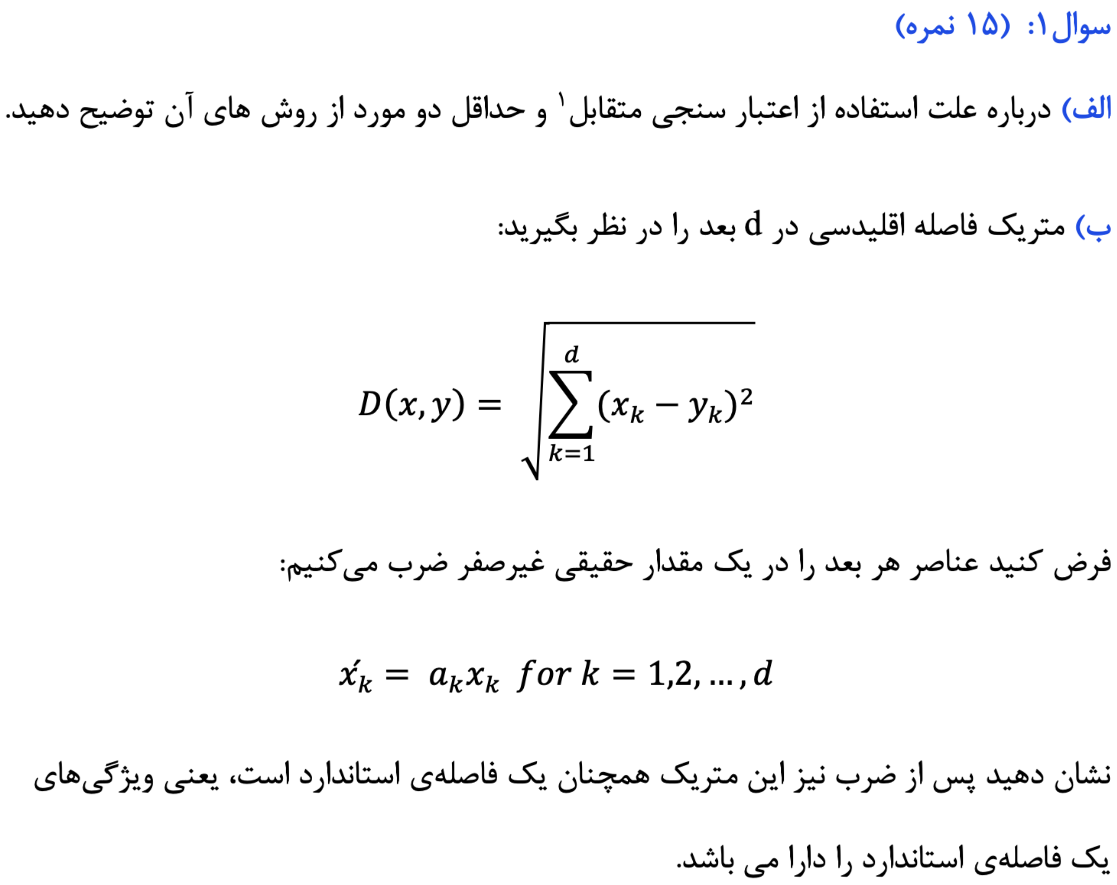
Project 1:



سوال ۱: (۱۵ نمره)

الف) درباره علت استفاده از اعتبار سنجی متقابل[۱] و حداقل دو مورد از روش های آن توضیح دهید.

ب) متریک فاصله اقلیدسی در d بعد را در نظر بگیرید:

$$D(x,y) = \sqrt{\sum_{k=1}^{d}(x_k - y_k)^2}$$

فرض کنید عناصر هر بعد را در یک مقدار حقیقی غیرصفر ضرب می‌کنیم:

$$x'_k = a_k x_k \ \ for\ k = 1,2,\dots,d$$

نشان دهید پس از ضرب نیز این متریک همچنان یک فاصله‌ی استاندارد است، یعنی ویژگی‌های
یک فاصله‌ی استاندارد را دارا می باشد.

Question 1 :

Part (a)

Question: Explain why reciprocal validity is used and describe two methods of it.

Answer:
In machine learning and statistics,
reciprocal validity (mutual validity) refers to the concept of ensuring that two

measurements or evaluations are comparable and validate each other. This concept is important when we are assessing the quality of our models or measurements.

Reciprocal validity can be especially useful in cases where:

1. You have two datasets or measurement methods, and you want to ensure that they provide consistent results.
2. You want to check if two models or approaches to the same problem yield similar outcomes, which strengthens the reliability of your analysis.

Two common methods for achieving reciprocal validity include:

1. **Cross-validation**: Cross-validation involves dividing the data into several subsets, or "folds." One fold is used for testing, while the others are used for training, and this process is repeated multiple times. This way, each data point gets a chance to be in a training set and a test set, providing a comprehensive evaluation of the model's validity.
2. **Holdout validation**: In holdout validation, the dataset is split into two parts: a training set and a testing (or validation) set. The model is trained on the training set and validated on the test set. Reciprocal validity can be established if the model performs consistently well across different random splits.

Reciprocal validity is crucial for ensuring that a model generalizes well to unseen data and for validating the reliability of the results.

Part (b)

Question: Consider the Euclidean distance in $d-dimensional$space, defined by:

$$D(x, y) = \sqrt{\sum_{k=1}^{d}(x_k - y_k)^2}$$

Suppose each element of each dimension is multiplied by a non-zero real scalar a_k Show that this scaling factor preserves the properties of a standardized distance metric, meaning it still represents a standardized distance.

Solution:

The Euclidean distance between two points $x = (x_1, x_2, \ldots, x_d)$ and $y = (y_1, y_2, \ldots, y_d)$ is given by:

$$D(x,y) = \sqrt{\sum_{k=1}^{d} (x_k - y_k)^2}$$

Now, let's consider that each element x_k and y_k is scaled by a non-zero factor a_k , resulting in transformed coordinates $x'_k = a_k x_k$ and $y'_k = a_k y_k$ for $k = 1, 2, \ldots, d.$

The distance $D(x', y')$ between the scaled points $x' = (a_1 x_1, a_2 x_2, \ldots, a_d x_d)$ and y' $= (a_1 y_1, a_2 y_2, \ldots, a_d y_d)$ becomes:

$$D(x', y') = \sqrt{\sum_{k=1}^{d} (x'_k - y'_k)^2}$$

Substitute $x'_k = a_k x_k$ and $y'_k = a_k y_k$:

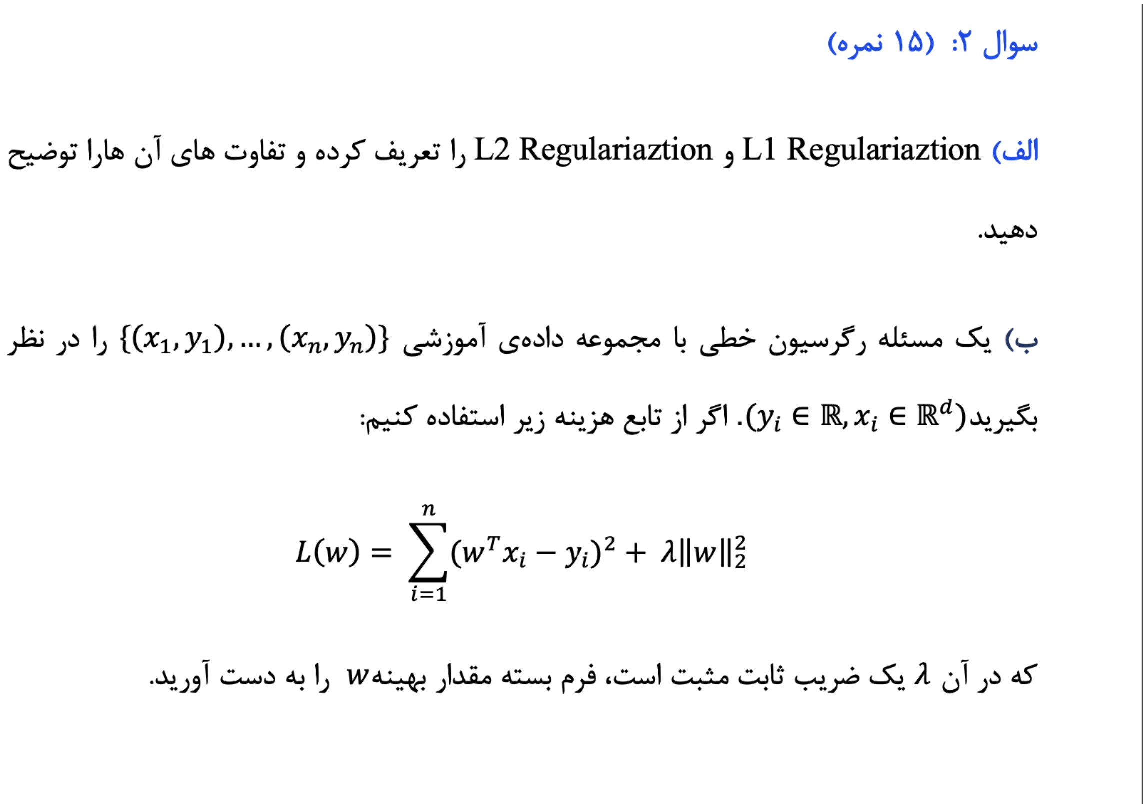
$$D(x', y') = \sqrt{\sum_{k=1}^{d} (a_k x_k - a_k y_k)^2}$$

Since a_k is a constant for each k , we can factor a_k out of the square:

$$D(x', y') = \sqrt{\sum_{k=1}^{d} a_k^2 (x_k - y_k)^2}$$

This modified formula introduces a weighting factor a_k^2 for each dimension, which scales the Euclidean distance but does not affect the fundamental structure of the metric. In other words, while the distance value itself is scaled, the relative distances and properties such as symmetry and the triangle inequality are preserved. Thus, this transformation still represents a **standardized distance** in the sense that it respects the form and properties of the Euclidean distance.

This scaling technique is common in machine learning, especially in cases where features have different units or variances, and it helps in emphasizing certain dimensions over others, making the metric more adaptable to the specifics of the data.



سوال ۲: (۱۵ نمره)

الف) L1 Regulariaztion و L2 Regulariaztion را تعریف کرده و تفاوت های آن هارا توضیح دهید.

ب) یک مسئله رگرسیون خطی با مجموعه داده‌ی آموزشی $\{(x_1,y_1),\dots,(x_n,y_n)\}$ را در نظر بگیرید$(y_i \in \mathbb{R}, x_i \in \mathbb{R}^d)$. اگر از تابع هزینه زیر استفاده کنیم:

$$L(w) = \sum_{i=1}^{n}(w^T x_i - y_i)^2 + \lambda\|w\|_2^2$$

که در آن λ یک ضریب ثابت مثبت است، فرم بسته مقدار بهینهw را به دست آورید.

Let's go through each part of this question in detail and provide a complete solution in English.

Part (a)

Question: Define $L1$ Regularization and $L2$ Regularization, and explain their differences.

Answer:
In the context of machine learning, regularization is a technique used to prevent overfitting by adding a penalty term to the model's loss function. This penalty term discourages the model from fitting too closely to the training data, which can improve its ability to generalize to new, unseen data.

1. L1 Regularization:

- L1 regularization, also known as **Lasso** (Least Absolute Shrinkage and Selection Operator), adds a penalty term equal to the absolute value of the magnitude of the coefficients. The cost function for L1 regularization is:

$$L(w) = \sum_{i=1}^{n}(f(x_i; w) - y_i)^2 + \lambda \sum_{j=1}^{d} |w_j|$$

where f(x_i; w) is the model's prediction, y_i is the true value, w represents the model parameters, and \lambda is a regularization hyperparameter.

- **Effect**: L1 regularization tends to produce sparse solutions, meaning it drives some of the weights w_j to zero. This effectively performs feature selection, as features with a weight of zero are effectively removed from the model.

2. L2 Regularization:

- L2 regularization, also known as **Ridge** regularization, adds a penalty term equal to the square of the magnitude of the coefficients. The cost function for L2 regularization is:

$$L(w) = \sum_{i=1}^{n}(f(x_i; w) - y_i)^2 + \lambda \sum_{j=1}^{d} w_j^2$$

- **Effect**: L2 regularization penalizes large weights but does not drive them exactly to zero. Instead, it shrinks the weights towards zero in a continuous manner, leading to a smoother, more stable model.

Differences between L1 and L2 Regularization:

- **Penalty type**: L1 regularization uses the absolute values of the coefficients, while L2 regularization uses the squared values.
- **Sparsity**: L1 regularization can result in sparse models with some coefficients exactly equal to zero, which is useful for feature selection. L2 regularization, on the other hand, typically results in non-zero coefficients for all features, reducing the model's complexity without completely removing any features.
- **Optimization**: L1 regularization leads to a non-differentiable function at zero, making it more challenging to optimize compared to L2, which is differentiable everywhere.
- **Use cases**: L1 is often used when feature selection is desired, whereas L2 is preferred when all features are relevant and we just want to reduce their impact.

Part (b)

Question: Consider a linear regression problem with a training dataset \{(x_1, y_1), \dots, (x_n, y_n)\} where y_i \in \mathbb{R} and x_i \in \mathbb{R}^d . If we use the following cost function:

$$L(w) = \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|_2^2$$

where \lambda is a positive constant, find the closed form of the optimal weight vector w .

Solution:

This cost function L(w) includes both a mean squared error term and an L2 regularization term. Expanding this cost function:

$$L(w) = \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \sum_{j=1}^{d} w_j^2$$

This can be written in matrix notation. Let:

- X be the n \times d matrix of input features where each row represents x_i .
- y be the n -dimensional vector of target values y_i .
- w be the d -dimensional vector of weights.

Then, the loss function can be rewritten as:

$$L(w) = \|Xw - y\|_2^2 + \lambda \|w\|_2^2$$

Expanding \|Xw - y\|_2^2 , we get:

$$L(w) = (Xw - y)^T (Xw - y) + \lambda w^T w$$

This simplifies to:

$$L(w) = w^T X^T X w - 2y^T X w + y^T y + \lambda w^T w$$

To find the optimal w , we take the derivative of $L(w)$ with respect to w and set it to zero:

$$\frac{dL(w)}{dw} = 2X^T X w - 2X^T y + 2\lambda w = 0$$

Dividing by 2 and rearranging terms, we get:

$$(X^T X + \lambda I)w = X^T y$$

where I is the identity matrix of dimension d \times d . This is a linear equation in w and can be solved as:

$$w = (X^T X + \lambda I)^{-1} X^T y$$

Summary of the Solution:

The closed-form solution for the weight vector w in the linear regression problem with $L2$ regularization is:

$$w = (X^T X + \lambda I)^{-1} X^T y$$

This formula is commonly used in **Ridge Regression**, where the regularization term $\lambda \|w\|_2^2$ helps prevent overfitting by penalizing large weights. The value of λ controls the strength of the regularization; larger values of λ lead to greater shrinkage of the weights.

سوال ۳: (۲۰ نمره)

در یک مســـــئله رگرســـــیون خطی، مجموعه داده‌ی $D = \{(x_1, y_1), \dots , (x_n, y_n)\}$را در اختیار داریم. رابطه‌ی احتمالاتی میان x و y را به صورت زیر در نظر می‌گیریم:

$$y_i = wx_i + \epsilon_i$$
$$\epsilon_i = \ \mathcal{N}(0{,}1)$$

که در آن w پارامتر مدل و ϵ_iیک نویز گوسی با میانگین صفر و واریانس ۱ است.

با فرض i.i.d بودن داده‌ها، تابع log-likelihood را تشکیل دهید و نشان دهید که بیشینه کردن تابع log-likelihood روی پارامتر معادل اســـت با کمینه کردن مجموع مجذور خطا، به عبارت دیگر نشان دهید:

$$\arg\max_{w} logP(D|w) = \arg\min_{w} \sum_{i=1}^{n} (y_i - wx_i)^2$$

Problem Statement

We are given a linear regression problem with a dataset $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ and a probabilistic relationship between X and Y as follows:

$$y_i = w^T x_i + \epsilon_i$$

where:

- w is the parameter vector of the model,
- ϵ_i is Gaussian noise with mean 0 and variance 1, i.e., $\epsilon_i \sim \mathcal{N}(0, 1)$.

The objective is to:

1. Formulate the log-likelihood function $\log P(D|w)$ given the data D and show that maximizing the log-likelihood is equivalent to minimizing the sum of squared errors:

   $\arg\max_w \log P(D|w) = \arg\min_w \sum_{i=1}^{n}(y_i - w^T x_i)^2$

Solution

Step 1: Define the Likelihood Function

Since y = w^T x_i + \epsilon_i and psilon_i \sim \mathcal{N}(0, 1) , each observed y_i is normally distributed around w^T x_i with variance 1. This gives us:

$$y_i \sim \mathcal{N}(w^T x_i, 1)$$

The probability density function of y_i given w is:

$$P(y_i|x_i, w) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2}\right)$$

Assuming the data points are independent and identically distributed (i.i.d.), the likelihood of the entire dataset D given w is the product of the probabilities for each y_i :

$$P(D|w) = \prod_{i=1}^{n} P(y_i|x_i, w) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2}\right)$$

Step 2: Derive the Log-Likelihood Function

To simplify the optimization, we take the logarithm of the likelihood function, which gives us the **log-likelihood**:

$$\log P(D|w) = \sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2}\right)\right)$$

Breaking this down:

$$\log P(D|w) = \sum_{i=1}^{n} \left(\log \frac{1}{\sqrt{2\pi}} - \frac{(y_i - w^T x_i)^2}{2}\right)$$

$$= \sum_{i=1}^{n} \left(-\frac{1}{2} \log(2\pi) - \frac{(y_i - w^T x_i)^2}{2} \right)$$

$$= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

Since -\frac{n}{2} \log (2\pi) is a constant with respect to w , we can ignore it for the purpose of optimization. Therefore, maximizing the log-likelihood function \log P(D | w) is equivalent to minimizing the following expression:

$$\frac{1}{2} \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

Step 3: Simplify to the Sum of Squared Errors

Since the factor $\frac{1}{2}$ does not affect the minimization, we can remove it, yielding the objective:

$$\arg\max_w \log P(D|w) = \arg\min_w \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

Conclusion

We have shown that maximizing the log-likelihood function \log P(D | w) is indeed equivalent to minimizing the sum of squared errors:

$$\arg\max_w \log P(D|w) = \arg\min_w \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

This result forms the basis for **Ordinary Least Squares (OLS)** regression, where minimizing the sum of squared errors is equivalent to maximizing the likelihood under the assumption of Gaussian noise with mean zero and variance one.

سوال ۴: (۲۰ نمره)

در یک مسئله رگرسیون، می خواهیم رابطه بین ورودی و مقدار خروجی را به صورت زیر مدل کنیم:

$$y = \exp wx$$

در رابطه بالا، $y \in \mathbb{R}$ و $x \in \mathbb{R}$ است و $w \in \mathbb{R}$ پارامتر مدل است. فرض کنید مجوعه داده آموزشی $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ را در اختیار داریم.

الف) تابع هزینه مجموع مجذور خطا را برای مجموعه داده ی D تشکیل دهید.

ب) اگر بخواهیم با استفاده از روش کاهش گرادیان مقدار بهینه w را به دست آوریم، رابطه بروزرسانی w چه خواهد بود؟ به عبارت دیگر w_{t+1} چگونه از w_t به دست می آید.

ج) با انجام محاسبات نشان دهید که برای کمینه کردن تابع هزینه، مقدار بهینه پارامتر w باید در کدامیک از روابط زیر صدق کند؟

الف) $\sum_{i=1}^{n} x_i \exp wx_i = \sum_{i=1}^{n} x_i y_i \exp wx_i$

ب) $\sum_{i=1}^{n} \exp wx_i = \sum_{i=1}^{n} x_i y_i \exp wx_i$

ج) $\sum_{i=1}^{n} x_i \exp 2wx_i = \sum_{i=1}^{n} x_i y_i \exp wx_i$

Problem Statement

We are dealing with a regression problem, where we want to model the relationship between an input and an output using the following model:

$$y = \exp(wx)$$

Here:

- y is the output,

- $w \in \mathbb{R}$ is the model parameter,
- $x \in \mathbb{R}$ is the input feature.

Suppose we have a dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.

Part (a)

Question: Formulate the cost function (sum of squared errors) for the dataset D .

Solution for (a)

The sum of squared errors (SSE) is a common cost function in regression tasks, defined as the sum of the squared differences between the observed values and the predicted values. For this model, the predicted value for an input x_i is \exp(w x_i) , so the cost function L(w) becomes:

$$L(w) = \sum_{i=1}^{n} \left(y_i - \exp(w x_i)\right)^2$$

This function L(w) represents the total squared error across all data points in D , and minimizing L(w) with respect to w will give us the best fit for the data according to the sum of squared errors criterion.

Part (b)

Question: If we want to find the optimal value of w using gradient descent, derive the update rule for w at each step, from w_t to w_{t+1} .

Solution for (b)

To use gradient descent, we need to calculate the derivative of L(w) with respect to w and then update w in the opposite direction of the gradient.

1. **Compute the Gradient**:

   $$\frac{dL(w)}{dw} = \frac{d}{dw} \sum_{i=1}^{n} \left(y_i - \exp(w x_i)\right)^2$$

   Using the chain rule, this becomes:

   $$\frac{dL(w)}{dw} = \sum_{i=1}^{n} 2 \left(y_i - \exp(w x_i)\right) \cdot \left(-x_i \exp(w x_i)\right)$$

Simplifying, we get:

$$\frac{dL(w)}{dw} = -2 \sum_{i=1}^{n} x_i \exp(wx_i) \left(y_i - \exp(wx_i)\right)$$

2. **Gradient Descent Update Rule**:
With a learning rate \eta , the update rule for w from w_t to w_{t+1} is:

$$w_{t+1} = w_t - \eta \left(-2 \sum_{i=1}^{n} x_i \exp(w_t x_i) \left(y_i - \exp(w_t x_i)\right)\right)$$\]

Simplifying further:

$$w_{t+1} = w_t + 2\eta \sum_{i=1}^{n} x_i \exp(w_t x_i) \left(y_i - \exp(w_t x_i)\right)$$

This is the update rule for w using gradient descent.

Part (c)

Question: Show through calculations that for minimizing the cost function, the optimal parameter w satisfies one of the following relations:

$$\text{(a) } \sum_{i=1}^{n} x_i y_i = \sum_{i=1}^{n} x_i y_i \exp(wx_i)$$

$$\text{(b) } \sum_{i=1}^{n} \exp(wx_i) = \sum_{i=1}^{n} x_i y_i \exp(wx_i)$$

$$\text{(c) } \sum_{i=1}^{n} x_i \exp(2wx_i) = \sum_{i=1}^{n} x_i y_i \exp(wx_i)$$

Solution for (c)

To find the correct relation, we need to set the derivative of L(w) with respect to w to zero. As we derived in part (b), the gradient of L(w) is:

$$\frac{dL(w)}{dw} = -2 \sum_{i=1}^{n} x_i \exp(wx_i) \left(y_i - \exp(wx_i)\right)$$

Setting this equal to zero for minimization:

$-2\sum_{i=1}^{n} x_i \exp(wx_i)\,(y_i - \exp(wx_i)) = 0$\]

Dividing by -2 , we get:

$$\sum_{i=1}^{n} x_i \exp(wx_i)\,(y_i - \exp(wx_i)) = 0$$

Expanding this expression:

$$\sum_{i=1}^{n} x_i y_i \exp(wx_i) - \sum_{i=1}^{n} x_i \exp(2wx_i) = 0$$

Rearranging terms, we get:

$$\sum_{i=1}^{n} x_i \exp(2wx_i) = \sum_{i=1}^{n} x_i y_i \exp(wx_i)$$

This matches **option (c)**:

$$\sum_{i=1}^{n} x_i \exp(2wx_i) = \sum_{i=1}^{n} x_i y_i \exp(wx_i)$$

Final Answer

The correct relation that satisfies the minimization condition for w is:

$$\sum_{i=1}^{n} x_i \exp(2wx_i) = \sum_{i=1}^{n} x_i y_i \exp(wx_i)$$