CSE343/CSE543/ECE363/ECE563: Machine Learning Sec A (Monsoon 2024)
Mid-sem

Date of Examination: 11.10.2024 Duration: 1 hour Total Marks: 20 marks

---

Instructions –

- Attempt all questions. State any assumptions you have made clearly.
- MCQs may have multiple correct options. No evaluation without suitable justification.
- Standard institute plagiarism policy holds.

---

1. **[1 mark]** Given the Boolean function $A \wedge \neg B$, which of the following is the correct decision tree (Figure 1) representation for it?

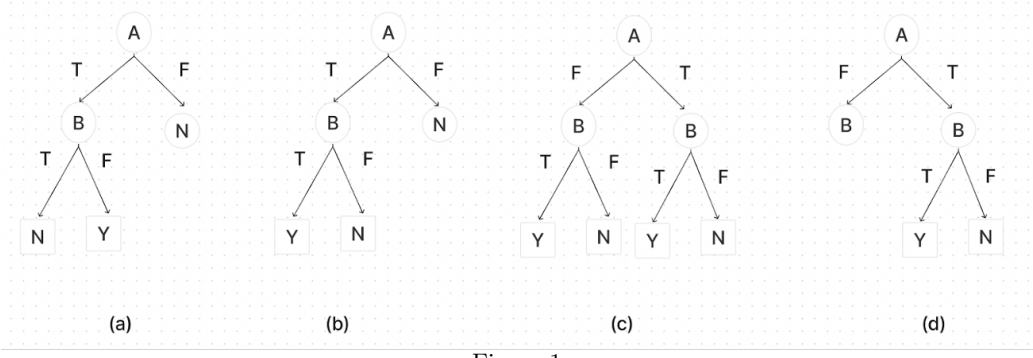

Figure 1

solution-(a) **1 mark for correct option and correct reason**

2. **[1 mark]** Assume the following likelihoods in Figure 2 for each word being part of a positive or negative movie review and equal prior probabilities for each class. What class will Naive Bayes assign to the sentence **"I always like foreign films."**?

	pos	neg
I	0.09	0.16
always	0.07	0.06
like	0.29	0.06
foreign	0.04	0.15
films	0.08	0.11

Figure 2

We classify the sentence **"I always like foreign films."** using Naive Bayes by calculating the likelihoods for both positive and negative classes. Given the word likelihoods and equal priors, we compute the following:
Positive class likelihood:

$$P(\text{sentence}|\text{positive}) = 0.09 \times 0.07 \times 0.29 \times 0.04 \times 0.08 \times 0.5 = 0.0000002912$$

Negative class likelihood:

$$P(\text{sentence}|\text{negative}) = 0.16 \times 0.06 \times 0.06 \times 0.15 \times 0.11 \times 0.5 = 0.00004752$$

Since $0.0000002912 < 0.00004752$, the Naive Bayes classifier assigns the sentence to the **negative** class.
1 mark for correct answer with correct positive and negative class likelihood

3. **[1 mark]** You are reviewing four papers submitted to a conference on machine learning for medical expert systems. All four papers validate their superiority on a standard benchmarking cancer dataset, which has only 5% of positive cancer cases. Which of the experimental settings is acceptable to you?

   - **Paper i**: We evaluated the performance of our model through a 5-fold cross-validation process and reported an accuracy of 93%.
   - **Paper ii**: The area under the ROC curve on a single left-out test set of our model is around 0.8, the highest among all the different approaches.
   - **Paper iii**: We computed the average area under the ROC curve through 5-fold cross-validation and found it to be around 0.75 – the highest among all the approaches.
   - **Paper iv**: The accuracy on a single left-out test set of our model is 95%, which is the highest among all the different approaches.

Which of the following options is acceptable?

(a) Paper i
(b) Paper i and Paper iv
(c) Paper ii and Paper iv
(d) Paper iii

Paper iii: This paper reports the average AUC over a 5-fold cross-validation, a more robust and reliable evaluation method. It accounts for variability across different subsets of data, making the results more generalizable.

1 mark for correct option and correct reason

4. **[1 mark]** In Logistic regression log(odds), transformation maps the ranges of probabilities to what range?

(a) $[-\infty, +\infty]$
(b) $[0, 1]$
(c) $[-1, +1]$
(d) $[-\infty, 0]$

Option (a)

1 mark for correct option and correct reason

5. **[1 mark]** Consider a Multi-Layer Perceptron (MLP) model with one hidden layer and one output layer. The hidden layer has 10 neurons, and the output layer has 3 neurons. The input to the MLP is a 5-dimensional vector. Each neuron is connected to every neuron in the previous layer, and a bias term is included for each neuron. The activation function used is the sigmoid function. Calculate the total number of trainable parameters in this MLP model.

Trainable Parameters Calculation in MLP Model **Input to Hidden Layer:**

- **Weights:** 5 input neurons $\times$ 10 hidden neurons = 50 weights
- **Biases:** 1 bias for each hidden neuron = 10 biases
- **Total parameters:** 50 + 10 = 60

Hidden to Output Layer:

- **Weights:** 10 hidden neurons $\times$ 3 output neurons = 30 weights
- **Biases:** 1 bias for each output neuron = 3 biases
- **Total parameters:** 30 + 3 = 33

Sum of Parameters Across Layers:

- **Total trainable parameters:** 60 (input-hidden) + 33 (hidden-output) = 93

0.25 mark for correct Input to Hidden Layer
0.25 mark for correct Hidden to Output Layer
0.5 mark for Sum of Parameters Across Layers

6. **[2 marks]** Show that the 'tanh' function and the 'logistic sigmoid' function are related by:

$$\tanh(a) = 2\sigma(2a) - 1$$

We want to show that the `tanh` function and the `logistic sigmoid` function are related by the equation:

$$\tanh(a) = 2\sigma(2a) - 1$$

1. **Logistic Sigmoid Function** $\sigma(x)$:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

2. **Hyperbolic Tangent (tanh) Function**:

$$\tanh(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}}$$

We begin by expressing $\tanh(a)$ in terms of the logistic sigmoid function $\sigma(x)$.

1. Start by expressing $\tanh(a)$:

$$\tanh(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}}$$

2. Now, manipulate the logistic sigmoid function for $2a$:

$$\sigma(2a) = \frac{1}{1 + e^{-2a}}$$

3. Multiply both sides of the equation for $\sigma(2a)$ by 2:

$$2\sigma(2a) = \frac{2}{1+e^{-2a}}$$

4. Subtract 1 from both sides:

$$2\sigma(2a) - 1 = \frac{2}{1+e^{-2a}} - 1$$

5. Simplify the right-hand side:

$$2\sigma(2a) - 1 = \frac{2-(1+e^{-2a})}{1+e^{-2a}} = \frac{1-e^{-2a}}{1+e^{-2a}}$$

6. Finally, compare this with the original definition of $\tanh(a)$:

$$\tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$$

We observe that:

$$2\sigma(2a) - 1 = \tanh(a)$$

Thus, we have shown that:

$$\tanh(a) = 2\sigma(2a) - 1$$

0.25 mark for correct Logistic Sigmoid Function
0.25 mark for correct Hyperbolic Tangent (tanh) Function
0.25 mark for each step

7. **[2 marks]** Given the equation representing the derivative of the sigmoidal activation function in terms of the output of any hidden neuron j, where a is an adjustable positive parameter:

$$\varphi_j'(v_j(n)) = a y_j(n)[1 - y_j(n)]$$

When does φ' attain its maximum and minimum values? Use this to infer when the synaptic weights change the most for a sigmoid activation function.

- **Maximum Value of $\varphi_j'(v_j(n))$:**
  The derivative $\varphi_j'(v_j(n)) = a y_j(n)[1 - y_j(n)]$ is a quadratic function in terms of $y_j(n)$, which attains its maximum value when $y_j(n) = \frac{1}{2}$. To find the maximum, we compute:

$$f(y_j(n)) = y_j(n)(1 - y_j(n)) \quad \text{and the maximum occurs when} \quad y_j(n) = \frac{1}{2}$$

  Substituting $y_j(n) = \frac{1}{2}$ gives the maximum value:

$$\varphi_j'(v_j(n)) = a \cdot \frac{1}{4}$$

  Hence, the derivative attains its maximum when $y_j(n) = \frac{1}{2}$.
- **Minimum Value of $\varphi_j'(v_j(n))$:**
  The derivative $\varphi_j'(v_j(n))$ attains its minimum value when $y_j(n) = 0$ or $y_j(n) = 1$, as:

$$\varphi_j'(v_j(n)) = 0 \quad \text{when} \quad y_j(n) = 0 \quad \text{or} \quad y_j(n) = 1$$

- **When Do Synaptic Weights Change the Most?**
  Since synaptic weight updates depend on the derivative of the activation function, the synaptic weights change the most when the derivative is largest, i.e., when $y_j(n) = \frac{1}{2}$. Therefore, the synaptic weights experience the most significant change when the output of the neuron is near $\frac{1}{2}$, which corresponds to the point of greatest learning sensitivity for the sigmoid activation function.

0.25 mark for correct Maximum Value
0.25 mark for correct Minimum Value
1 mark for correct 'When Do Synaptic Weights Change the Most'

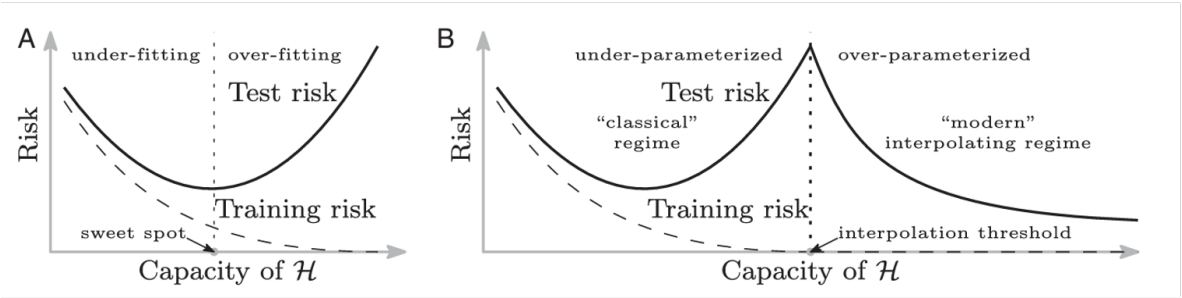

Figure 3

8. **[2 marks]** Figure 3, (A) represents the classical understanding of bias-variance tradeoff, where the test error increases as model capacity increases due to overfitting. Observe and explain the model behaviour as depicted in diagram (B). Analyse the reasoning behind this phenomenon and comment on the concept of the "interpolation threshold".

   Solution

   https://www.pnas.org/doi/10.1073/pnas.1903070116

   In diagram (A), the classical understanding of the bias-variance tradeoff is depicted, where the test risk (solid line) initially decreases as model complexity increases (reducing bias), but after a certain point, the test risk begins to increase due to overfitting (higher variance). This is a typical U-shaped curve that represents the classical regime. However, in diagram (B), we observe the **double-descent risk curve**, which incorporates both the classical U-shaped behavior as well as the behavior observed from high-capacity function classes, separated by the **interpolation threshold**.

   The interpolation threshold occurs when the model complexity increases to a point where it can perfectly fit or interpolate the training data, resulting in zero training risk (dashed line). This is where the curve transitions from the classical U-shaped regime to the modern interpolating regime. In this modern regime, even though the model perfectly fits the training data, the test risk starts to decrease again after an initial increase, leading to a "double descent" behavior in the overall risk curve.

   Analysis of the Risk Curve:

   - Left of the interpolation threshold: As the model complexity increases, the test risk decreases due to lower bias, but still remains higher than the training risk.
   - Right of the interpolation threshold: The model perfectly fits the training data (training risk is zero). As the complexity increases further, the test risk begins to decrease again, as the model starts to generalize well to unseen data, although this behavior is not typical in the classical regime. This is a departure from the classical U-shaped bias-variance curve, and it suggests that increasing model complexity beyond the interpolation threshold may lead to better performance on the test data, challenging the classical notion of overfitting.

   In conclusion, the double-descent curve in diagram (B) shows that after the interpolation threshold, model complexity can actually reduce the test risk, contrary to the classical understanding. Predictors to the right of the interpolation threshold have zero training risk, which implies that the model can fit the training data perfectly but still generalize well to unseen data under certain conditions.

   1 mark for correct 'model behaviour as depicted in diagram (B)'

   1 mark for correct Analysis. Give marks for correct approach.

9. **[2 marks]** Given a dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ with $x_i \in \mathbb{R}$ and $y_i \in \mathbb{R}$, derive the Maximum Likelihood Estimate (MLE) for linear regression model.

   Given a dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where $x_i \in \mathbb{R}$ and $y_i \in \mathbb{R}$, the linear regression model is:

$$y_i = w_0 + w_1 x_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

   where w_0 and w_1 are the parameters, and ϵ_i is the Gaussian noise with zero mean and variance σ^2.

   Step 1: Likelihood Function

   The likelihood function, assuming the errors ϵ_i are independent and normally distributed, is given by:

$$\mathcal{L}(w_0, w_1, \sigma^2 | D) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - (w_0 + w_1 x_i))^2}{2\sigma^2}\right)$$

   Step 2: Log-likelihood Function

   To simplify the calculations, we take the logarithm of the likelihood function:

$$\log \mathcal{L}(w_0, w_1, \sigma^2 | D) = -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - (w_0 + w_1 x_i))^2$$

The log-likelihood function needs to be maximized with respect to w_0 and w_1. Since σ^2 is not directly related to the weights w_0 and w_1, we can ignore the first term in the log-likelihood function and focus on minimizing the sum of squared residuals:

$$\log \mathcal{L}(w_0, w_1|D) = -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - (w_0 + w_1 x_i))^2$$

This is equivalent to minimizing the negative of the sum of squared errors:

$$S(w_0, w_1) = \sum_{i=1}^{N} (y_i - (w_0 + w_1 x_i))^2$$

Step 3: Maximum Likelihood Estimate (MLE) Using Argmax

To find the Maximum Likelihood Estimates for w_0 and w_1, we solve the following optimization problem:

$$\hat{w_0}, \hat{w_1} = \arg\max_{w_0, w_1} \log \mathcal{L}(w_0, w_1|D)$$

This is equivalent to solving:

$$\hat{w_0}, \hat{w_1} = \arg\min_{w_0, w_1} S(w_0, w_1)$$

Step 4: Solving for w_0 and w_1

We now compute the partial derivatives of $S(w_0, w_1)$ with respect to w_0 and w_1 to find the minimum.

Derivative with respect to w_0:

$$\frac{\partial S(w_0, w_1)}{\partial w_0} = -2 \sum_{i=1}^{N} (y_i - (w_0 + w_1 x_i)) = 0$$

This simplifies to:

$$N w_0 + w_1 \sum_{i=1}^{N} x_i = \sum_{i=1}^{N} y_i$$

Derivative with respect to w_1:

$$\frac{\partial S(w_0, w_1)}{\partial w_1} = -2 \sum_{i=1}^{N} x_i (y_i - (w_0 + w_1 x_i)) = 0$$

This simplifies to:

$$w_0 \sum_{i=1}^{N} x_i + w_1 \sum_{i=1}^{N} x_i^2 = \sum_{i=1}^{N} x_i y_i$$

Step 5: Solve the System of Equations

These two equations can be solved simultaneously to find $\hat{w_0}$ and $\hat{w_1}$. Solving gives the following Maximum Likelihood Estimates:

$$\hat{w_1} = \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N} (x_i - \bar{x})^2}$$

$$\hat{w_0} = \bar{y} - \hat{w_1} \bar{x}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of x_i and y_i.

Thus, the Maximum Likelihood Estimates for w_0 and w_1 are:

$$\hat{w_1} = \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N} (x_i - \bar{x})^2}$$

$$\hat{w_0} = \bar{y} - \hat{w_1} \bar{x}$$

2 marks for correct derivation. Give step marking

10. **[2 marks]** Train model on multinomial naive Bayes, with add-1 smoothing, on the following document counts for key sentiment words, with positive or negative class assigned as noted. Using the model assign a class (pos or neg) to the sentence: ***"A good, good plot and great characters, but poor acting"*** (see Figure 4).

doc	"good"	"poor"	"great"	(class)
d1.	3	0	3	pos
d2.	0	1	2	pos
d3.	1	3	0	neg
d4.	1	5	2	neg
d5.	0	2	0	neg

Figure 4

Note: Add-1 smoothing (Laplace smoothing) is a technique that adds 1 to all counts in a probability model to prevent zero probabilities for unseen events. The probability of a word w_i given a class c is given by:

$$\hat{p}(w_i \mid c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V}(\text{count}(w, c) + 1)}$$

The probability $p(w_i \mid c)$ is the fraction of times the word w_i appears among all words in all documents of topic/class c, divided by the total number of words in all documents of topic/class c. Where w_i is the i-th word in the vocabulary, and c is the class or topic.

We will calculate the probability of the given sentence belonging to both the positive and negative classes using **Multinomial Naive Bayes with Add-1 smoothing**.
Let the words in the sentence be denoted as $w_1 = \text{"A"}, w_2 = \text{"good"}, w_3 = \text{"good"}, w_4 = \text{"plot"}, w_5 = \text{"and"}, w_6 = \text{"great"}, w_7 = \text{"characters"}, w_8 = \text{"but"}, w_9 = \text{"poor"}, w_{10} = \text{"acting"}$.
First, we compute the prior probabilities for the classes `pos` and `neg` based on the class distribution in the dataset.

- Number of positive documents = 2 (d1, d2)
- Number of negative documents = 3 (d3, d4, d5)

The prior probabilities are calculated as follows:

$$P(\text{pos}) = \frac{2}{5}, \quad P(\text{neg}) = \frac{3}{5}$$

Vocabulary Size The vocabulary size is the number of unique words in the dataset. From the given table, the unique words are "good", "poor", and "great". Thus: V=3

Step 1: Calculate word probabilities for positive and negative classes

From the given table, we compute the probabilities for each word in the vocabulary with Add-1 smoothing. The probabilities for "good", "poor", and "great" in both the positive and negative classes are as follows:

$$P(\text{good} \mid \text{pos}) = \frac{3+1}{12} = \frac{4}{12} = \frac{1}{3}, \quad P(\text{poor} \mid \text{pos}) = \frac{1+1}{12} = \frac{2}{12} = \frac{1}{6}, \quad P(\text{great} \mid \text{pos}) = \frac{5+1}{12} = \frac{6}{12} = \frac{1}{2}$$

For each unseen word (6 unseen words in the sentence: "A," "plot," "and," "characters," "but," "acting."):

$$P(\text{unseen word} \mid \text{pos}) = \frac{0+1}{12} = \frac{1}{12}$$

$$P(\text{good} \mid \text{neg}) = \frac{2+1}{17} = \frac{3}{17}, \quad P(\text{poor} \mid \text{neg}) = \frac{10+1}{17} = \frac{11}{17}, \quad P(\text{great} \mid \text{neg}) = \frac{2+1}{17} = \frac{3}{17}$$

For each unseen word (6 unseen words in the sentence: "A," "plot," "and," "characters," "but," "acting."):

$$P(\text{unseen word} \mid \text{neg}) = \frac{0+1}{17} = \frac{1}{17}$$

Step 2: Calculate the Likelihood of the Sentence for Each Class

The sentence "A good, good plot and great characters, but poor acting" contains the words: "good" (2x), "poor", "great". We now calculate the likelihood of this sentence given each class.

Likelihood for Class pos:

$$P(\text{sentence} \mid \text{pos}) = P(\text{good} \mid \text{pos})^2 \times P(\text{poor} \mid \text{pos}) \times P(\text{great} \mid \text{pos}) = \left(\frac{1}{3}\right)^2 \times \frac{1}{6} \times \frac{1}{2} \times \left(\frac{1}{12}\right)^6 = \frac{1}{9} \times \frac{1}{6} \times \frac{1}{2} \times \left(\frac{1}{12}\right)^6 = \frac{1}{322466112}$$

Likelihood for Class neg:

$$P(\text{sentence} \mid \text{neg}) = P(\text{good} \mid \text{neg})^2 \times P(\text{poor} \mid \text{neg}) \times P(\text{great} \mid \text{neg}) = \left(\frac{3}{17}\right)^2 \times \frac{11}{17} \times \frac{3}{17} \times \left(\frac{1}{17}\right)^6$$

$$= \frac{9}{289} \times \frac{11}{17} \times \frac{3}{17} \times \left(\frac{1}{17}\right)^6 = \frac{297}{2015993900449}$$

Step 3: Prediction

The class with the higher posterior probability is assigned to the sentence. Based on the calculated likelihood values:

$$P(\text{sentence} \mid \text{pos}) = \frac{1}{322466112}, \quad P(\text{sentence} \mid \text{neg}) = \frac{297}{2015993900449}$$

Since $\frac{1}{322466112} \approx 3.1 \times 10^{-9}$ is greater than $\frac{297}{2015993900449} \approx 1.47 \times 10^{-10}$,
the sentence "A good, good plot and great characters, but poor acting" is more likely to belong to the **positive** class.
1 mark for correct 'Step 1: Calculate word probabilities for positive and negative classes'
0.5 mark for correct 'Compute the posterior probabilities for both classes.'
0.5 mark for correct prediction. Give marks for correct approach.

11. **[2 marks]** Given the Table 1, which presents information about various hypothesis functions h_i and their corresponding training error (TRAIN-ERR) and 10-fold cross-validation error (10-FOLD-CV-ERR), which model would you choose as the optimal hypothesis function? Why?

i	Hypothesis	TRAIN-ERR	10-FOLD-CV-ERR
1	h_1	0.90	0.91
2	h_2	0.78	0.45
3	h_3	0.44	0.29
4	h_4	0.38	0.55
5	h_5	0.25	0.72
6	h_6	0.12	0.88

Table 1

Key points to consider:
1. **Overfitting:** A very low training error but a high cross-validation error usually indicates overfitting. The model fits the training data too closely but fails to generalize well to unseen data.
2. **Bias-Variance Tradeoff:** A good model should have both low training and cross-validation errors, indicating a balance between bias and variance.
Analysis:
- Hypothesis h_6 has the **lowest training error** (0.12) but the **highest cross-validation error** (0.88), suggesting **overfitting**. The model performs very well on the training data but poorly on unseen data.

- Hypothesis h_5 has **low training error** (0.25), but the cross-validation error (0.72) is much higher, indicating some overfitting but not as severe as h_6.

- Hypotheses h_3 and h_4 strike a good balance between **training error** and **cross-validation error**:

- h_3: Training error = 0.44, 10-fold CV error = 0.29
- h_4: Training error = 0.38, 10-fold CV error = 0.55

h_3 shows a good balance between training and cross-validation error, suggesting that it generalizes well.
Conclusion:
The optimal hypothesis function is h_3 as it has: - A reasonable **training error** (0.44) that is not too low, avoiding overfitting. - A low **cross-validation error** (0.29), indicating good generalization to unseen data. Thus, h_3 strikes the best balance between bias and variance, making it the most suitable model for generalization.
1 mark for choosing correct hypothesis and 1 mark for the correct raeson

12. **[3 marks]** Prove that the variance of an ensemble is strictly smaller than the variance of an individual model, even if the variables are simply i.d. (identically distributed, but not necessarily independent).
Let us consider an ensemble of N models and denote the predictions from the i-th model as $f_i(x)$ where $i \in \{1, 2, ..., N\}$, and x is the input data. Assume that the models are identically distributed but not necessarily independent.
Definitions and Notations:
Ensemble Prediction: The prediction of the ensemble is taken as the average of the predictions from the individual models:

$$f_{\text{ensemble}}(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

Variance of the Ensemble Prediction: The variance of the ensemble prediction is given by:

$$\text{Var}(f_{\text{ensemble}}(x)) = \text{Var}\left(\frac{1}{N} \sum_{i=1}^{N} f_i(x)\right)$$

Since variance is a linear operator, we can use the following identity for the variance of a sum of random variables:

$$\text{Var}\left(\sum_{i=1}^{N} f_i(x)\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} \text{Cov}(f_i(x), f_j(x))$$

Thus, the variance of the ensemble becomes:

$$\text{Var}(f_{\text{ensemble}}(x)) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \text{Cov}(f_i(x), f_j(x))$$

Variance of an Individual Model: The variance of an individual model $f_i(x)$ is given by:

$$\text{Var}(f_i(x)) = \mathbb{E}[(f_i(x))^2] - (\mathbb{E}[f_i(x)])^2$$

For identically distributed models, the variance of each model is the same:

$$\text{Var}(f_i(x)) = \text{Var}(f_j(x)) \quad \forall i, j$$

Step 1: Variance of the Ensemble

We can write the variance of the ensemble as:

$$\text{Var}(f_{\text{ensemble}}(x)) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \text{Cov}(f_i(x), f_j(x))$$

Step 2: Decomposing the Covariance Terms

We can decompose the double sum into two parts: the diagonal terms where $i = j$ and the off-diagonal terms where $i \neq j$.

- **Diagonal terms** ($i = j$): These terms are the variances of the individual models:

$$\sum_{i=1}^{N} \text{Cov}(f_i(x), f_i(x)) = \sum_{i=1}^{N} \text{Var}(f_i(x)) = N \cdot \text{Var}(f_i(x))$$

- **Off-diagonal terms** ($i \neq j$): These are the covariances between different models. In general, the models may not be independent, but the covariance terms between different models will be less than the variance of the individual models:

$$\sum_{i \neq j} \text{Cov}(f_i(x), f_j(x))$$

Step 3: Simplify the Expression

The variance of the ensemble is now:

$$\text{Var}(f_{\text{ensemble}}(x)) = \frac{1}{N^2}\left(N \cdot \text{Var}(f_i(x)) + \sum_{i \neq j} \text{Cov}(f_i(x), f_j(x))\right)$$

Since the covariance between different models is non-negative and $\sum_{i \neq j} \text{Cov}(f_i(x), f_j(x)) \leq N \cdot \text{Var}(f_i(x))$, we can conclude:

$$\text{Var}(f_{\text{ensemble}}(x)) < \text{Var}(f_i(x))$$

Step 4: Intuition Behind the Result

The variance of the ensemble is strictly smaller than the variance of an individual model due to the following reasons:

- **Averaging**: The ensemble prediction is the average of the individual model predictions. Averaging reduces the impact of the noise (variance) of any single model.
- **Covariance Terms**: Even if the models are not independent, the sum of the covariance terms is still bounded by the variance of the individual models. This implies that averaging reduces the variability of the ensemble compared to the variability of any single model.

The variance of the ensemble is strictly smaller than the variance of an individual model, even if the models are identically distributed but not necessarily independent. The reduction in variance arises due to the averaging process, which mitigates the individual model's fluctuations.

Alternate Approach

3 mark. Give marks for correct approach. Give step marking

Q12] Variance of ensemble model

$$Var(\bar{x}) = Var\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)$$

$$= \frac{1}{n^2}\left(\sum_{i=1}^{n} X_i\right)$$

$$\Rightarrow \frac{1}{n^2}\left(\sum_{i=1}^{n} Var(X_i) + \underbrace{\sum_{i\neq j} Cov(X_i, X_j)}_{\text{non diag entries}}\right)$$

$$\Rightarrow \frac{1}{n^2}\left[n\sigma^2 + (n^2-n)\rho\sigma^2\right]$$

$$Var(\bar{x}) = \frac{\sigma^2 + (n-1)\rho\sigma^2}{n}$$

Variance of single model $= \sigma^2$

To prove $\Big|$ $Var(\text{Ensemble model}) < Var(\text{single model})$

Lets prove by contradiction ;

assume, $Var(\text{Ensemble model}) \geq Var(\text{single model})$

$$\frac{\sigma^2 + (n-1)\rho\sigma^2}{n} \geq \sigma^2$$

$$1 + (n-1)\rho \geq n$$

Good Write

$$(n-1)\rho \geq n-1$$

$$\rho \geq 1$$

$\rho > 1$ → not possible by definition of corr. coef.

$\rho = 1$ → not possible as given in the question variable are not necessarily independent or in other words. it means not perfectly correlated.

Hence, assumption is wrong.

So, by contradiction ;

$$Var(\text{Ensemble model}) < Var(\text{single model})$$