

General guidelines:

- All solutions to theoretical and practical problems must be submitted in this jupyter notebook, and equations wherever required, should be formatted using LaTeX math-mode.
- All discussion regarding practical problems, along with solutions and plots should be specified in this notebook. All plots/results should be visible such that the notebook do not have to be run. But the code in the notebook should reproduce the plots/results if we choose to do so.
- Your name, personal number and email address should be specified above.
- All tables and other additional information should be included in this notebook.
- Before submitting, make sure that your code can run on another computer. That all plots can show on another computer including all your writing. It is good to check if your code can run here: <https://colab.research.google.com>
- Upload both the .ipynb and the generatable .html file (that can be exported through File > Download As > HTML)

Jupyter/Python Notebook is a collaborative Python web-based environment. This will be used in all our Homework Assignments. It is installed in the halls ES61-ES62, E-studio and MT9. You can also use google-colab: <https://colab.research.google.com> to run these notebooks without having to download, install, or do anything on your own computer other than a browser. Some useful resources:

- <https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/> (Quick-start guide)
- <https://www.kdnuggets.com/2016/04/top-10-python-nb-tutorials.html>
- <http://data-blog.udacity.com/posts/2016/10/latex-primer/> (latex-primer)
- <http://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Working%20With%20Markdown%20Cells.html> (markdown)

## Theoretical problems

### [Bayes Rule, 3 points]

After your yearly checkup, the doctor has bad news and good news. The bad news is that you tested positive for a very serious cancer and that the test is 99.2% accurate i.e. the probability of testing positive given you have the disease is 0.992. The probability of testing negative if you don't have the disease is the same (also 0.992). The good news is that it is a rare condition affecting only 1 in 1,000 people. What is the probability you actually have the disease?

After doing all your calculations you realize that there was a misprint on the test, and the accuracy was actually only 92% (for both testing positive given that you have the disease and for testing negative given that you do not have the disease). How will this change your probability of having the disease?

(Show all calculations and the final result.)

### [Setting hyperparameters, 2 points]

Suppose  $\theta$  is a random variable generated from a beta distribution as:  $\theta \sim \text{Beta}(a^2, b)$ . Also assume that the expectation of  $\theta$  is  $m$ :  $E[\theta] = m$  and the variance of  $\theta$  is  $v$ :  $\text{var}(\theta) = v$ . Express  $a$  and  $b$  in term of (only)  $m$  and  $v$ . For more information about the Beta distribution see [https://en.wikipedia.org/wiki/Beta\\_distribution](https://en.wikipedia.org/wiki/Beta_distribution).

### [Correlation and Independence, 2 points]

Let  $X$  be a continuous random variable, uniformly distributed in  $[-2, +2]$  and let  $Y := X^4$ . Clearly  $Y$  is not independent of  $X$  -- in fact it is uniquely determined by  $X$ . However, show that the covariance of  $X$  and  $Y$  is 0:  $\text{cov}(X, Y) = 0$ . Show and justify every step of the proof. Statements like "it is obvious that, it is trivial ..." will not be accepted.

### [Spherical Gaussian estimation, 4 points]

Consider a dataset  $X$  consisting of i.i.d. observations generated from a spherical Gaussian distribution  $N(\mu, \sigma^2 I)$ , where  $\mu \in \mathbb{R}^D$ ,  $I$  is the  $p \times p$  identity matrix, and  $\sigma^2$  is a scalar.

Write the mathematical expression for the Maximum Likelihood Estimator (MLE) for  $\mu$  and  $\sigma$  in above setup.

For more information about the spherical Gaussian distribution, see [https://en.wikipedia.org/wiki/Multivariate\\_normal\\_distribution](https://en.wikipedia.org/wiki/Multivariate_normal_distribution). For more information about the identity matrix see: [https://en.wikipedia.org/wiki/Identity\\_matrix](https://en.wikipedia.org/wiki/Identity_matrix)

Write Answers here. Double-click (or enter) to edit. Latex mathematical expressions can just be written by starting with "\$" and ending with the same character.

### Bayes Rules

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

What is the probability to have the disease with a test accuracy of 99.2% ?

$$P(T^+|D) = 0.992$$

$$P(T^-|D) = 0.992$$

$$P(D) = \frac{1}{1000}$$

$$P(D^{-1}) = 0.999$$

$$P(T^+|D^{-1}) = 1 - 0.992$$

$$= 0.008$$

$$P(D|T^+) = \frac{P(T^+|D)P(D)}{P(T^+)}$$

$$P(T^+) = P(T^+|D)P(D) + P(T^+|D^{-1})P(D^{-1})$$

$$= (0.992 \times 0.001) + (0.008 \times 0.999)$$

$$P(D|T^+) = \frac{(0.992 \times 0.001)}{0.992 \times 0.001 + (0.008 \times 0.999)}$$

$$P(D|T^+) = 0.11041 = \boxed{11.04\%}$$

What is the probability to have the disease with a test accuracy of 92% ?

$$P(T^+|D) = 0.92$$

$$P(T^-|D) = 0.92$$

$$P(D) = \frac{1}{1000}$$

$$P(D^{-1}) = 0.999$$

$$P(T^+|D^{-1}) = 0.08$$

$$P(D|T^+) = \frac{P(T^+|D)P(D)}{P(T^+|D)P(D) + P(T^+|D^{-1})P(D^{-1})}$$

$$P(D|T^+) = \frac{(0.92 \times 0.001)}{0.92 \times 0.001 + (0.08 \times 0.999)}$$

$$P(D|T^+) = 0.0113 = \boxed{1.13\%}$$

### Setting hyperparameters

$$\theta \sim \text{Beta}(a^2, b)$$

Express  $a$  and  $b$  in term of (only)  $m$  and  $v$

$$E[\theta] = m$$

$$\text{var}[\theta] = v$$

$$E[\theta] = \frac{a^2}{a^2+b} = m$$

$$\text{var}[\theta] = \frac{a^2b}{(a^2+b)^2(a^2+b+1)}$$

$$\frac{a^2}{a^2+b} = m \Rightarrow a^2 + b = \frac{a^2}{m}$$

$$b = \frac{a^2}{m} - a^2 = a^2\left(\frac{1}{m} - 1\right)$$

$$\text{var}[\theta] = \frac{a^2b}{(a^2+b)^2(a^2+b+1)} = \frac{a^2\left(\frac{1}{m} - a^2\right)}{(a^2\left(\frac{1}{m} - a^2\right))^2(a^2\left(\frac{1}{m} - a^2\right) + 1)}$$

$$\text{var}[\theta] = v = \frac{a^2\left(\frac{1}{m} - a^2\right)}{\left(\frac{1}{m}\right)^2\left(\frac{1}{m} + 1\right)}$$

$$\text{var}[\theta] = v = \frac{a^2\left(\frac{1}{m} - 1\right)}{\left(\frac{1}{m}\right)^2\left(\frac{1}{m} + 1\right)} = \frac{(1-m)m}{\left(\frac{1}{m}\right)^2 + 1}$$

$$\left(\frac{1}{m}\right)^2 + 1 = \frac{m-m^2}{v}$$

$$a^2 = \left(\frac{m^2-m^3}{v} - 1\right) = \frac{m^3(1-m)}{v} - 1$$

$$\beta = \left(\frac{m^3(1-m)}{v} - 1\right)\left(\frac{1-m}{m}\right) = \frac{m(1-m)^2}{v} - \frac{1-m}{m}$$

$$\boxed{b = (1-m)\left(\frac{m(1-m)}{v} - \frac{1}{m}\right)}$$

$$a^2 = \frac{m^3}{v}(1-m) - 1$$

$$\boxed{a = \pm m\sqrt{\frac{1-m}{v} - 1}}$$

### Correlation

Show that the covariance of  $X$  and  $Y$  is 0

$$E[x^m] = \int_{-\infty}^{\infty} x^m b(x) dx = \int_{-\infty}^{\infty} \frac{e^{-x^2}}{\sqrt{\pi}} dx$$

$$= \frac{e^{a+1}}{(b-a)(m+1)} = \frac{b^{a+1}-a^{a+1}}{(b-a)(m+1)}$$

In our case

$$(a = -2, b = 2)$$

$$\text{cov}[x, y] = \text{cov}[x, x^4], y = x^4$$

$$\text{cov}[x, y] = E[x^5] - E[x]E[y]$$

$$= E[x \cdot x^4] - E[x]E[x^4]$$

$$= E[x^5] - E[x]E[x^4]$$

$$E[x^m] = \frac{b^{a+1}-a^{a+1}}{(m+1)(b-a)} = \frac{2^{a+1}-(-2)^{a+1}}{(m+1)(b-a)}$$

$$E[x^m] = 0 \text{ for all odd values}$$

$$\therefore \text{cov}[x, y] = 0$$

### Spherical Gaussian

Write the mathematical expression for the Maximum Likelihood Estimator (MLE) for  $\mu$  and  $\sigma$  in above setup.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}, P = 1, 2, 3, \dots$$

Likelihood function

$$L = N(y|\mu, \sigma^2 I) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right]$$

$$L = N(X|\mu, \sigma^2 I) = \frac{\exp\left[-\frac{1}{2}(X-\mu)^T(\sigma^2 I)(X-\mu)\right]}{\sqrt{(2\pi)\sigma^2}}$$

$$\text{cov}[X] = \sigma^2 I$$

The correlation between the data elements is zero i.e., data elements are independent from each other because the covariance matrix is directly proportional to the identity matrix

$$L(X|\mu, \sigma^2 I) = \prod_{i=1}^p \left[ \frac{\exp\left[-\frac{1}{2}\left(\frac{X_i-\mu_i}{\sigma}\right)^2\right]}{\sqrt{(2\pi)\sigma^2}} \right]$$

$$= (2\pi\sigma^2)^{-\frac{p}{2}} \prod_{i=1}^p \left( \exp\left[-\frac{1}{2}\left(\frac{X_i-\mu_i}{\sigma}\right)^2\right] \right)$$

$$= (2\pi\sigma^2)^{-\frac{p}{2}} \left[ \frac{1}{2\pi\sigma^2} \sum_{i=1}^p \left( \frac{X_i-\mu_i}{\sigma} \right)^2 \right]$$

$$= (2\pi\sigma^2)^{-\frac{p}{2}} \exp\left[ -\frac{1}{2\sigma^2} \left( \sum_{i=1}^p X_i^2 - 2\mu \sum_{i=1}^p X_i + p\mu^2 \right) \right]$$

$$\log(L) = -\frac{p}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \left( \sum_{i=1}^p X_i^2 - 2\mu \sum_{i=1}^p X_i + p\mu^2 \right)$$

$$\frac{\partial \log(L)}{\partial \mu} = 0$$

$$= \frac{p}{2\sigma^2} (-2 \sum_{i=1}^p X_i + 2p\mu) = 0$$

$$= -\sum_{i=1}^p X_i + p\mu = 0$$

$$\boxed{\mu = \frac{\sum_{i=1}^p X_i}{p}}$$

$$\frac{\partial \log(L)}{\partial \sigma^2} = \frac{\partial}{\partial \sigma^2} \left[ -\frac{p}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^p (X_i - \mu)^2 \right] = 0$$

$$= -\frac{p}{2} \frac{2\pi}{2\pi\sigma^2} + \frac{1}{2\sigma^2} \sum_{i=1}^p (X_i - \mu)^2 = 0$$

$$\frac{\sum_{i=1}^p (X_i - \mu)^2}{\sigma^2} = p$$

$$\sigma^2 = \frac{\sum_{i=1}^p (X_i - \mu)^2}{p}$$

$$\mu_p = \mu \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_p = \frac{\sum_{i=1}^p X_i}{p} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_p$$

$$\text{cov}[X] = \sigma^2 I_p = \frac{\sum_{i=1}^p (X_i - \mu)^2}{p} \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}_{p \times p}$$

### Practical problems

### [Linear Regression with regularization 9pt]

You are newly recruited as a Data Scientist at a leading consultancy company in Gothenburg. Your first task at the job is to help the Swedish Public Health Agency (folkskolsmyndigheten) for predicting the diabetes progression of patients. Assume that you are given a dataset D of  $n$  patients with  $D = ((\mathbf{x}_i, y_i))_{i=1}^n$  where  $\mathbf{x}_i \in \mathbb{R}^D$  represents numerical features of each patients and  $y_i \in \mathbb{R}$  represent the numerical diabetes progression. One can also view the dataset D as a pair of matrices  $(\mathbf{X}, \mathbf{y})$  with  $\mathbf{X} \in \mathbb{R}^{n \times D}$  and  $\mathbf{y} \in \mathbb{R}^{n \times 1}$ .

Fresh with the lectures in the machine learning course at Chalmers, you would like to use a linear model to quickly perform the task. In order words, you would like to find a vector  $\mathbf{w} \in \mathbb{R}^{D+1}$  such that  $\mathbf{y} = \mathbf{X}\mathbf{w}$ . However, you have just read one of the most popular machine learning book and it argues that standard linear regression (for finding  $\mathbf{w}$ ) can lead to various problems such as non-uniqueness of the solution, overfitting, etc. As a result, you decided to add a penalty term called regularization to control the optimisation problem. More specifically, you want to solve for:  $\min_{\mathbf{w}} \mathcal{L}(\mathbf{w})$  where  $\mathcal{L}(\mathbf{w}) = \left( \sum_{i=1}^n (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \right) + \left( \alpha \sum_{j=1}^D w_j^2 \right)$  with  $\alpha \in \mathbb{R}$  a small coefficient that you will decide later on.

Please note the slight changes in the notation. Recall that in the lectures we had a dataset  $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$  where  $\mathbf{x}_n \in \mathbb{R}^D$  and  $t_n \in \mathbb{R}$ . We also appended 1 to the beginning of  $\mathbf{x}_n$  so both  $\mathbf{x}_n$  and  $\mathbf{w}$  were in  $\mathbb{R}^{D+1}$ . Thus, here  $p$  is the same thing as  $D+1$ . Compare  $w_1, w_2, \dots, w_p$  with  $w_0, w_1, \dots, w_D$ .

1-1pt Write down  $\mathcal{L}(\mathbf{w})$  in matrix/vector forms using only  $\mathbf{X}$ ,  $\mathbf{y}$  and  $\mathbf{w}$  and the L2 norm. In other words, you are not allowed to use any components  $y_i$ ,  $w_j$  or  $\mathbf{x}_i$ ! (For any vector  $\mathbf{z}$  use the following notation  $\|\mathbf{z}\|_2$  to mean the L2 norm of  $\mathbf{z}$  See <http://mathworld.wolfram.com/L2-Norm.html> for more information about the L2 norm.)

$$L(w) = (\sum_{i=1}^n (y_i - w^T X_i)^2) + (\alpha \sum_{j=1}^p w_j^2)$$

$$(y - w^T X)^2 + \alpha w^T$$

$$\boxed{(y - wX)(y - wX)^T + \alpha w w^T}$$

2-1pt Derive and write down in matrix/vector forms the gradient of  $\mathcal{L}(\mathbf{w})$  with respect to  $\mathbf{w}$ . Show all the derivations. (Hint: You can start by computing the gradient of the full expression and then convert it to matrix/vector forms. You can also directly get the gradients from your answer in 1-)

$$L(w) = (y - wX)(y - wX)^T + \alpha w w^T$$

$$(y - wX)(y^T - w^T X^T) + \alpha w w^T$$

$$L(w) = (yy^T - yw^T X^T - wXy^T + ww^T XX^T) + \alpha ww^T$$

$$\frac{\partial L}{\partial w} = (0 - X^T y - X^T y + 2wXX^T) + 2\alpha w$$

$$= 0 - 2X^T y + 2wXX^T + 2\alpha w$$

$$\boxed{2X^T Xw - 2X^T y + 2\alpha w}$$

3-2pt Derive and write down in matrix/vector forms the solution  $\mathbf{w}^*$  to the optimization problem  $\min_{\mathbf{w}} \mathcal{L}(\mathbf{w})$ . Show all your derivations. (Hint:  $\mathcal{L}(\mathbf{w})$  is convex in  $\mathbf{w}$ )

$$\frac{\partial L}{\partial w} = 2X^T Xw - 2X^T y + 2\alpha w$$

$$2X^T Xw - 2X^T y + 2\alpha w = 0$$

$$X^T Xw + \alpha w = X^T y$$

$$(X^T X + \alpha)w = X^T y$$

$$w^* = \frac{X^T y}{(X^T X + \alpha I)}$$

4-2pt Under which condition on the  $\alpha$  is the solution  $\mathbf{w}^*$  unique? Prove rigorously your statement. Make no assumptions on  $\mathbf{X}$ . (Hint: If your solution  $\mathbf{w}^*$  requires to invert a matrix, then one necessary condition for uniqueness is for the matrix to be invertible. And any positive definite matrix [https://en.wikipedia.org/wiki/Definiteness\\_of\\_a\\_matrix](https://en.wikipedia.org/wiki/Definiteness_of_a_matrix) is invertible. You might also want to look at the properties of transposition <https://en.wikipedia.org/wiki/Transpose>)

For the solution to be unique,  $X^T X + \alpha I$  needs to be invertible. A matrix is invertible if it is positive definite. In our case we need to have the sum of  $X^T X + \alpha I$  positive definite and this is possible if one is positive definite and the other is positive semi-definite. The identity matrix is positive definite, and scaling a positive definite matrix by a factor larger than zero gives a positive definite matrix. If  $\alpha > 0$  we will have  $\alpha I$  positive definite and if we have only real entries in  $X$  we have  $X^T X$  positive semi-definite. If these conditions are satisfied the matrix is invertible and we have  $w^*$  as a unique solution

5-2pt Implement in Python a well commented function `fit_linear_with_regularization` that takes as input  $\mathbf{X}$ ,  $\mathbf{y}$  and  $\alpha$  and return  $\mathbf{w}^*$  as computed in question 3. You are not allowed to use any loops (for-loop, while-loop ...) to do the implementation. Instead use and abuse as much as possible numpy vectorization techniques. A skeleton of the function is shown in the code cell below.

6-3pt Implement in Python a well commented function `predict` that takes as input a dataset  $\mathbf{X}_{test}$  in the same dimensions as  $\mathbf{X}$  and return the predictions. Write down the mean squared error ([https://en.wikipedia.org/wiki/Mean\\_squared\\_error](https://en.wikipedia.org/wiki/Mean_squared_error)) of your predictions. Then on the same plot with legends, show:

a) A scatter plot of the first feature of  $\mathbf{X}_{test}$  (x-axis) and the diabetes progression  $\mathbf{y}_{test}$ .

b) A plot of your prediction for  $\mathbf{X}_{test}$ .

The skeleton code in the cell below already implements most of data loading and you should only have to fill in the TODO part. Again here no loops are allowed (for-loop, while loop in the implementation of the plots and the `predict`)

```
In [3]: # Make it possible to show plots in the notebooks.
import matplotlib inline
import numpy as np
from sklearn import datasets
import matplotlib
import matplotlib.pyplot as plt

def fit_linear_with_regularization(X, y, alpha):
    X = np.c_[np.ones(len(X)), X] # Adding a column with 1's in the beginning, shape - (422,18) to shape -(422,11)
    w = np.matmul(np.linalg.pinv(np.matmul(np.transpose(X), X) + np.identity(np.shape(X)[1]*alpha),
                    np.matmul(np.transpose(X), y))

    return w

def predict(X_test, w):
    X_test = np.c_[np.ones(len(X_test)), X_test] # Adding a column with 1's in the beginning, from (20,18) to(20,11)
    y_pred = np.matmul(X_test, w)
    return y_pred

def plot_training(X_train, y_train): # visualizing the data
    X_train = np.c_[np.ones(len(X_train)), X_train]
    plt.scatter(X_train[:, 1], y_train, label='real data')
    plt.title('Visualization of the training data')
    plt.xlabel('First attribute')
    plt.ylabel('Diabetes data')
    plt.legend()
    plt.show()

def plot_prediction(X_test, y_test, y_pred):
    # Scatter plot of first feature for all X test points compared to y_test
    fig, ax = plt.subplots()
    plt.scatter(X_test[:,0], y_test)

    # Plot y_pred using the first feature of X_test as x-axis
    plt.scatter(X_test[:,0], y_pred)
    ax.set_title("Diabetes test data against prediction data")
    ax.set_xlabel("First feature of X_test")
    ax.set_ylabel("Diabetes data")
    ax.set_xlabel("Diabetes data")
    ax.legend(['Actual Test', 'Prediction'])

    # Compute the mean squared error
    mean_squared_error = 2/len(y_test)*np.sum((y_test - y_pred)**2)
    return mean_squared_error

# Load the diabetes dataset
X, y = datasets.load_diabetes(return_X_y=True) # shape - (442,18)

# Split the dataset into training and test set
num_test_elements = 20

X_train = X[:num_test_elements] # shape - (422,18)
X_test = X[num_test_elements:] # shape - (20,18)

y_train = y[:num_test_elements] # shape - (422,)
y_test = y[num_test_elements:] # shape - (20,)

# Set alpha
alpha = 0.01

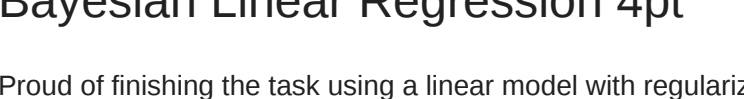
# Train using linear regression with regularization and find optimal model
w = fit_linear_with_regularization(X_train, y_train, alpha)

# Make predictions using the testing set X_test
y_pred = predict(X_test, w)

# plot_training(X_train, y_train)
# Plots and mean squared error
error = plot_prediction(X_test, y_test, y_pred)
print("Mean Squared error is ", error)

# Show the plot
plt.show();

Mean Squared error is 2003.7999650068894
```



### Bonus question

Answering this question, will not give you any additional points. Not answering this question will not prevent you from getting full points (if all other questions are answered correctly). However, if you answer this question, we will pick exactly one question where you didn't receive full points in this assignment and give you at most 4 more points there. In particular, the questions for which you have reasonably attempted a solution, we will pick the one where the difference between the full point and the point you received is the maximum.

### Bayesian Linear Regression 4pt

Proud of finishing the task using a linear model with regularization, you show your results to a representative of the Swedish Public Health Agency. You barely finish explaining your solution when the face of the representative turns red and you could distinctly hear: "Bayesian is the only way. How come didn't you use any probabilities?".

You quickly head back to your desk and now assume a Gaussian prior on the solution  $\mathbf{w}$ , that is  $p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \lambda^{-1} \mathbf{I})$  where  $\lambda \in \mathbb{R}$  is a constant real number,  $I$  is the  $p \times p$  identity matrix and  $\mathcal{N}(\mathbf{0}, \lambda^{-1} \mathbf{I})$  is used to mean the multivariate gaussian distribution with mean  $\mathbf{0} \in \mathbb{R}^p$ ,  $\lambda$  a vector of zeros of dimension  $p$  and covariance matrix  $\lambda^{-1} \mathbf{I}$ . Then, you use the following likelihood:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^n \mathcal{N}(\mathbf{w}^T \mathbf{x}_i, \gamma^{-1}) \text{ where } \gamma \in \mathbb{R} \text{ is a constant real number and } \mathcal{N}(\mathbf{w}^T \mathbf{x}_i, \gamma^{-1}) \text{ is the gaussian distribution with mean } \mathbf{w}^T \mathbf{x}_i \text{ and variance } \gamma^{-1}.$$

7-2pt Derive and express in vector/matrix form as a function of  $\mathbf{X}$ ,  $\mathbf{y}$ ,  $\mathbf{w}$  the log posterior  $\ln p(\mathbf{w}|\mathbf{y}, \mathbf{X})$ . Show all the derivations. You can ignore normalizing constants.