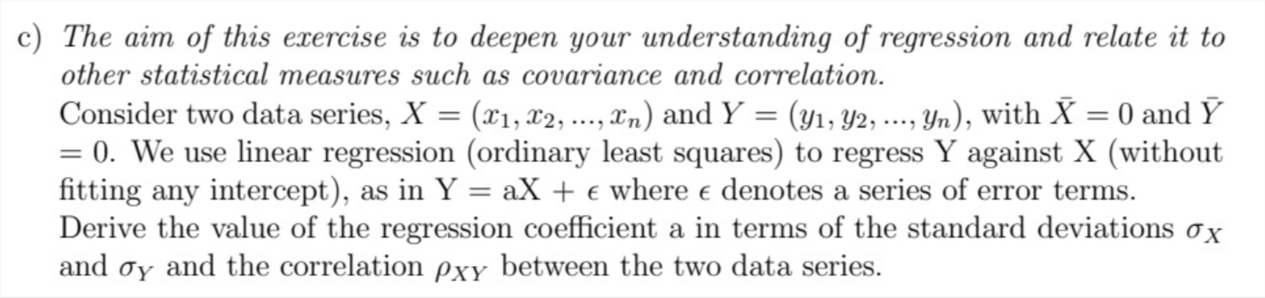


c) *The aim of this exercise is to deepen your understanding of regression and relate it to other statistical measures such as covariance and correlation.*
Consider two data series, $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$, with $\bar{X} = 0$ and $\bar{Y} = 0$. We use linear regression (ordinary least squares) to regress Y against X (without fitting any intercept), as in Y = aX + ϵ where ϵ denotes a series of error terms. Derive the value of the regression coefficient a in terms of the standard deviations σ_X and σ_Y and the correlation ρ_{XY} between the two data series.

From class we know that $a = (X^T X)^{-1} X^T y$. In our one dimensional case this yields

$$a = \left(\sum_{i=1}^{n} x_i^2\right)^{-1} \sum_{i=1}^{n} x_i y_i \qquad (1.1)$$

We recognize

$$\sum_{i=1}^{n} x_i^2 \overset{\bar{x}=0}{=} \sum_{i=1}^{n} (x_i - \bar{x})^2 = N \cdot Var(X) \qquad (1.2)$$

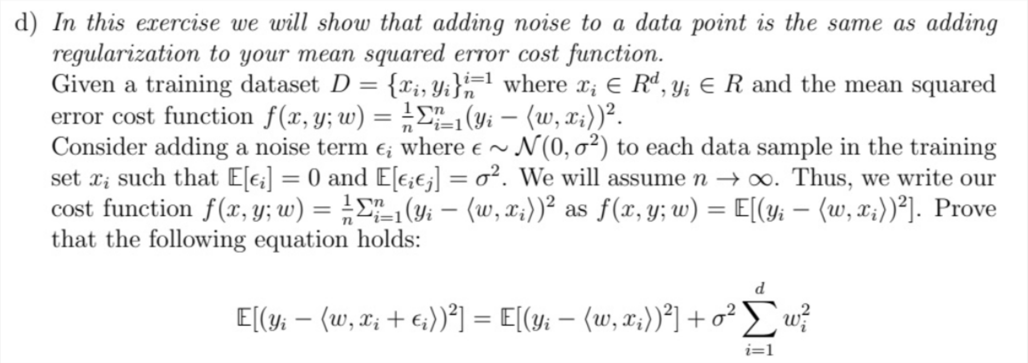
and

$$\sum_{i=1}^{n} x_i y_i \overset{\substack{\bar{x}=0 \\ \bar{y}=0}}{=} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) = N \cdot Cov(X,Y) \qquad (1.3)$$

Plugging in the results (1.2) and (1.3) into (1.1) yields

$$a = (N \cdot Var(X))^{-1} \cdot N \cdot Cov(X,Y) = \frac{Cov(X,Y)}{Var(X)}$$

∎



d) *In this exercise we will show that adding noise to a data point is the same as adding regularization to your mean squared error cost function.*

Given a training dataset $D = \{x_i, y_i\}_n^{i=1}$ where $x_i \in R^d, y_i \in R$ and the mean squared error cost function $f(x, y; w) = \frac{1}{n}\Sigma_{i=1}^n (y_i - \langle w, x_i \rangle)^2$.

Consider adding a noise term ϵ_i where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ to each data sample in the training set x_i such that $\mathbb{E}[\epsilon_i] = 0$ and $\mathbb{E}[\epsilon_i \epsilon_j] = \sigma^2$. We will assume $n \to \infty$. Thus, we write our cost function $f(x, y; w) = \frac{1}{n}\Sigma_{i=1}^n (y_i - \langle w, x_i \rangle)^2$ as $f(x, y; w) = \mathbb{E}[(y_i - \langle w, x_i \rangle)^2]$. Prove that the following equation holds:

$$\mathbb{E}[(y_i - \langle w, x_i + \epsilon_i \rangle)^2] = \mathbb{E}[(y_i - \langle w, x_i \rangle)^2] + \sigma^2 \sum_{i=1}^{d} w_i^2$$

$$E[(y_i - \langle w, x_i + \varepsilon_i \rangle)^2] = E[(y_i - \langle w, x_i \rangle - \langle w, \varepsilon_i \rangle)^2] \quad | \text{ Linearity of inner product}$$

$$= E[(y_i - \langle w, x_i \rangle)^2 - 2(y_i - \langle w, x_i \rangle)\langle w, \varepsilon_i \rangle + \langle w, \varepsilon_i \rangle^2]$$

$$= E[(y_i - \langle w, x_i \rangle)^2] + E[\langle w, \varepsilon_i \rangle^2] \quad | \ \langle w, \varepsilon_i \rangle = \varepsilon_i^T w$$

Now we have to make a small explanatory detour:

$$\langle w, \varepsilon_i \rangle^2 = \langle w, \varepsilon_i \rangle \cdot \langle w, \varepsilon_i \rangle$$

$$= \underbrace{(w^T \varepsilon_i)}_{\in \mathbb{R}} \cdot \underbrace{(w^T \cdot \varepsilon_i)}_{\in \mathbb{R}}$$

$$= w^T \cdot (\varepsilon_i \varepsilon_i^T) \cdot w$$

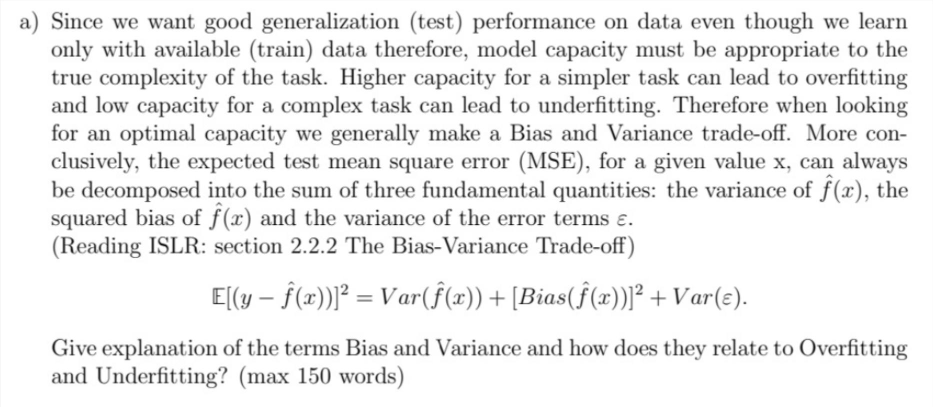
$$= w^T \begin{pmatrix} \varepsilon_{i1}\varepsilon_{i1} & \cdots & \varepsilon_{i1}\varepsilon_{id} \\ \vdots & \ddots & \vdots \\ \varepsilon_{id}\varepsilon_{i1} & \cdots & \varepsilon_{id}\varepsilon_{id} \end{pmatrix} w \quad | \text{ Since matrix multiplication is associative.}$$

$$= \sum_{j=1}^{d} \sum_{k=1}^{d} w_j \cdot \varepsilon_{ij}\varepsilon_{ik} \cdot w_k$$

From $E[\varepsilon_i \varepsilon_i^T] = \sigma^2 \cdot I$ we know that $E[\varepsilon_{ij}\varepsilon_{ik}] = \sigma^2$ if $k = j$ and 0 otherwise. Therefore we get

$$E[\langle w, \varepsilon_i \rangle^2] = \sum_{j=1}^{d} \sum_{k=1}^{d} w_j \cdot E[\varepsilon_{ij}\varepsilon_{ik}] \cdot w_k = \sigma^2 \cdot \sum_{j=1}^{d} w_i$$

∎



a) Since we want good generalization (test) performance on data even though we learn only with available (train) data therefore, model capacity must be appropriate to the true complexity of the task. Higher capacity for a simpler task can lead to overfitting and low capacity for a complex task can lead to underfitting. Therefore when looking for an optimal capacity we generally make a Bias and Variance trade-off. More conclusively, the expected test mean square error (MSE), for a given value x, can always be decomposed into the sum of three fundamental quantities: the variance of $\hat{f}(x)$, the squared bias of $\hat{f}(x)$ and the variance of the error terms ε.
(Reading ISLR: section 2.2.2 The Bias-Variance Trade-off)

$$\mathbb{E}[(y - \hat{f}(x))]^2 = Var(\hat{f}(x)) + [Bias(\hat{f}(x))]^2 + Var(\varepsilon).$$

Give explanation of the terms Bias and Variance and how does they relate to Overfitting and Underfitting? (max 150 words)

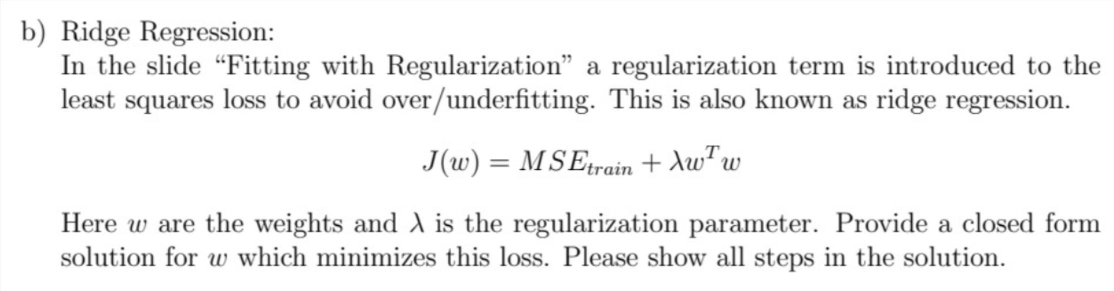
- **Variance** refers to the amount by which f would change if we estimated it using a different training data set. Since the training data are assumed to be sampled from a probability distribution, different training data sets will result in a different f.

- **Bias** refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model. In mathematical terms bias is given as

$$Bias(\hat{f}(x)) = E[\hat{f}] - f$$

  Where f is the function we are trying to approximate.

- Under the condition that the overall error remains constant, increasing the variance enables the model to fit more complex functions (i.e increased capacity). Thus, the model is able to fit training observations very well (i.e low MSE). Due to the fact that the model is incentivized by the loss function to minimize its MSE, this can lead to overfitting. Otherwise, increasing the bias reduces the capacity and potentially leads to underfitting.



b) Ridge Regression:
In the slide "Fitting with Regularization" a regularization term is introduced to the least squares loss to avoid over/underfitting. This is also known as ridge regression.

$$J(w) = MSE_{train} + \lambda w^T w$$

Here w are the weights and λ is the regularization parameter. Provide a closed form solution for w which minimizes this loss. Please show all steps in the solution.

We have $MSE_{train} = \frac{1}{m} \|Xw - y\|_2^2$, thus

$$J(w) = \frac{1}{m} \|Xw - y\|_2^2 + \lambda w^T w$$

We now take the gradient of $J(w)$

$$\nabla_w J(w) = \frac{1}{m} \cdot 2X^T(Xw - y) + 2\lambda w$$

We then set the gradient to zero

$$\nabla_w J(w) \hat{=} 0$$

$$\Leftrightarrow \quad X^T(Xw - y) = m\lambda w$$

$$\Leftrightarrow \quad X^T X w - X^T y = m\lambda w$$

$$\Leftrightarrow \quad (X^T X - m\lambda I) w = X^T y$$

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