k, & k>1. \end{cases}$$

Given $X=x$, find the MLE for $a \in \mathbb{R}$. The conditional independence structure makes this a Markov process, meaning that we can factor the distribution for $X \in \mathbb{R}^N$ as

$$f_X(x) = f_{X_1}(x_1) \cdot f_{X_2}(x_2|x_1) \cdot f_{X_3}(x_3|x_2) \cdots f_{X_N}(x_N|x_{N-1}).$$

$$MLE = \hat{\theta}_{MLE} = \underset{\theta \in T}{\arg\max}\, L(\theta; x_1, \dots, x_N) = \arg\min -\ell(\theta; x_1, \dots x_N)$$

Here $\theta = (a)$

$$L(\theta; x_1, \dots x_N) = f_{X_1,\dots,X_N}(x_1,\dots,x_N) = f_{X_1}(x_1) \cdot f_{X_2}(x_2|x_1) \cdots f_{X_N}(x_N|x_{N-1})$$

$$\ell(\theta; x_1, \dots x_N) = \log L(\theta; x_1, \dots, x_N) = \log(f_{X_1}(x_1)) + \sum_{k=2}^{N} \log(f_{X_k}(x_k|x_{k-1}))$$

$$f_{X_1}(x_1) = f_{Z_1}(z_1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\tfrac{1}{2}(z_1-\mu)\sigma^{-2}(z_1-\mu)\right) \quad \text{with } \sigma=1, \mu=0$$

$$= \frac{1}{\sqrt{2\pi}} \exp\left(-\tfrac{1}{2} z_1^2\right) \quad \Rightarrow \sigma_{X_1} = 1 \quad \mu_{X_1} = 0$$

$f_{X_k}(x_k|x_{k-1})$: $\mu_{X_k} = E[aX_{k-1} + Z_k] = aE[X_{k-1}] + E[\overset{0}{Z_k}] = a\mu_{X_{k-1}}$ For $\mu_{X_2} = a\mu_{X_1} = a(0)$ and so on.

Thus $\underline{\underline{\mu_{X_k} = 0}}$

This is incorrect see further below!!

$$\sigma_{X_k}^2 = E[X_k^2] - E[X_k]^2 = E[(aX_{k-1}+Z_k)(aX_{k-1}+Z_k)] - E[aX_{k-1}+Z_k]^{\overset{0}{2}}$$

$$= E[(a^2X_{k-1}^2 + Z_k^2 + 2aX_{k-1}Z_k)] = a^2E[X_{k-1}^2] + E[Z_k^2] + 2aE[X_{k-1}Z_k]$$ independent Gaussians

$$= a^2\sigma_{X_{k-1}}^2 + \sigma_{Z_k}^2 = a^2\sigma_{X_{k-1}}^2 + 1 \qquad \Rightarrow E[X^2] = \sigma^2 \text{ for } \mu=0 \text{ and Gaussians}$$

For X_1: $\sigma_{X_1}^2 = \sigma_{Z_k}^2 = 1$

X_2: $\sigma_{X_2}^2 = a^2\sigma_{X_1}^2 + 1 = a^2(1) + 1 = a^2 + 1$

X_3: $\sigma_{X_3}^2 = a^2\sigma_{X_2}^2 + 1 = a^2(a^2+1) + 1 = a^4 + a^2 + 1$

$\underline{X_k: \sigma_{X_k}^2 = 1 + \sum_{i=1}^{k-1} a^{2i}}$

$\sigma^2_{x_k} = 1 + \sum_{i=1}^{k-1} a^{2i} \qquad \mu_{x_k} = 0$

$$f_{x_k}(x_k|x_{k-1}) = \frac{1}{\sqrt{2\pi\sigma^2_{x_k}}} \exp\left(-\tfrac{1}{2} x_k^2 \sigma^{-2}_{x_k}\right)$$

$$\ell(\theta; x_1,\dots,x_N) = \log L(\theta; x_1,\dots,x_N) = \log(f_{x_1}(x_1)) + \sum_{k=2}^{N} \log(f_{x_k}(x_k|x_{k-1}))$$

$$= \log\left(\frac{1}{\sqrt{2\pi}} \exp\left(-\tfrac{1}{2}x_1^2\right)\right) + \sum_{k=2}^{N} \log\left(\frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{\sigma^2_{x_k}(a)}} \exp\left(-\tfrac{1}{2} x_k^2 \sigma^{-2}_{x_k}(a)\right)\right)$$

$\sigma_{x_k}(a)$ signifies function of a.

$$= \log\left(\frac{1}{\sqrt{2\pi}} \exp\left(-\tfrac{1}{2}x_1^2\right)\right) + \sum_{k=2}^{N} \log\left(\frac{1}{\sqrt{2\pi}}\right) + \sum_{k=2}^{N} \log\left(\frac{1}{\sqrt{\sigma^2_{x_k}(a)}}\right) + \sum_{k=2}^{N} \log\left(\exp\left(-\tfrac{1}{2} x_k^2 \sigma^{-2}_{x_k}(a)\right)\right)$$

$$= \swarrow + \swarrow + \swarrow + \sum_{k=2}^{N} \frac{-\frac{1}{2}x_k^2}{\sigma^2_{x_k}(a)}$$

$$\frac{\partial \ell(\theta;\cdot)}{\partial \theta} = \frac{\partial \ell(a;\cdot)}{\partial a} = 0 + 0 + \sum_{k=2}^{N} \sqrt{\sigma^2_{x_k}(a)} \frac{\partial \sigma^2_{x_k}}{\partial a} + \sum_{k=2}^{N} \frac{\frac{1}{2}x_k^2}{\sigma^3_{x_k}(a)} \frac{\partial \sigma^2_{x_k}(a)}{\partial a} \quad ?$$

This seems too complicated hmm...
above I treat X_{k-1} as a random variable. But since $X_{k-1} = x_{k-1}$ is given I should treat it as a constant.

$f(x_k|x_{k-1})$: $E[X_k] = E[aX_{k-1} + Z_k] = E[\underbrace{aX_{k-1}}_{\text{given}}] + \cancelto{0}{E[Z_k]} = aX_{k-1}$

$Var[X_k] = Var[aX_{k-1} + Z_k] = Var[Z_k] = 1$

constants do not change variance

$f(x_k|x_{k-1})$: $X_k \sim Normal(a x_{k-1}, 1)$

$$\ell(\theta; x_1,\dots,x_N) = \log L(\theta; x_1,\dots,x_N) = \log(f_{x_1}(x_1)) + \sum_{k=2}^{N} \log(f_{x_k}(x_k|x_{k-1}))$$

$$= \log\left(\frac{1}{\sqrt{2\pi}} \exp\left(-\tfrac{1}{2}x_1^2\right)\right) + \sum_{k=2}^{N} \log\left(\frac{1}{\sqrt{2\pi}} \exp\left(-\tfrac{1}{2}(x_k - a x_{k-1})^2\right)\right)$$

$$= A + \sum_{k=2}^{N} B + \sum_{k=2}^{N} -\tfrac{1}{2}(x_k - a x_{k-1})^2$$

replace terms w/o a as constants

$$\frac{\partial \ell(\theta;x)}{\partial \theta} = \frac{\partial \ell(a;\cdot)}{\partial a} = 0 + 0 + \sum_{k=2}^{N} -(x_k - a x_{k-1})(-x_{k-1}) = \sum_{k=2}^{N} (x_k x_{k-1} - a x_{k-1}^2)$$

Set equal to 0:

$$\sum_{k=2}^{N} (x_k x_{k-1} - a x_{k-1}^2) = 0 \qquad \sum_{k=2}^{N} a x_{k-1}^2 = \sum_{k=2}^{N} x_k x_{k-1}$$

$$\boxed{a = \frac{\sum_{k=2}^{N} x_k x_{k-1}}{\sum_{k=2}^{N} x_{k-1}^2}}$$

3. Let A be an $M \times N$ matrix with full column rank. Let E be a Gaussian random vector in $\mathbb{R}^M$ with mean 0 and covariance R_e. Suppose we observe

$$Y = A\theta_0 + E,$$

where $\theta_0 \in \mathbb{R}^N$ is unknown.

a) What is the distribution of Y and how does it depend on θ_0?

$$E[Y] = E[A\theta_0 + E] = E[A\theta_0] + E[E] = A\theta_0 + 0$$

$$Var[Y] = Var[A\theta_0 + E] = Var[E] = R_e$$

variance is invariant to changes with constants

$$\boxed{Y \sim \text{Normal}(A\theta_0, R_e)}$$

Depends on θ_0 by scaling the mean.

b) Find a closed form expression for the maximum likelihood estimate of θ_0. (In this case, we are working from a single sample of a random vector.)

$$L(\theta_0; y) = \frac{1}{(2\pi)^{D/2}\sqrt{\det(R_e)}} \exp\left(-\tfrac{1}{2}(y - A\theta_0)^T R_e^{-1}(y - A\theta_0)\right)$$

This problem is similar to pg. 38 of the notes where the result for N samples is as follows:

$$\hat{\mu}_{MLE} = (A^T A)^{-1} A^T y = \frac{1}{N}\sum_{n=1}^{N} x_n \quad \text{where } A = \begin{bmatrix} R^{1/2} \\ \vdots \\ R^{1/2} \end{bmatrix} \text{ and } y = \begin{bmatrix} R^{1/2}x_1 \\ \vdots \\ R^{1/2}x_N \end{bmatrix}$$

Note this is a different A.

Collapsing this case for our problem we have: (single sample)

$$\hat{\mu}_{MLE} = (R^{T/2} R^{1/2})^{-1} R^{T/2} R^{1/2} y = y$$

For our case we are also given a linear transformation hence:

$$\hat{\mu}_{MLE} = A\hat{\theta}_{MLE} = y \qquad A^T A \hat{\theta}_{MLE} = A^T y \qquad \boxed{\hat{\theta}_{MLE} = (A^T A)^{-1} A^T y}$$

c) What is the distribution of the MLE estimator $\hat{\Theta}$? Is $\hat{\Theta}$ unbiased?

$$\hat{\Theta} = g(Y) = (A^T A)^{-1} A^T Y$$

$\hat{\theta}_{MLE} = (A^T A)^{-1} A^T y$ and y itself is sampled from random variable Y.

Thus $\hat{\Theta} = (A^T A)^{-1} A^T Y$ Linear transformation so still Gaussian.

$$\circledast\ (A^TA)^{-1}A^TR_eA(A^TA)^{-T} = \cancel{(A^TA)^{-1}}\cancel{A^T}A(A^+R_e^{-1}A)^{-T} = (A^TR_e^{-1}A)^{-T} = \underline{(A^TR_e^{-1}A)^{-1}}$$

$$E[\hat{\Theta}] = E[\overbrace{(A^TA)^{-1}A^T}^{A^+}Y] = (A^TA)^{-1}A^TE[Y] = (A^TA)^{-1}A^TA\theta_0 = \boxed{\theta_0 = E[\hat{\Theta}]}$$

$$Var[\hat{\Theta}] = E[(\hat{\Theta}-E(\hat{\Theta}))(\hat{\Theta}-E(\hat{\Theta}))^T] = E[\hat{\Theta}\hat{\Theta}] - E[\hat{\Theta}]E[\hat{\Theta}]^T = E[\hat{\Theta}\hat{\Theta}^T] - \underbrace{\theta_0\cancel{\theta_0^T}}$$

$$E[(A^+Y)(A^+Y)^T] = E[A^+YY^TA^{+T}] = A^+E[YY^T]A^{+T} = A^+R_eA^{+T} + \cancel{\theta_0\theta_0^T}$$

$$\boxed{Var[\hat{\Theta}] = A^+R_eA^{+T}} = \hat{R} = (A^TR_e^{-1}A)^{-1}\ \circledast \quad / \quad \underbrace{A^+A}_{I}\theta_0\theta_0^T\underbrace{A^TA^{+T}}_{I} + A^+R_eA^{+T}$$

$\boxed{\text{Unbiased, since } E[\hat{\Theta}] = \theta_0\ \checkmark}$

$$Y = A\theta_0 + E \qquad A\theta_0\theta_0^TA^T \qquad R_e$$

$$E[YY^T] = E[(A\theta_0+E)(A\theta_0+E)^T] = E[A\theta_0\theta_0^TA^T] + E[EE^T]$$

d) What is the MSE of the MLE, $E[\|\hat{\theta}-\theta_0\|_2^2]$?

$$MSE(\hat{\theta}) = trace(\hat{R}) + Bias(\hat{\theta})^2$$

Since the MLE is unbiased:

$$\boxed{MSE(\hat{\theta}) = trace(\hat{R}), = trace(Var[\hat{\Theta}]) = trace(A^+R_eA^{+T}) = trace((A^TR_e^{-1}A)^{-1})}$$

e) Compute the Fisher information matrix $J(\theta_0)$ and verify that the MLE meets the Cramer-Rao lower bound.

$$J(\theta_0) = E[s(\theta_0;Y)s(\theta_0;Y)^T]$$
$$= E[\nabla_\theta \log f_Y(Y;\theta_0)\nabla_\theta \log f_Y(Y;\theta_0)^T]$$

$$f_Y(Y;\theta_0) = \underbrace{\frac{1}{(2\pi)^{D/2}\sqrt{\det(R_e)}}}_{C}\exp\left(-\tfrac{1}{2}(y-A\theta)^TR_e^{-1}(y-A\theta)\right)$$

$$\log f_Y(y;\theta_0) = \log(C) + \log(\exp(-\tfrac{1}{2}(y-A\theta_0)^TR_e^{-1}(y-A\theta_0)))$$
$$= D - \tfrac{1}{2}(y-A\theta_0)^TR_e^{-1}(y-A\theta_0) = D - \tfrac{1}{2}(y^TR_e^{-1}y + (A\theta)^TR_e^{-1}(A\theta) - 2y^TR_e^{-1}(A\theta))$$

$$\nabla_\theta \log f_Y(y;\theta_0) = 0 - 0 + y^TR_e^{-1}A - \tfrac{1}{2}2\theta^TA^TR_e^{-1}A$$
$$= y^TR_e^{-1}A - \theta^TA^TR_e^{-1}A = (y^T - \theta^TA^T)R_e^{-1}A = (y^T-(A\theta)^T)R_e^{-1}A) = (y-A\theta)^TR_e^{-1}A$$

$1\times m \quad m\times m \quad m\times n$

$$\nabla_\theta \log f_Y(y;\theta_0)^T = ((y-A\theta)^TR_e^{-1}A)^T = A^T\underset{\text{symmetric}}{R_e^{-1}}(y-A\theta)$$

$n\times m$

$$J(\theta_0) = E[\nabla_\theta \log f_Y(y;\theta_0)^T\nabla_\theta \log f_Y(y;\theta_0)] = E[A^TR_e^{-1}(y-A\theta)(y-A\theta)^TR_e^{-1}A]$$
$$= A^TR_e^{-1}E[(y-A\theta)(y-A\theta)^T]R_e^{-1}A = \qquad R_e = E[(y-A\theta)(y-A\theta)^T]$$
$$= \underline{A^TR_e^{-1}R_e}R_e^{-1}A$$
$$= \boxed{A^TR_e^{-1}A}$$

Verify the Cramer Rao Bound:

Cramer-Rao Bound:

$$MSE(\hat{\Theta}) \geq \frac{1}{N} trace(J(\theta_0)^{-1}) \quad N=1 \text{ sample}$$

$$trace(A^{+}R_e A^{+}) \geq trace((A^T R_e^{-1} A)^{-1}) \checkmark \quad trace(A^{+}R_e A^{+}) = trace((A^T R_e^{-1} A)^{-1})$$

f) Defend the following statement: The MLE is the best unbiased estimator of θ_0.

In addition to satisfying the Cramer-Rao lower-bound the MSE for the MLE is actually equal to the trace of the inverse Fisher Matrix. Since we cannot obtain a value any lower than this the MLE is the best unbiased estimator of θ_0.

g) Compare your answer to that of Homework 8, Problem 4. How is it different and why?

$Y = A\theta_0 + E \quad$ MSE: $trace((A^T R_e^{-1} A)^{-1})$

$\hat{\theta}_{MLE} = (A^T A)^{-1} A^T y$

HW8:

$Y = AX + E$

$\hat{X} = R_x A^T (A R_x A^T + R_e)^{-1} y$

MSE: $trace(R_x - R_x A^T (A R_x A^T + R_e^{-1}) A R_x)$

In Prob 8.4 X is a random variable, whereas in this problem the corresponding variable θ_0 is not a random variable. Hence we do not see R_x in our MLE.

4. A Cauchy random variable with "location parameter" v has a density function

$$f_X(x;v) = \frac{1}{\pi(1+(x-v)^2)}, \quad x \in \mathbb{R} \quad (1)$$

Despite its simple definition, this is a strange animal. First of all, its mean is not defined, as the integral $\int x/(1+x^2)\,dx$ is not absolutely convergent. It is also easy to see that the variance is infinite. But as you can see (especially if you sketch it), then density is symmetric around v, and v is certainly the median.

Let $X_1, X_2, \ldots, X_N$ be iid Cauchy random variables distributed as in (1). From observed data $X_1 = x_1, \ldots, X_N = x_N$, we will compare three estimates: the sample mean $\hat{v}_{mn} = \frac{1}{N}\sum_{n=1}^{N} x_n$

the sample median $\hat{v}_{md} = \begin{cases} x_{((N+1)/2)}, & N \text{ odd}, \\ \frac{x_{(N/2)} + x_{(N/2+1)}}{2}, & N \text{ even}, \end{cases}$

where $x_{(i)}$ is the largest value in $\{x_1, \ldots, x_N\}$, and the MLE

$$\hat{v}_{mle} = \arg\max_v L(v; x_1, \ldots, x_N) = \arg\max_v \sum_{n=1}^{N} \ell(v; x_n)$$

where $\ell(v; x_n) = \log f_X(x_n; v)$.

c) Find an integral expression for the expected log liklihood function $e(v) = E[\ell(v;X)]$ when X has Cauchy density $f_X(x;v_0)$ as in (1). Your expression should have the form

$$e(v) = \int_{-\infty}^{\infty} (\text{something that depends on } x, v, v_0)\, dx$$

$$E[\ell(v,X)] = \int_{-\infty}^{\infty} \ell(v;x) f_X(x;v_0)\,dx$$
$$= \int_{-\infty}^{\infty} \log f_X(x,v) f_X(x;v_0)\,dx$$

See code and plots for complete solution.

5. Three friends, Aaron, Blake, and Colin, meet together every week to play poker. They each buy in for \$100, and play until one of them has it all. Poker is a game of skill, but also a game of luck - the winner each week is modeled as a discrete random variable X with distribution parameterized by θ_a, θ_b, with

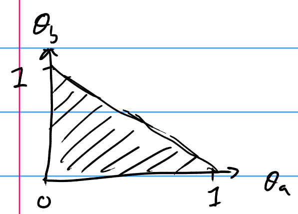
$$P(X=A)=\theta_a,\ P(X=B)=\theta_b,\ P(X=C)=1-\theta_a-\theta_b,$$

where $\theta_a, \theta_b \geq 0$, and $\theta_a+\theta_b \leq 1$.

Above, event A corresponds to Aaron winning, B corresponds to Blake winning and C corresponds to Colin winning.

The parameters θ_a and θ_b are unknown, and we want to infer them after observing the winners each week for many weeks. We have no idea of the relative skill of the players at the beginning of this experiment, so our prior is uniform on the triangular region specified by the constraints in (2):

$$\Theta=\begin{bmatrix}\theta_a\\ \theta_b\end{bmatrix},\ f_\Theta(\theta)=\begin{cases}2, & \theta\in S\\ 0, & \theta\notin S\end{cases} \qquad S=\{\theta\in\mathbb{R}^2 : \theta_a,\theta_b\geq 0,\ \theta_a+\theta_b\leq 1\}.$$



Area: $\frac{1}{2}(1)(1)=\frac{1}{2}$

Inside triangle: $f_\Theta(\theta)=2$, elsewhere $f_\Theta(\theta)=0$

a) Show that after N weeks, where we have observed N_a wins for Aaron, N_b wins for Blake, and $N_c=N-N_a-N_b$ wins for Colin, the posterior for Θ is given by the Dirichlet distribution

$$f_\Theta(\Theta \mid X_1=x_1,\ldots,X_N=x_N) \propto \theta_a^{N_b}\theta_b^{N_b}(1-\theta_a-\theta_b)^{N-N_a-N_b}$$

(The constant in front of the expression on the right turns out to be

$$\frac{\Gamma(N+3)}{\Gamma(N_a+1)\Gamma(N_b+1)\Gamma(N-N_a-N_b+1)}$$

which is the inverse of the integral of the expression on the right over the constraint set S.

Bayes Rule

$$f_\Theta(\theta|X_1=x_1,\dots,X_N=x_N) = \frac{f_X(X_1=x_1,\dots,X_N=x_N|\Theta=\theta)\cdot f_\Theta(\theta)}{f_X(X_1=x_1,\dots,X_N=x_N)}$$

$$f_X(X_1=x_1,\dots,X_N=x_N|\theta=\theta) = P(X=A)^{N_a}P(X=B)^{N_B}P(X=C)^{N_c}$$

$$= \theta_a^{N_a}\theta_b^{N_b}\theta_c^{N_c} = \theta_a^{N_a}\theta_b^{N_b}(1-\theta_a-\theta_b)^{(N-N_a-N_b)}$$

$f_\theta(\theta) = 2$ in S 0 elsewhere

$$f_X(X_1=x_1,\dots,X_N=x_N) = \int f_X(X_1=x_1,\dots,X_N=x_N|\theta=\theta)\, f_\theta(\theta)\, d\theta$$

$$= \int_S f_X(X_1=x_1,\dots,X_N=x_N|\theta=\theta) f_\theta(\theta)\, d\theta + \cancel{\int_{\bar S}(\cdot)} \to 0$$

$f_\theta(\theta)=0$ when not in S

$$= 2\int_S f_X(X_1=x_1,\dots,X_N=x_N|\theta=\theta)\, d\theta$$

$$= 2\int_S \theta_a^{N_a}\theta_b^{N_b}(1-\theta_a-\theta_b)^{(N-N_a-N_b)}\, d\theta$$

$$f_\Theta(\theta|X_1=x_1,\dots,X_N=x_N) = \frac{f_X(X_1=x_1,\dots,X_N=x_N|\Theta=\theta)\cdot f_\Theta(\theta)}{f_X(X_1=x_1,\dots,X_N=x_N)}$$

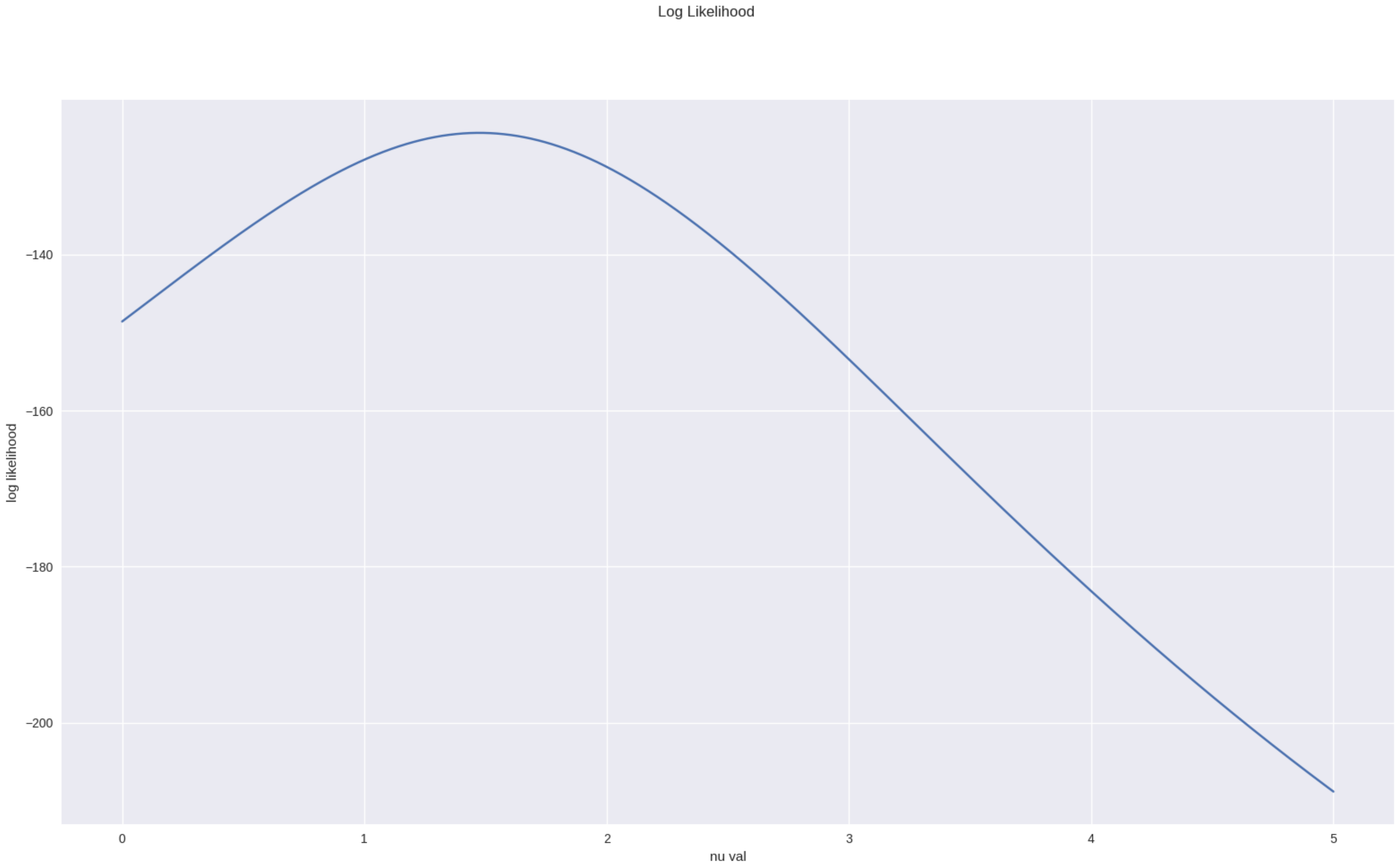
$$= \frac{\theta_a^{N_a}\theta_b^{N_b}(1-\theta_a-\theta_b)^{(N-N_a-N_b)}\cdot \cancel{2}}{\cancel{2}\underbrace{\int_S \theta_a^{N_a}\theta_b^{N_b}(1-\theta_a-\theta_b)^{(N-N_a-N_b)}\, d\theta}_{\text{integrates to a constant}}} \quad \boxed{\propto \theta_a^{N_a}\theta_b^{N_b}(1-\theta_a-\theta_b)^{(N-N_a-N_b)}}\ ✓$$

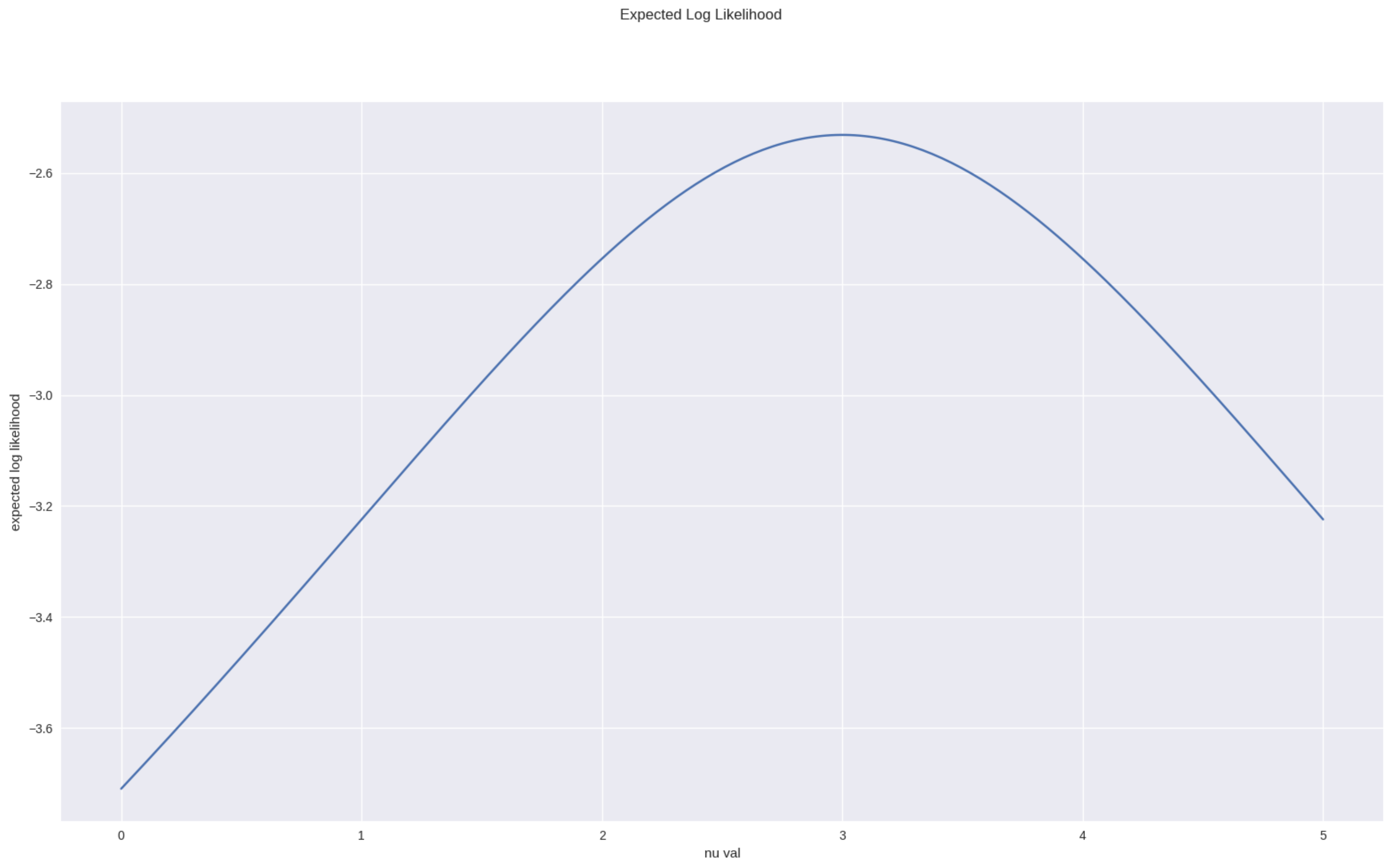
```python
"""Problem 4."""
import matplotlib as mpl
import matplotlib.pyplot as plt

import numpy as np

import scipy.integrate as integrate
import scipy.io as sio

mpl.style.use('seaborn')

MAT4A_FILENAME = 'hw09p4a.mat'
data4a_samples = sio.loadmat(MAT4A_FILENAME)
x4a = data4a_samples['x'].flatten()


def cauchy_density(x_val, nu_val):
    """Cauchy Density function."""
    return 1/(np.pi * (1 + (x_val - nu_val)**2))


def log_likelihood(x_sample, nu_val):
    """Compute log likelihood of Cauchy distribution."""
    if isinstance(x_sample, float):
        # For when sample size = 1
        x_val = x_sample
        return np.log(cauchy_density(x_val, nu_val))

    return sum(np.log(cauchy_density(x_val, nu_val)) for x_val in
               x_sample)


def compute_mle(x_sample, nu_tolerance):
    """Approximate mle of Cauchy distribution."""
    nu_vec = np.linspace(0, 5, int(5/nu_tolerance))
    log_likelihood_vec = log_likelihood(x_sample, nu_vec)
    nu_mle_index = np.argmax(log_likelihood_vec)
    nu_mle = nu_vec[nu_mle_index]
    return log_likelihood_vec, nu_vec, nu_mle


def part_a(x_sample):
    """Part a."""
    print('Part a')
    nu_tolerance = 1e-4

    log_likelihood_vec, nu_vec, nu_mle = compute_mle(x_sample, nu_tolerance)

    fig = plt.figure()
    fig.suptitle('Log Likelihood')
    axes = fig.add_subplot(111)

    axes.plot(nu_vec, log_likelihood_vec)

    axes.set_xlabel('nu val')
    axes.set_ylabel('log likelihood')

    plt.show()

    print('MLE nu: ' + str(nu_mle))


part_a(x4a)
```

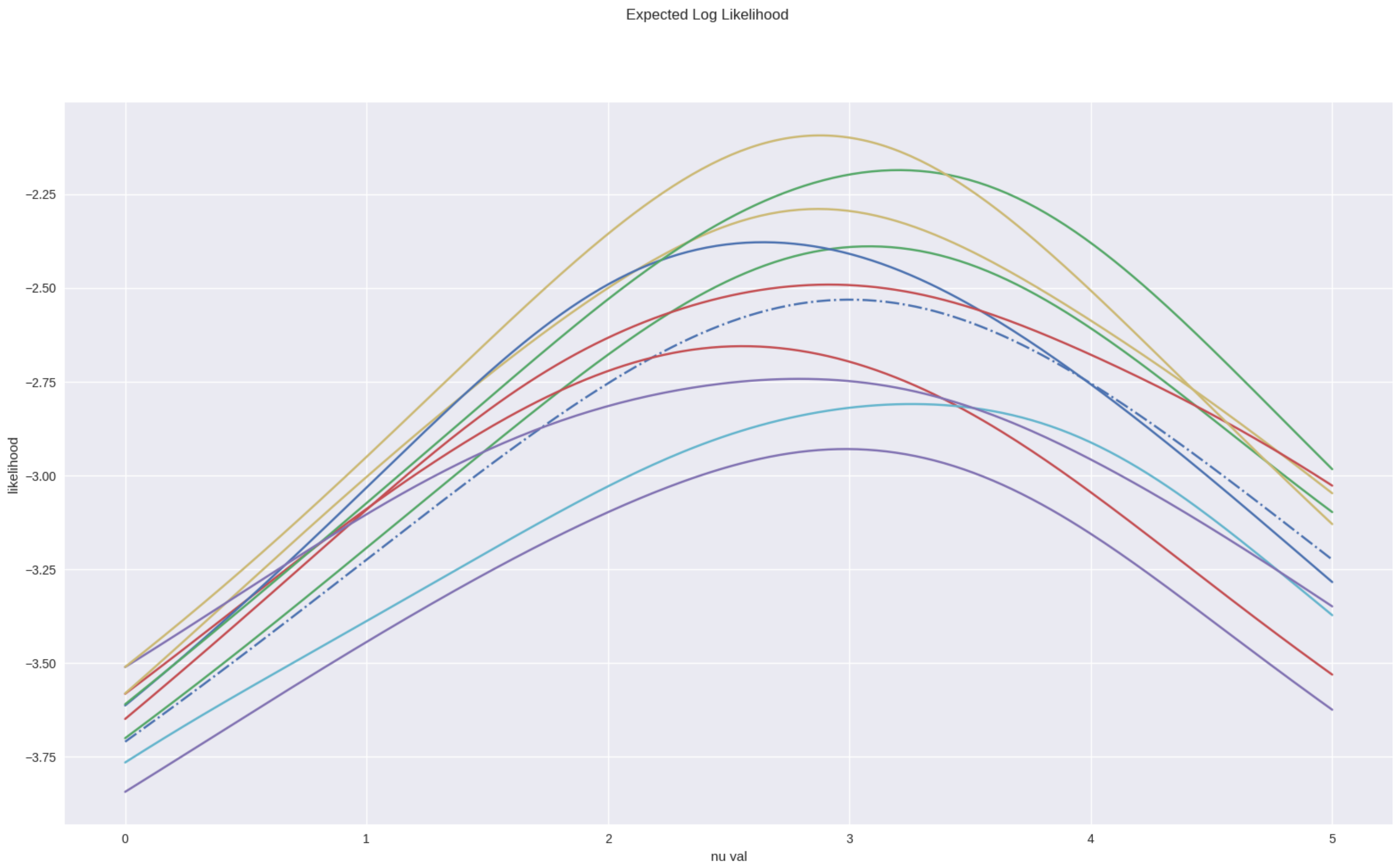
```
MAT4B_FILENAME = 'hw09p4b.mat'
data4b_samples = sio.loadmat(MAT4B_FILENAME)
x4b = data4b_samples['X']


def part_b(x_samples):
    """Part b."""
    print('Part b')
    nu_o = 3
    nu_tolerance = 1e-2
    trials = x_samples.shape[1]

    def sample_mean(x_sample):
        return 1/x_sample.shape[0] * sum(x_sample)

    def sample_median(x_sample):
        # order the sample
        x_sample = np.sort(x_sample)
        sample_size = x_sample.shape[0]
        if x_sample.shape[0] % 2 == 1:
            return x_sample[sample_size//2]
        return (x_sample[sample_size//2] + x_sample[sample_size//2 - 1]) / 2

    sample_mean_vec = [sample_mean(x_samples[:, trial])
                       for trial in range(trials)]

    sample_median_vec = [sample_median(x_samples[:, trial])
                         for trial in range(trials)]

    mle_vec = [compute_mle(x_samples[:, trial], nu_tolerance)[2]
               for trial in range(trials)]

    def emse(nu_o, nu_hat_vec):
        return 1/len(nu_hat_vec) * sum((nu_hat - nu_o)**2
                                       for nu_hat in nu_hat_vec)

    sample_mean_emse = emse(nu_o, sample_mean_vec)
    sample_median_emse = emse(nu_o, sample_median_vec)
    mle_emse = emse(nu_o, mle_vec)

    print('Empirical Mean Squared Error (EMSE)')
    print('Sample Mean EMSE: ' + str(sample_mean_emse))
    print('Sample Median EMSE: ' + str(sample_median_emse))
    print('MLE EMSE: ' + str(mle_emse))


part_b(x4b)


def part_c():
    """Part c."""
    print('Part c')

    def expected_log_likelihood(x_val, nu_val, nu_o):
        return cauchy_density(x_val, nu_o) * log_likelihood(x_val, nu_val)

    nu_o = 3
    nu_vec = np.linspace(0, 5, 250)

    expected_log_likelihood_vec = [integrate.quad(expected_log_likelihood,
                                                  -np.Inf, np.Inf,
                                                  args=(nu_val, nu_o))[0]
                                   for nu_val in nu_vec]
```

```
    fig = plt.figure()
    fig.suptitle('Expected Log Likelihood')
    axes = fig.add_subplot(111)

    axes.plot(nu_vec, expected_log_likelihood_vec)

    axes.set_xlabel('nu val')
    axes.set_ylabel('expected log likelihood')

    plt.show()

    return expected_log_likelihood_vec


expected_log_likelihood_vec_part_c = part_c()


def part_d(x_samples, expected_log_likelihood_vec):
    """Part d."""
    print('Part d')

    sample_size = x_samples.shape[0]
    nu_vec = np.linspace(0, 5, 250)

    fig = plt.figure()
    fig.suptitle('Expected Log Likelihood')
    axes = fig.add_subplot(111)

    axes.plot(nu_vec, expected_log_likelihood_vec, linestyle='-.')

    for sample_index in range(10):
        x_sample = x_samples[:, sample_index]
        norm_log_likelihood_vec = 1/sample_size * \
            log_likelihood(x_sample, nu_vec)
        axes.plot(nu_vec, norm_log_likelihood_vec)

    axes.set_xlabel('nu val')
    axes.set_ylabel('likelihood')

    plt.show()


part_d(x4b, expected_log_likelihood_vec_part_c)
```

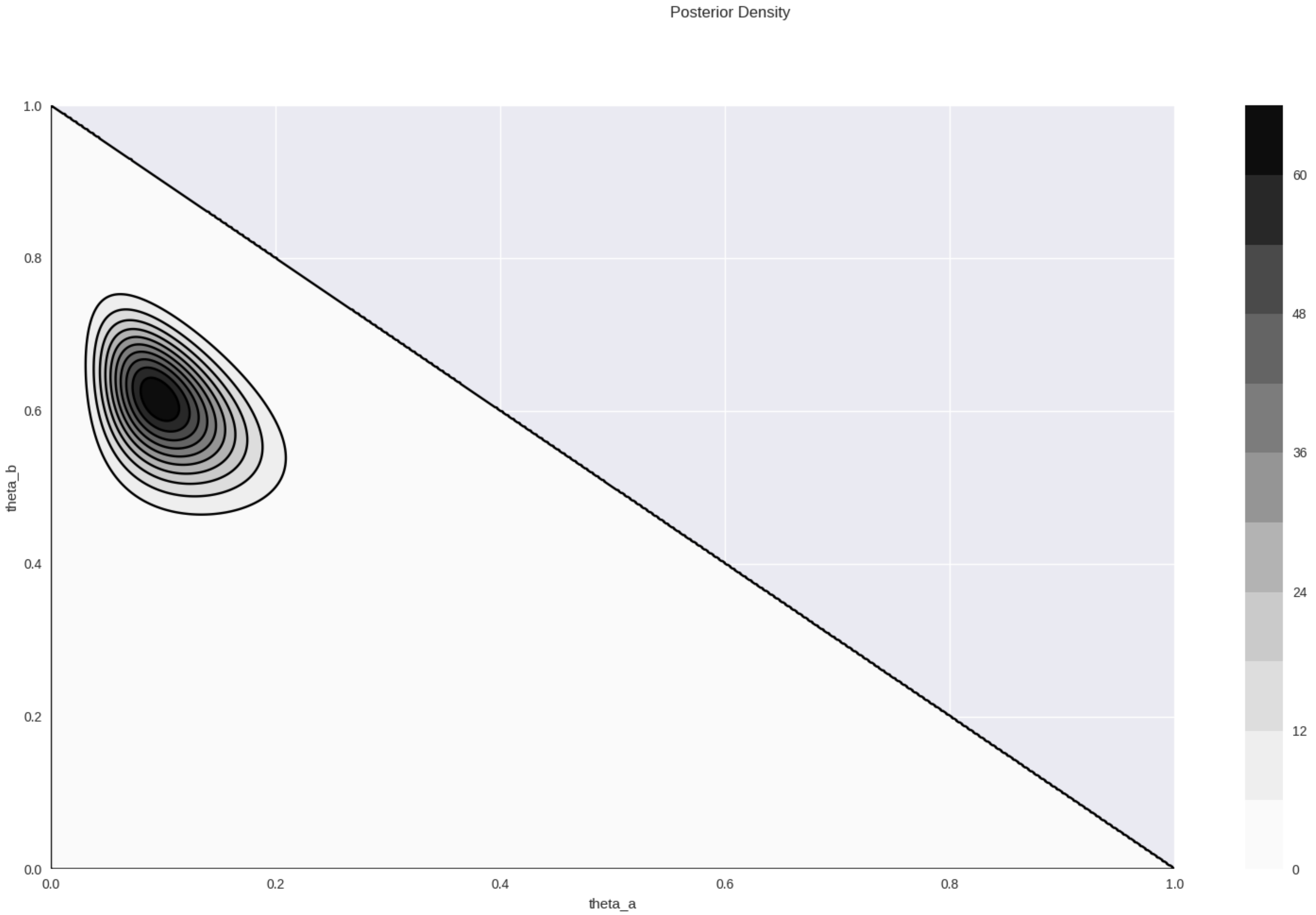
```
Part a
MLE nu: 1.4743294865897316
Part b
Empirical Mean Squared Error (EMSE)
Sample Mean EMSE: 1411.150288215387
Sample Median EMSE: 0.050116063063892664
MLE EMSE: 0.04039323135248446
Part c
Part d
```


Log Likelihood
log likelihood
−140
−160
−180
−200
0
1
2
3
4
5
nu val


Expected Log Likelihood
expected log likelihood
nu val
−2.6
−2.8
−3.0
−3.2
−3.4
−3.6
0
1
2
3
4
5


Expected Log Likelihood
likelihood
−2.25
−2.50
−2.75
−3.00
−3.25
−3.50
−3.75
0
1
2
3
4
5
nu val

```python
"""Problem 5."""
import matplotlib as mpl
import matplotlib.pyplot as plt

import numpy as np

from scipy.special import gamma

mpl.style.use('seaborn')


def dirichlet_distribution(theta_a, theta_b, n_val, na_val, nb_val):
    """Dirichlet Distribution."""
    constant = gamma(n_val + 3)/(gamma(na_val + 1) * gamma(nb_val + 1) *
                                 gamma(n_val - na_val - nb_val + 1))
    return constant * theta_a**na_val * theta_b**nb_val * \
        (1 - theta_a - theta_b)**(n_val - na_val - nb_val)


def part_b():
    """Part b."""
    print('Part b')

    a_wins = 5
    b_wins = 32
    c_wins = 15

    weeks = a_wins + b_wins + c_wins

    num_values = 1000
    theta_a_vec = np.linspace(0, 1, num_values)
    theta_b_vec = np.linspace(0, 1, num_values)

    theta_a_mat, theta_b_mat = np.meshgrid(theta_a_vec, theta_b_vec)
    dirichlet_mat = dirichlet_distribution(theta_a_mat, theta_b_mat, weeks,
                                           a_wins, b_wins)

    mask = theta_a_mat + theta_b_mat > 1
    dirichlet_mat[mask] = 0

    fig = plt.figure()
    fig.suptitle('Posterior Density')
    axes = fig.add_subplot(111)

    csetf = axes.contourf(theta_a_mat, theta_b_mat, dirichlet_mat, levels=10)
    axes.contour(theta_a_mat, theta_b_mat, dirichlet_mat, csetf.levels,
                 colors='k')

    fig.colorbar(csetf, ax=axes)
    axes.set_xlabel('theta_a')
    axes.set_ylabel('theta_b')

    plt.show()


part_b()
```



Posterior Density
1.0
0.8
0.6
0.4
0.2
0.0
theta_b
0.0
0.2
0.4
0.6
0.8
1.0
theta_a
60
48
36
24
12
0