BAYESIAN MACHINE LEARNING PROJECT

EMTIAZ SAMAD

MASTER OF SCIENCE IN DATA SCIENCE
THE UNIVERSITY OF BATH
2021-2022

CONTENTS

1 EXPLORATORY ANALYSIS……………………………………………………….3
2 BAYESIAN LINEAR REGRESSION……………………………………………….4
2.1 TYPE-II MAXIMUM LIKELIHOOD………………………………………….4
2.1 VARIATIONAL INFERENCE……………………………………………...5
3 VERIFYING HMC ON A STANDARD 2D GAUSSIAN EXAMPLE ………………..7
4 APPLYING HMC TO THE LINEAR REGRESSION MODEL ………………………9
APPENDIX……………………………………………………………………..11

1 Exploratory Analysis

The data set used in this project contains data on energy efficiency as is distributed by the University of Oxford and is available for download from the UCI Machine Learning Repository. The dataset consists of 768 exemplars consisting of a constant bias x_0 and the input variables $x_1, x_2, \ldots, x_8$. For convenience, the data has been standardised such that the inputs have a mean of 0 and standard deviation of 1. Furthermore, the data has been equally split into training and test sets where the test set is purely reserved for assessing model performance.

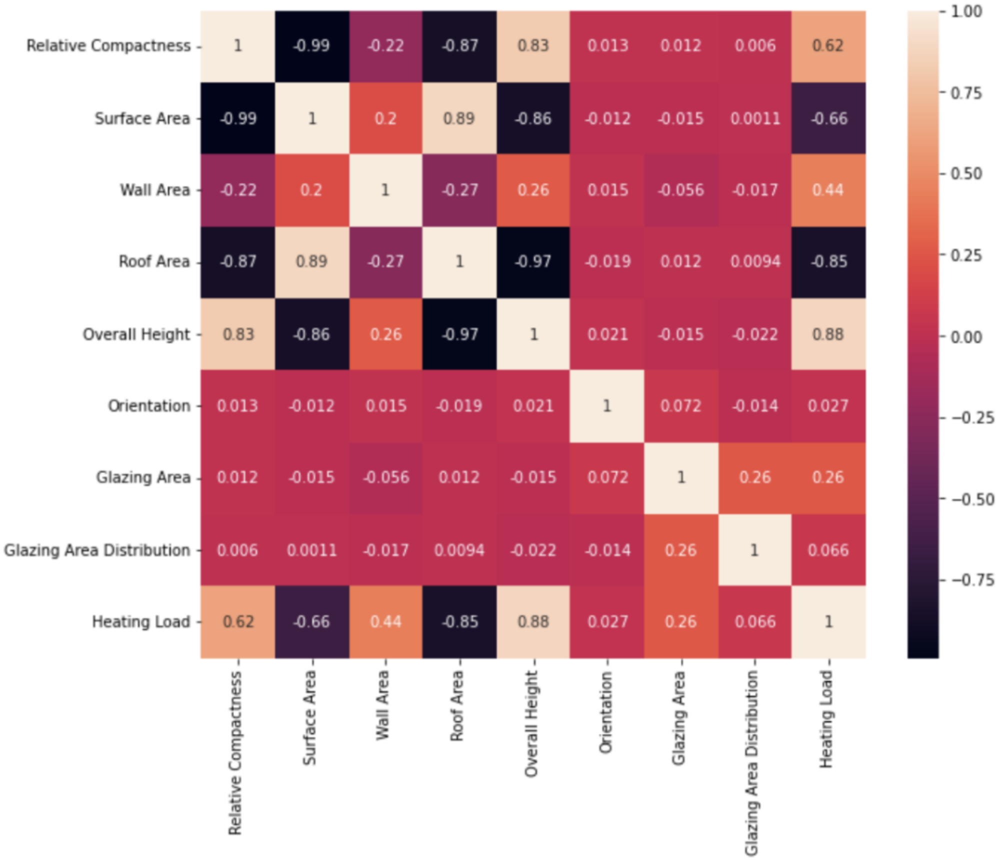


Figure 1 Correlation plot of all features of the dataset.

From observing *Figure 1*, it is evident that *Heating Load* is strongly correlated with *Overall Height* and somewhat correlated with *Relative Compactness*. Additionally, *Heating Load* has a strong negative correlation with *Roof Area*. Features such as *Orientation*, *Glazing Area* and *Glazing Area Distribution* have almost no correlation with *Heating Load*.

2 Bayesian Linear Regression

In this task we estimated the most probable values for the hyper parameters using *Type-II Maximum Likelihood* and *Variational Inference* with *Mean-Field Theory* in a standard Bayesian linear regression model. In this model:

- $\boldsymbol{w}$ is assumed to have a Gaussian prior $\mathcal{N}(0, \sigma_{\boldsymbol{w}}^2)$ such that the precision of the prior is defined as $\alpha = \frac{1}{\sigma_w^2}$

- The problem is modelled with an additive Gaussian noise $\mathcal{N}(0, \sigma_\epsilon^2)$ such that the precision of the noise is defined as $\beta = \frac{1}{\sigma_\epsilon^2}$

- Thus, the hyperparameter space can be defined as $\theta = (\sigma_\epsilon^2, \sigma_{\boldsymbol{w}}^2) = (\frac{1}{\alpha}, \frac{1}{\beta})$

2.1 Type-II Maximum Likelihood

In this subtask, we estimated the most probable values for θ using Type-II Maximum Likelihood. This involved creating a likelihood function to calculate the log of the posterior $p(\boldsymbol{w}, \alpha, \beta | X, y)$. We computed the log evidence from 100 samples of α and β from in the range (-5, 0) to create a contour plot demonstrating log-posterior distribution.

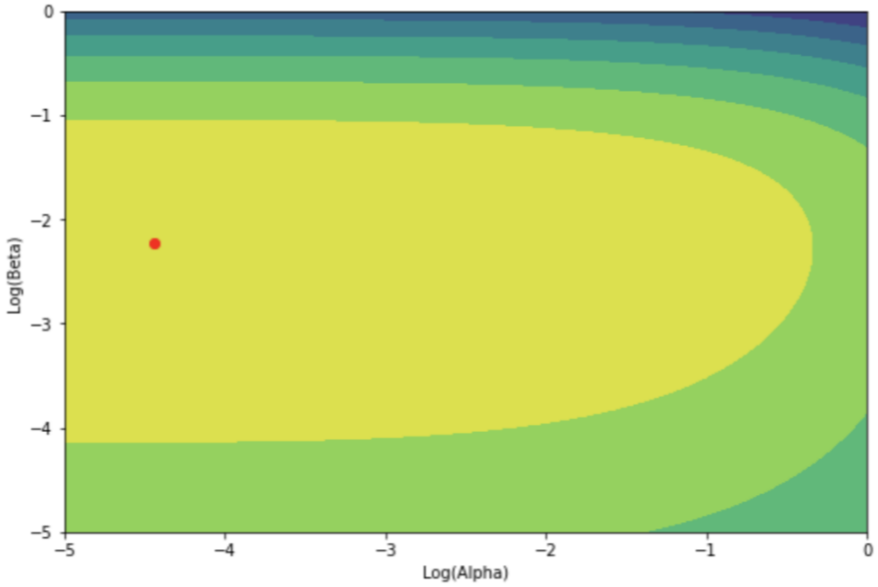


Figure 2 Log-Posterior Distribution containing the most probable values of the hyperparameters marked in red.

Most Probable $\boldsymbol{\alpha}$	0.01174
Most Probable $\boldsymbol{\beta}$	0.10837
Corresponding log-likelihood	-1001.458
Training RMSE	3.0117
Testing RMSE	3.0926

Figure 3 Table containing the most probable hyperparameters, the corresponding log-likelihood and the RMSE of the training and test data using such hyperparameters.

2.2 VARIATIONAL INFERENCE

The mathematical equations used to implement Variational Inference can were derived as follows:

- The posterior weights were updated using the formulas directly below:

$$\boldsymbol{\mu}_N = \frac{\boldsymbol{\Sigma}_N \boldsymbol{\Phi}^T \boldsymbol{t}}{\sigma^2}$$

$$\boldsymbol{\Sigma}_N = \left(\frac{\boldsymbol{\Phi}^T \boldsymbol{\Phi}}{\sigma^2} + \langle \alpha \rangle_{Q_\alpha(\alpha)} \boldsymbol{I} \right)^{-1}$$

- The a_N and b_N values to calculate α were updated, such that:

$$a_N = a_0 + \frac{M}{2}, \quad b_N = b_0 + \frac{1}{2} \langle \boldsymbol{w}^T \boldsymbol{w} \rangle_{Q_w(\boldsymbol{w})}$$

$$\langle \alpha \rangle_{Q_\alpha(\alpha)} = \frac{a_N}{b_N}, \quad \langle \boldsymbol{w}^T \boldsymbol{w} \rangle_{Q_w(\boldsymbol{w})} = \boldsymbol{\mu}_N{}^T \boldsymbol{\mu}_N + tr(\boldsymbol{\Sigma}_N)$$

- The c_N and d_N values to calculate β were updated, such that:

$$c_N = c_0 + \frac{N}{2}, \quad d_N = d_0 + \frac{1}{2} (X\boldsymbol{w} - y)^T (X\boldsymbol{w} - y)$$

In this subtask, we estimated the most probable values for θ using Variational Inference. We computed the log evidence from 100 samples of α and β from in the range (-5, 0) to create a contour plot demonstrating log-posterior distribution.

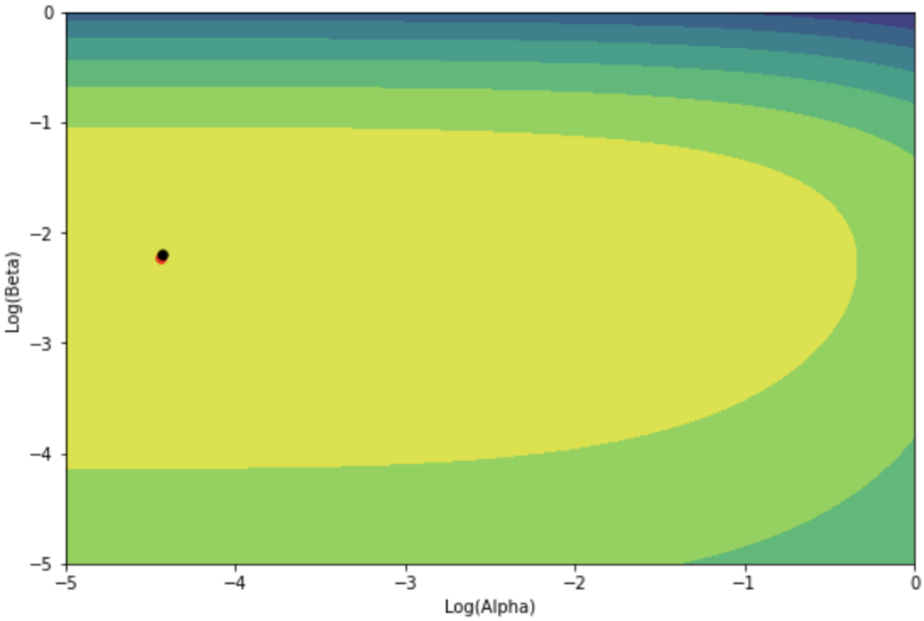


Figure 4 Log-Posterior Distribution containing the most probable values of the hyperparameters marked in black.

Most Probable $\boldsymbol{\alpha}$	0.01192439250898537
Most Probable $\boldsymbol{\beta}$	0.11024997373399267
Training RMSE	3.0116940662643006
Testing RMSE	3.092588870671393

Figure 5 Table containing the most probable hyperparameters, the corresponding log-likelihood and the RMSE of the training and test data using such hyperparameters.

3 VERIFYING HMC ON A STANDARD 2D GAUSSIAN EXAMPLE

In task 4, we formulated the energy function by taking the negative log of the Bivariate Gaussian probability density function as follows:

$$p(\alpha, \beta) = \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-(\frac{\alpha^2 + \beta - 2\alpha\beta\rho}{2(1-\rho^2)})},$$

with the covariance matrix:

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho & \sigma_y^2 \end{bmatrix}$$

Thus, the energy function is:

$$-\log \rho(\alpha, \beta) = \log\left(2\pi\sqrt{1-\rho^2}\right) + \frac{\alpha^2 + \beta^2 - 2\alpha\beta\rho}{2(1-\rho^2)}$$

The corresponding gradients were calculated by taking the respective partial derivatives as follows:

- $\frac{\partial p}{\partial \alpha} = \frac{\alpha - \beta\rho}{1-\rho^2}$
- $\frac{\partial p}{\partial \beta} = \frac{\beta - \alpha\rho}{1-\rho^2}$

Note that we implemented the code for the gradient function from scratch despite having used *scipy.stats.multivariate_norm.logpdf* to obtain identical results. The code for the energy and gradient functions can be found in the appendix.

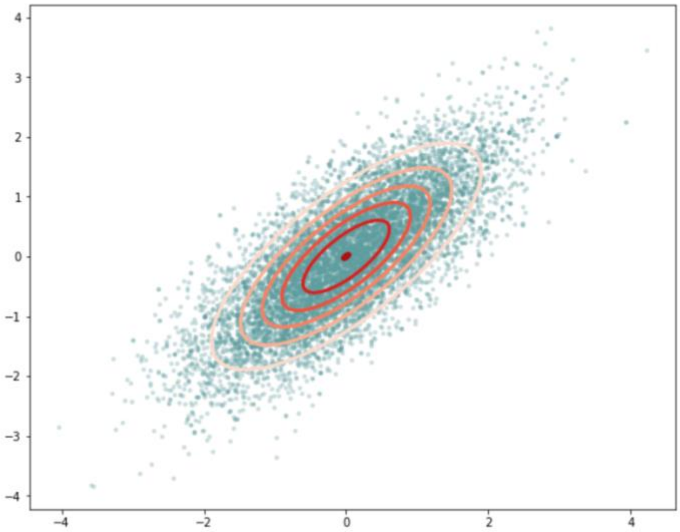
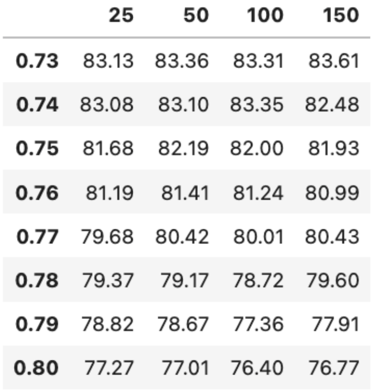


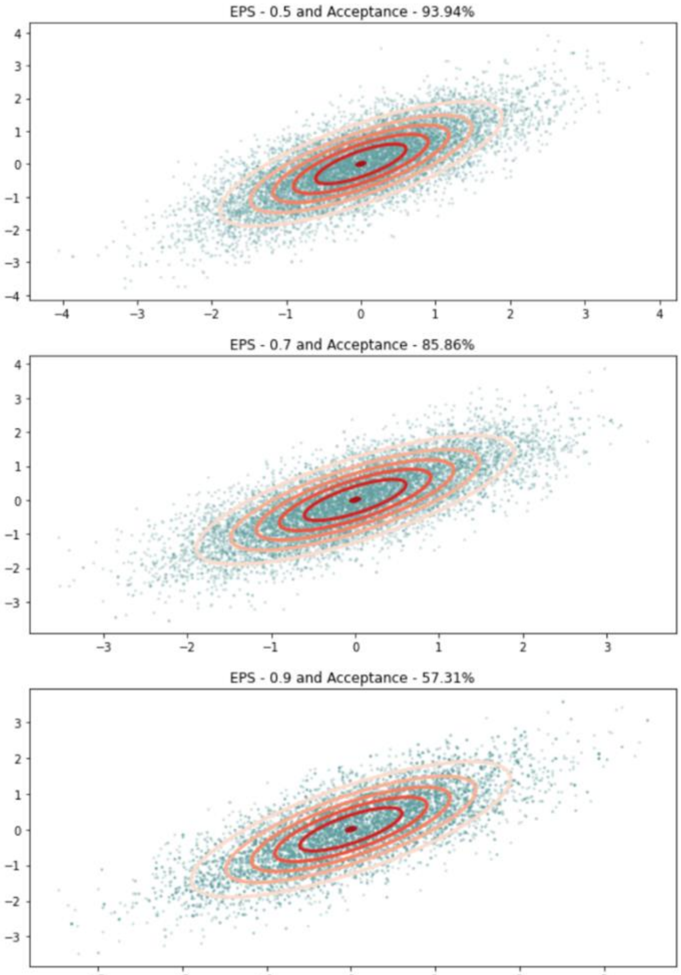
Figure 6 Application of HMC on a 2D Multivariate Gaussian with parameters L=25, R=10000 and eps=0.77, yielding an acceptance of 80%.

From *Figure 6*, it is clear that the HMC algorithm has been correctly implemented as required. Using the parameters L = 25, R = 10000 and eps = 0.77, the algorithm produced an acceptance rate of 80%. This aligns with our methodology in choosing suitable parameters L, R and eps such that eps is maximised to give an acceptance rate of 80% and then reduced slightly if desired. This is to ensure that the step length is large enough to explore the state space efficiently. We performed a grid search to identify the optimal parameters of eps and L resulting in an acceptance of 80% and stored the information in a table displayed below.



	25	50	100	150
0.73	83.13	83.36	83.31	83.61
0.74	83.08	83.10	83.35	82.48
0.75	81.68	82.19	82.00	81.93
0.76	81.19	81.41	81.24	80.99
0.77	79.68	80.42	80.01	80.43
0.78	79.37	79.17	78.72	79.60
0.79	78.82	78.67	77.36	77.91
0.80	77.27	77.01	76.40	76.77

Figure 7 Table containing the acceptance rates of varying eps and L in the 2D Multivariate Gaussian case.



EPS - 0.5 and Acceptance - 93.94%

EPS - 0.7 and Acceptance - 85.86%

EPS - 0.9 and Acceptance - 57.31%

Figure 8 Plots demonstrating varying eps and the corresponding acceptance rates.

4 APPLYING HMC TO THE LINEAR REGRESSION MODEL

Task 5 requires the estimation of the parameters $\alpha, \beta\ and\ \boldsymbol{w}$ by using the HMC algorithm. This requires solving the Gaussian posterior given by

$$p(\boldsymbol{w}, \alpha, \beta | X, y) = \frac{p(y|X, \boldsymbol{w}, \beta) p(\boldsymbol{w}|\alpha) p(\alpha) p(\beta)}{p(X, y)},$$

such that $p(\alpha)\ and\ p(\beta)$ have uniform distribution and can be considered constant, thus

$$p(\boldsymbol{w}, \alpha, \beta | X, y) \propto p(y|X, \boldsymbol{w}, \beta) p(\boldsymbol{w}|\alpha),$$

where:

- $p(y|X, \boldsymbol{w}, \beta)$ represents the likelihood
- $p(\boldsymbol{w}|\alpha)$ represents the prior

Therefore, the energy function can be calculated by taking the negative log of the posterior as

$$-\log p(\boldsymbol{w}, \alpha, \beta | X, y) = -\log p(y|X, \boldsymbol{w}, \beta) - \log p(\boldsymbol{w}|\alpha).$$

The likelihood is given by the following equation:

$$p(y|X, \boldsymbol{w}, \beta) = \left(\frac{1}{\sqrt{2\pi(\frac{1}{\beta})}} \right)^{N} e^{-\frac{\|X\boldsymbol{w}-y\|^2}{2(\frac{1}{\beta})}}$$

$$= \left(\frac{\beta}{2\pi} \right)^{\frac{N}{2}} e^{-\frac{\beta\|X\boldsymbol{w}-y\|^2}{2}}.$$

Taking the log of the likelihood, we obtain:

$$\log p(y|X, \boldsymbol{w}, \beta) = \frac{N}{2}(\log \beta - \log 2\pi) - \frac{\beta}{2}(X\boldsymbol{w} - y)^T (X\boldsymbol{w} - y).$$

Similarly, the prior is given by the following equation:

$$p(\boldsymbol{w}|\alpha) = \prod_{m=1}^{M} (\frac{\alpha}{2\pi})^{\frac{1}{2}} e^{-\frac{\alpha}{2} w_m^2}$$

$$= (\frac{\alpha}{2\pi})^{\frac{M}{2}} e^{-\frac{\alpha}{2} \sum_{m=1}^{M} w_m^2}.$$

Taking the log of the prior, we obtain:

$$\log p(\boldsymbol{w}|\alpha) = \frac{M}{2}(\log \alpha - \log 2\pi) - \frac{\alpha}{2} \boldsymbol{w}^T \boldsymbol{w}.$$

Combining the equations for the likelihood and the prior, we calculate the negative log posterior as:

$$-log\ p(\boldsymbol{w}, \alpha, \beta | X, y) = \frac{\beta}{2}(X\boldsymbol{w} - y)^T (X\boldsymbol{w} - y) - \frac{N}{2}(\log \beta) + \frac{\alpha}{2} \boldsymbol{w}^T \boldsymbol{w} - \frac{M}{2}(\log \alpha) + 2 \log 2\pi.$$

The corresponding gradients can be calculated by taking the respective partial derivatives as follows:

- $\frac{\partial p}{\partial \alpha} = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \frac{M}{2\alpha}$
- $\frac{\partial p}{\partial \beta} = \frac{1}{2}(X\boldsymbol{w} - y)^T(X\boldsymbol{w} - y) - \frac{N}{2\beta}$
- $\frac{\partial p}{\partial w} = \beta X^T X\boldsymbol{w} - \beta X^T y + \alpha\boldsymbol{w}$

From the results attained, we can confirm that the HMC implementation works as required and achieves an appropriate acceptance rate of approximately 90.5%.

The values of L, R and eps were chosen according to the same methodology applied in the previous task. Essentially, this involved setting the number of steps L at 20 (as recommended) and increasing eps to be as large as possible without having the acceptance rate drop below 80%, then slightly lowered. As previously mentioned, an extremely high acceptance rate would not be favourable as it may indicate that the step length is too small, so the state space is not efficiently explored. In accordance with the aforementioned method, we chose to set the parameters L, R and eps as 25, 10000 and 0.0132 respectively. This yielded an acceptance rate of 90.5%.

Training RMSE	3.068295840964306
Testing RMSE	3.0549193122824296

Figure 9 Table containing the RMSE of the training and test data using the most probable hyperparameters.

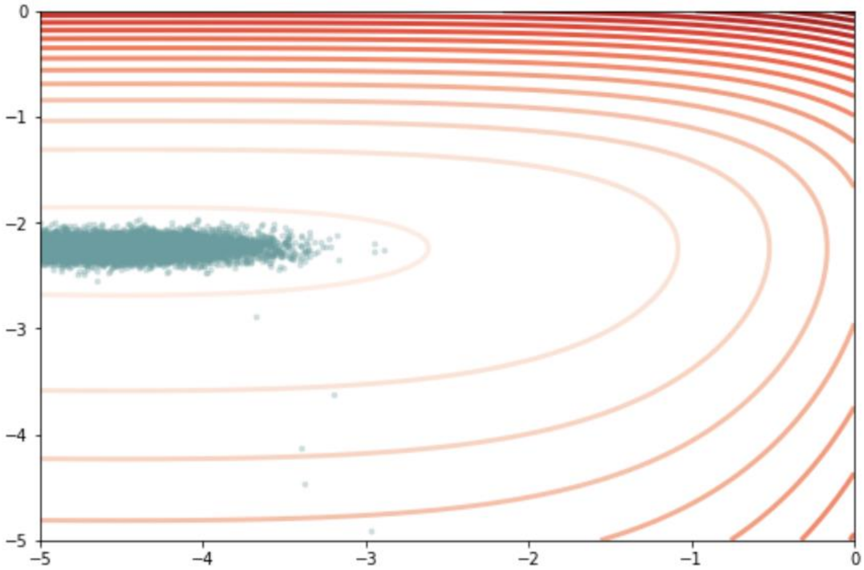


Figure 10 Application of HMC on a Linear Regression model with parameters L=25, R=10000 and eps=0.0132, yielding an acceptance of 90.5%.

APPENDIX

CODE

2D MULTIVARIATE GAUSSIAN HMC ENERGY AND GRADIENT FUNCTIONS

```
def energy_func(x, covar):
    neglgp = -stats.multivariate_normal.logpdf(x = x, cov = covar)
    return neglgp

def energy_grad(x, covar):
    #initialise array
    g = np.empty(2)
    #calc p
    p = (covar[0][1])/(((covar[0][0])*(covar[1][1])))
    # d/da
    g[0] = (x[0] - x[1]*p)/(1-p**2)
    # d/db
    g[1] = (x[1] - x[0]*p)/(1-p**2)
    return g
```

LINEAR REGRESSION HMC ENERGY AND GRADIENT FUNCTIONS

```
def energy_func_lr(hps, x, y):
    alpha = hps[0]
    beta = hps[1]
    w = hps[2:]

    log_likelihood = (1/2) * (x.shape[0] * np.log(np.exp(beta)/(2)) - (y -
x @ w).T @ (y - x @ w) * np.exp(beta))
    log_prior = (1/2) * (x.shape[1] * np.log(np.exp(alpha)/(2)) - (w.T @
w)*np.exp(alpha))
    neglgp = -(log_likelihood + log_prior)
    return neglgp

def energy_grad_lr(hps, x, y):
    g = np.empty(11)
    alpha = hps[0]
    beta = hps[1]
    w = hps[2:]

    # Alpha Partial
    g[0] = (1/2)*(np.exp(alpha)* w @ w.T - x.shape[1])
    # Beta Partial
    g[1] = (1/2)*(np.exp(beta)*sum((y - x @ w)**2) - x.shape[0])
    # W partial
    g[2:] = np.exp(beta)*(x.T @ x @ w - x.T @ y) + np.exp(alpha) * w
    return g
```