University of Stuttgart | ANALYTIC COMPUTING

Machine Learning (SS 24)

Main Exam, Masters (12 August 2024) (Solution)

Prof. Dr. Steffen Staab
Steffen.Staab@ki.uni-stuttgart.de

Akram Sadat Hosseini
Akram.Hosseini@ki.uni-stuttgart.de

Daniel Frank
Daniel.Frank@ki.uni-stuttgart.de

Tim Schneider
tim.schneider@ki.uni-stuttgart.de

Yi Wang
yi.wang@ki.uni-stuttgart.de

Farane Jalali-Farahani
farane.jalali-farahani@ki.uni-stuttgart.de

Osama Mohammed
osama.mohammed@ki.uni-stuttgart.de

Rodrigo Lopez Portillo Alcocer
rodrigo.lopez-portillo-alcocer@ki.uni-stuttgart.de

Student

First and last name

Immatriculation number

Email (optional)

Course of study (including B. Sc./M. Sc.)

Signature

Result

8	13	19	15	27	10	28	120
Maximum points							Sum
							Sum

Achieved points

Reviewer's signature

Grade

- Check that your exam copy is complete (38 pages, 7 tasks).
- Use a non-erasable writing medium, preferably ballpoint pen. Do *not* write in pencil.
- The use of aids of any kind is not permitted. This includes calculators, mobile phones, book, hand-written notes and the likes.
- If space permits, please provide your answers on the respective task page. If needed, you may request additional sheets (3 are already bundled with this exam at the end). Label these with your name, matriculation and task numbers and cross reference them from the respective task page.
- By taking this exam, you confirm that you understand and meet the eligibility requirements of this exam. If these are not fulfilled, the exam will be considered as not taken.
- Duration: 120 min

Good luck!

Task 1: Multiple Choice and short answer questions (8 Points)

Task (8 Points) For open questions, answer each question shortly. For multiple choice questions, tick the correct choice. Each multiple choice question has **one** correct choice. A question counts as 1 point when correct and 0 points when incorrect.

1. What is the primary purpose of activation functions in neural networks?

   Solution:
   To introduce non-linearity into the network.

2. What is the role of backpropagation in neural networks?

   Solution:
   To update the weights based on the gradient of the loss function.

3. What is the difference between Transductive SVM and regular/plain SVM?

   Solution:
   Transductive SVM extends regular/plain SVM by incorporating both labeled and unlabeled data into the training process.

4. In SVM, what can be said about the optimality of the solution for both the primal and dual problems?

Solution:
The solution is always globally optimal for both the primal and dual problems.

5. What is a key characteristic of Random Forest classifiers?
   - ☐ They are prone to overfitting with very large datasets.
   - ☐ They aggregate the predictions of multiple decision trees to improve accuracy.*
   - ☐ They perform best when the trees are fully grown without pruning.
   - ☐ They use a single feature at each split to ensure model simplicity.
6. Which of the following statements accurately describes a characteristic of the AdaBoost algorithm in the context of boosting?
   - ☐ AdaBoost assigns higher weights to correctly classified instances to ensure they influence subsequent learners more.
   - ☐ In AdaBoost, weak learners are combined using a simple average of their predictions.
   - ☐ The final model in AdaBoost is heavily influenced by the last weak learner added to the ensemble.
   - ☐ AdaBoost minimizes the exponential loss function. *
7. Assume you are given a machine learning model that is trained on a dataset. You notice that the training error is very low, but the test error is significantly higher. Which of the following statements best describes this scenario?
   - ☐ The model likely has high bias and low variance.
   - ☐ The model likely has low bias and high variance. *
   - ☐ The model likely has low bias and low variance.
   - ☐ The model likely has high bias and high variance.
8. Cross entropy is ...
   - ☐ ... another term for relative entropy.
   - ☐ ... the sum of entropy and Kullback-Leibler Divergence. *
   - ☐ ... measuring the uncertainty of a random variable.
   - ☐ ... defined as $H(x) := -\sum_x P(X = x) \log_b P(X = x)$.

Task 2: Naive Bayes (13 Points)

1. **Task (4 Points)** Shortly explain what the Naive Bayes assumption is and why it is used.

Solution:

- The Naive Bayes assumption states that all input variables (features) are conditionally independent of each other given the class label. This means that the presence or absence of a particular feature is assumed to be unrelated to the presence or absence of any other feature, provided we know the class. (2 point)
- This assumption is used primarily because it allows for the estimation of the joint probability distribution of the features with significantly less data. Without this assumption, learning the joint distribution would require an impractically large amount of data. By assuming independence, the calculation of the joint probability distribution becomes feasible, as it is reduced to the product of the individual conditional probabilities. This simplification also contributes to computational efficiency, making the Naive Bayes classifier practical and effective for many applications. (2 points)

2. **Task (2 Points)**
   - Why is Laplace smoothing used in Naive Bayes? (1 point)
   - How is Laplace smoothing applied in Naive Bayes? (1 point)

Solution:

1. Laplace smoothing is a technique used in Naive Bayes classifiers to address the problem of zero probability when a particular feature-value pair is not observed in the training data for a given class.
2. It works by adding a small, positive constant (typically 1) to both the counts of how often a feature value co-occurs with a class and the total count of occurrences across all possible feature values.

3. **Task (7 Points)** You are provided with a dataset containing information about whether a student passes or fails an exam based on two features: **Study Hours** and **Attendance**. The dataset is as follows:

Study Hours (High/Low)	Attendance (High/Low)	Pass (Yes/No)
High	High	Yes
High	Low	Yes
Low	High	No
Low	Low	No
High	High	Yes
Low	High	No
High	Low	No
Low	Low	No

Table 1 Dataset of Study Hours, Attendance, and Exam Pass Results

Using Naive Bayes classification, determine whether a student who studies **High** hours and has **Low** attendance will pass or fail.

Solution:
The Naive Bayes classifier uses Bayes' theorem with the assumption of independence between features. We need to calculate the probability of passing given High study hours and Low attendance.

1. Prior Probabilities (1 point)

$$P(\text{Pass} = \text{Yes}) = \frac{3}{8}, \quad P(\text{Pass} = \text{No}) = \frac{5}{8}$$

2. Likelihoods (2 point)

$$P(\text{Study Hours} = \text{High}|\text{Pass} = \text{Yes}) = \frac{2}{3}, \quad P(\text{Study Hours} = \text{High}|\text{Pass} = \text{No}) = \frac{1}{5}$$

$$P(\text{Attendance} = \text{Low}|\text{Pass} = \text{Yes}) = \frac{1}{3}, \quad P(\text{Attendance} = \text{Low}|\text{Pass} = \text{No}) = \frac{3}{5}$$

3. Posterior Probabilities (3 point)
For Pass = Yes:

$$P(\text{Pass} = \text{Yes}|\text{Study Hours} = \text{High}, \text{Attendance} = \text{Low}) \propto \left(\frac{2}{3}\right) \times \left(\frac{1}{3}\right) \times \left(\frac{3}{8}\right) = \frac{1}{12}$$

For Pass = No:

$$P(\text{Pass} = \text{No}|\text{Study Hours} = \text{High}, \text{Attendance} = \text{Low}) \propto \left(\frac{1}{5}\right) \times \left(\frac{3}{5}\right) \times \left(\frac{5}{8}\right) = \frac{3}{40}$$

4. Comparison (1 point)

$$\frac{1}{12} \approx 0.0833, \quad \frac{3}{40} = 0.075$$

Since $0.0833 > 0.075$, the Naive Bayes classifier predicts that a student with **High** study hours and **Low** attendance will **Pass** the exam.

Task 3: Decision Trees (19 Points)

You are asked to classify cars based on three features: "Engine Type", "Number of Doors", and "Fuel Efficiency". The target categories are "Sports", "Luxury", "Economy", and "SUV". The following table provides eight cars with different features and their corresponding target class.

Table 2 Car classification dataset

Car	Engine Type	Number of Seats	Fuel Efficiency	Target class
C1	Gasoline	Two	Low	Sports
C2	Diesel	Five	High	Economy
C3	Gasoline	Two	Low	Sports
C4	Diesel	Seven	Medium	Luxury
C5	Gasoline	Five	Medium	Sports
C6	Diesel	Seven	Medium	Luxury
C7	Diesel	Five	High	SUV
C8	Gasoline	Five	Medium	Sports

Based on the dataset in Table 2, you are asked to determine the optimal root-node split using information gain. Please answer the following questions.

1. **Task (3 Points)** Calculate the entropy of the initial dataset.

2. **Task (7 Points)** Calculate the information gain for the "Engine Type" feature.

3. **Task (1 Point)** Given that the IG(Number of Seats) = 1.22 and IG(Fuel Efficiency) = 1.69, based on your calculations in sub-task (b), determine which feature you would choose for the root node split. Justify your answer.

4. **Task (2 Points)** In classification, we use the maximum information gain to perform the split. For a regression with decision trees, what criterion would you use to perform the split, and why?

Solution:

(a) The entropy of the dataset is calculated using the formula:

$$\text{Entropy}(S) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

The probabilities for each category are (1 point):

- Sports: $4/8 = 1/2$
- Economy: $1/8$
- Luxury: $2/8 = 1/4$
- SUV: $1/8$

Therefore, the entropy is (2 points):

$$\text{Entropy}(S) = -\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{8}\log_2\left(\frac{1}{8}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{1}{8}\log_2\left(\frac{1}{8}\right)\right)$$

$$\text{Entropy}(S) = -\left(\frac{1}{2}\times(-1) + \frac{1}{8}\times(-3) + \frac{1}{4}\times(-2) + \frac{1}{8}\times(-3)\right) = 1.75$$

(b) Calculating Information Gain for "Engine Type":

$$\text{IG}(\text{Engine Type}) = \text{Entropy}(S) - \left(\frac{4}{8}\times\text{Entropy}(\text{Gasoline}) + \frac{4}{8}\times\text{Entropy}(\text{Diesel})\right)$$

For "Gasoline" cars (C1, C3, C5, C8) (3 points):

- Sports: $4/4$
- Luxury: 0
- Economy: 0
- SUV: 0

$$\text{Entropy}(\text{Gasoline}) = -\left(\frac{4}{4}\log_2\left(\frac{4}{4}\right) + 0 + 0 + 0\right) = 0$$

For "Diesel" cars (C2, C4, C6, C7) (3 points):

- Sports: 0
- Luxury: $2/4 = 1/2$
- Economy: $1/4$
- SUV: $1/4$

$$\text{Entropy}(\text{Diesel}) = -\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) = 1.5$$

(1 point)

$$\text{IG}(\text{Engine Type}) = 1.75 - \left(\frac{4}{8}\times 0 + \frac{4}{8}\times 1.5\right) = 1.75 - 0.75 = 1.0$$

(c) Comparing the information gains:

- "Engine Type": 1.0
- "Number of Seats": 1.22
- "Fuel Efficiency": 1.22

Therefore, **"Number of Seats"** and **"Fuel Efficiency"** are tied for the best feature for the root node split with the highest information gain of 1.22. We can choose either feature based on other criteria or further tiebreakers.

(d) In a regression problem, we predict continuous values instead of discrete classes (1 point). Therefore, we use criteria that minimize prediction error rather than maximizing information gain. Common criteria include (name one for 1 point):

- Mean Squared Error (MSE): Measures the average squared difference between actual and predicted values. Minimizing MSE helps in finding the split that results in the least prediction error.
- Mean Absolute Error (MAE): Measures the average absolute difference between actual and predicted values. Minimizing MAE can be more robust to outliers than MSE.

These criteria ensure that the splits reduce the overall error in predicting continuous outcomes.

5. Suppose we do a root node split and classify the dataset. The resulting classifications are as follows:

Car	Predicted Category	Actual Category
C1	Sports	Sports
C2	Economy	Economy
C3	Sports	Sports
C4	Luxury	Luxury
C5	Luxury	Sports
C6	Luxury	Luxury
C7	SUV	SUV
C8	Luxury	Sports

(a) **Task (3 Points)** Using this classification, create the confusion matrix.

(b) **Task (3 Points)** Calculate the following metrics for the "Luxury" class: accuracy, precision, recall, and F1-score.

- Accuracy =
- Precision =
- Recall =
- F1-Score =

Solution:

1. (d) Confusion matrix and metrics for "Luxury":

Predicted \Actual	Sports	Economy	Luxury	SUV
Sports	2 (0.5 points)	0	0 (0.5 points)	0
Economy	0	1 (0.5 points)	0	0
Luxury	2 (0.5 points)	0	2 (0.5 points)	0
SUV	0	0	0	1 (0.5 points)

   - **Accuracy** (0.5 points):

$$\text{Accuracy} = \frac{2+1+2+1}{8} = \frac{6}{8} = 0.75$$

   - **Precision for "Luxury"** (1 point):

$$\text{Precision} = \frac{2}{2+2} = \frac{2}{4} = 0.5$$

   - **Recall for "Luxury"** (1 point):

$$\text{Recall} = \frac{2}{2+0} = \frac{2}{2} = 1.0$$

   - **F1-Score for "Luxury"** (0.5 points):

$$\text{F1-Score} = 2 \times \frac{0.5 \times 1.0}{0.5 + 1.0} = 2 \times \frac{0.5}{1.5} = \frac{1.0}{1.5} = 0.667$$

Task 4: Bias and variance (15 Points)

Task 4.1: Expected squared error (2 Points)

1. **Task (2 Points)** What is the relation between bias, variance, and the expected squared model error $\mathbb{E}[(y - \hat{f})^2]$?
   Hint: *Provide the formula.*

Solution:

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

Task 4.2: Bias-variance trade-off for regression models (13 Points)

You are given a dataset $\mathcal{D} = \{(x_i, y_i)_{i=1}^N\}$ that is sampled from the model $y = 0.5x^2 + 2x + 1 + \mathcal{N}(0, 5)$. On the dataset, you trained the following regression models:

- **Linear Regression with linear features**: $\hat{f}_{\text{LinReg}}(x; \theta_{\text{LinReg}})$
- **Decision Tree Regressor**: $\hat{f}_{\text{Tree}}(x; \theta_{\text{Tree}})$
- **Random Forest Regressor**: $\hat{f}_{\text{RandomForest}}(x; \theta_{\text{RandomForest}})$

The models are mappings from the input $x \in X$ to the predicted label $\hat{y} \in Y$, $\hat{f}_k : X \to Y$ and are parametrized by some θ_k.

1. **Task (6 Points)** Analyze the bias and variance of each regression model ($\hat{f}_{\text{LinReg}}, \hat{f}_{\text{Tree}}, \hat{f}_{\text{RandomForest}}$) and match them to the corresponding dashed boxes in Figure 1. Explain your choices.

Solution:
3 points for correctly matching the models to the dashed boxed and 3 points for the explanations
- $\hat{f}_{\text{LinReg}}(x; \theta_{\text{LinReg}})$: linear regression model without features is not expressive enough to capture a quadratic function, therefore low variance and high bias.
- $\hat{f}_{\text{Tree}}(x; \theta_{\text{Tree}})$: A regular regression tree overfits the noisy data; therefore, it may generalizes badly to new inputs. The model has high variance and low bias.
- $\hat{f}_{\text{RandomForest}}(x; \theta_{\text{RandomForest}})$: Ensemble methods like Random Forest reduce the variance compared to the regression tree while retaining a rather low bias.

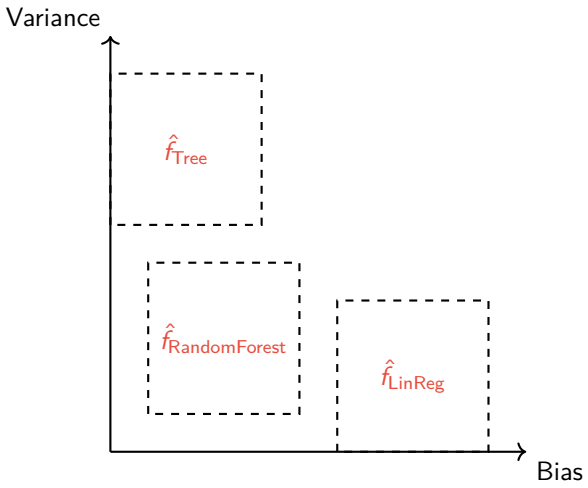


Figure 1 bias and variance for different regression models.

2. **Task (7 Points)** Assume you are given the generating function f, explain how you can calculate the **bias** and **variance** of a regression model $\hat{f}$ by providing a step-by-step description.

Solution:

- **(1 point)** Generate M datasets by sampling from f with the same size as $\mathcal{D}$. We denote them with $\mathfrak{D} = \{(x_i, y_i)_{i=1}^N\}$, $\mathfrak{M} = \{\mathfrak{D}\}_{j=1}^M$.
- **(1 point)** Train models $\hat{f}_j$ on the sampled datasets $\mathfrak{D}_j$ and generate predictions $\hat{y}_j$ on the same datasets.
- **(1 point)** Calculate the mean of the predictions $\mu = \frac{1}{M}\sum_{j=1}^M \hat{y}_j$. The mean consists of one value for each sample in each of the datasets, e.g., $\mu_i = \frac{1}{M}\sum_{j=1}^M \hat{f}_j(x_i)$
  - **(2 point)** The bias of $\hat{f}$ then follows as the mean squared error between the output of the original dataset $\mathcal{D}$ and the mean of the predictions

$$\text{Bias}^2(\hat{f}) = \frac{1}{N}\sum_{i=1}^N (\mu_i - y_i)^2.$$

  - **(2 point)** The variance describes how much the model $\hat{f}$ deviates from the mean. We can express this by calculating the mean squared error between the mean of the predictions and the predictions.

$$\text{Var}(\hat{f}) = \frac{1}{M}\frac{1}{N}\left(\sum_{j=1}^M \sum_{i=1}^N (\mu_i - (\hat{y}_i)_j)^2\right),$$

    where $(\hat{y}_i)_j = \hat{f}_j(x_i)$. (2 points)

Task 5: Logistic and Ridge Regression (27 Points)

Task 5.1: Ridge Regression (19 Points)

Let $f(\bar{\mathbf{x}}) = \phi(\bar{\mathbf{x}})^T\beta$ be a *ridge regression model* and $\phi : \mathbb{R}^d \to \mathbb{R}^D$ be a feature map. Assume you have a matrix of training data X and a vector of labels y compromised of n data points, s.t. $\hat{y} = \Phi(X)\beta$.

1. **Task (5 Points)** Annotate the right dimensions to the following vector- or matrix-valued quantities:

$X \in \mathbb{R}^{\square \times \square}$ $\quad$ $\bar{\mathbf{x}}^T \in \mathbb{R}^{1 \times \square}$

$X^TX \in \mathbb{R}^{\square \times \square}$ $\quad$ $\Phi(X)\Phi(X)^T \in \mathbb{R}^{\square \times \square}$

$\Phi(X)^T\Phi(X) \in \mathbb{R}^{\square \times \square}$ $\quad$ $y \in \mathbb{R}^{\square}$

Solution:

$X \in \mathbb{R}^{n \times d}$ $\quad$ $\bar{\mathbf{x}}^T \in \mathbb{R}^{1 \times d}$

$X^TX \in \mathbb{R}^{d \times d}$ $\quad$ $\phi(X)\phi(X)^T \in \mathbb{R}^{n \times n}$

$\phi(X)^T\phi(X) \in \mathbb{R}^{D \times D}$ $\quad$ $y \in \mathbb{R}^n$

0.5 points for each box. If students consistently switch the order all over the boxes but with correct numbers, they can have 4 points. If they get the correct values but switch the order just for some boxes, we give 0.25 points for those boxes.

2. **Task (3 Points)** What is a feature map $\phi(\cdot)$ and what is a kernel function $k(\cdot, \cdot)$ and how can they be related?

Solution:

- Feature maps allow to express non-linear relationships with the linear regression model (which is only linear in the parameters) by mapping each datapoint to a higher-dimensional data point. **(1 Point)**
- A kernel function provides a closed-form expression for the similarity of data points as measured by inner products of two datapoints. **(1 Point)**
- computing the inner products with a kernel is equivalent to the explict mapping to the corresponding feature space, but can be more efficient to compute **(1 Point)**

3. **Task (2 Points)** In the lecture we derived the closed-form solution to its optimal weights, as

$$\hat{\beta} = \left(\Phi(X)^T \Phi(X) + \lambda \cdot I\right)^{-1} \Phi(X)^T y$$

What happens if the parameter λ is increased? Explain your answer.

Solution:
Weights are penalized. This has an regularization effect and helps to prevent overfitting to the training data.

4. **Task (5 Points)** While introduced in the context of SVMs, kernels can also be applied to ridge regression. By plugging the optimal weights into the model definition, one gets

$$f(\bar{\mathbf{x}}) = \phi(\bar{\mathbf{x}})^T \hat{\beta} = \phi(\bar{\mathbf{x}})^T \left(\Phi(X)^T \Phi(X) + \lambda \cdot I\right)^{-1} \Phi(X)^T y$$

Show that $f(\bar{\mathbf{x}})$ can be expressed in terms of a kernel function $k_\phi(\cdot,\cdot)$ associated with the feature map ϕ. *Hint: Exploit the matrix identity*

$$\left(A^T A + cI\right)^{-1} A^T = A^T \left(AA^T + cI\right)^{-1}$$

Solution:

$$\begin{aligned} f(\bar{\mathbf{x}}) &= \phi(\bar{\mathbf{x}})^T \hat{\beta} \\ &= \phi(\bar{\mathbf{x}})^T \left(\phi(X)^T \phi(X) + \lambda \cdot I\right)^{-1} \phi(X)^T y \\ &= \phi(\bar{\mathbf{x}})^T \phi(X)^T \left(\phi(X)\phi(X)^T + \lambda \cdot I\right)^{-1} y \\ &= [k_\phi(\bar{\mathbf{x}}, x_j)]_{j=1,\dots,n}^T \left([k_\phi(x_i, x_j)]_{i,j=1,\dots,n} + I\right)^{-1} y \end{aligned}$$

Remark: $[k_\phi(\bar{\mathbf{x}}, x_j)]_{j=1,\dots,n} \in \mathbb{R}^n$ is a vector and $[k_\phi(x_i, x_j)]_{i,j=1,\dots,n} \in \mathbb{R}^{n \times n}$ is a matrix

5. **Task (4 Points)** The most expensive part in the training is the matrix inversion, which can be done in $\mathcal{O}(m^{2.807})$ with the Strassen algorithm for a matrix $A \in \mathbb{R}^{m \times m}$. Assign **one** of the complexities in

$$\{\mathcal{O}(d^{2.807}),\ \mathcal{O}(D^{2.807}),\ \mathcal{O}(n^{2.807}),\ \mathcal{O}((D+n)^{2.807})\}$$

to each algorithm. Multiple usages are possible.

Algorithm	Cost of the matrix inversion
linear regression (without feature map/ kernels)	
ridge regression (without feature map/ kernels)	
ridge regression (with feature map ϕ)	
ridge regression (with kernel function)	

Solution:

This is a trick questions that actually only asks for the size of the matrices involved: One needs to know that the kernel matrix is of the size of the number of datapoints.

Algorithm	Cost of the matrix inversion	
linear regression (without feature map/ kernels)	$\mathcal{O}(d^{2.807})$	
ridge regression (without feature map/ kernels)	$\mathcal{O}(d^{2.807})$	1 point for each
ridge regression (with feature map ϕ)	$\mathcal{O}(D^{2.807})$	
ridge regression (with kernel function)	$\mathcal{O}(n^{2.807})$	

correct complexity.

Task 5.2: Logistic Regression (8 Points)

You are given a classification problem with three classes.

1. **Task (3 Points)** Define the prediction $\sigma(\mathbf{x})$ of a logistic regression algorithm with weights $w_1, w_2, w_3 \in \mathbb{R}^2$ for a input $\mathbf{x} = (x_1, x_2)^T$ without features or bias terms.

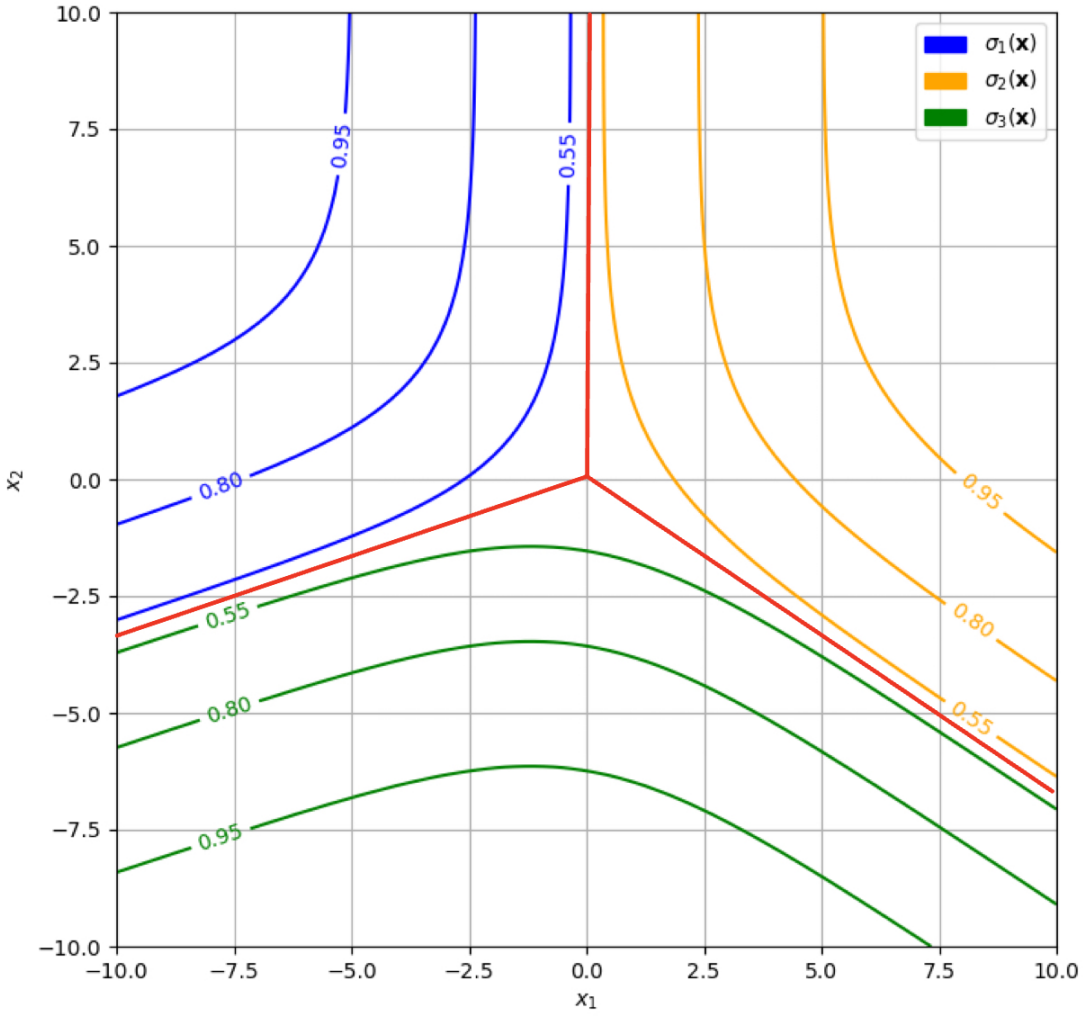
$$\sigma(\mathbf{x}) = \begin{pmatrix} \sigma_1(\mathbf{x}) \\ \sigma_2(\mathbf{x}) \\ \sigma_3(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \\ \\ \\ \\ \\ \\ \end{pmatrix}$$

Solution:

$$\sigma(\mathbf{x}) = \begin{pmatrix} \sigma_1(\mathbf{x}) \\ \sigma_2(\mathbf{x}) \\ \sigma_3(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \frac{\exp w_1^T \mathbf{x}}{\exp w_1^T \mathbf{x} + \exp w_2^T \mathbf{x} + \exp w_3^T \mathbf{x}} \\ \frac{\exp w_2^T \mathbf{x}}{\exp w_1^T \mathbf{x} + \exp w_2^T \mathbf{x} + \exp w_3^T \mathbf{x}} \\ \frac{\exp w_3^T \mathbf{x}}{\exp w_1^T \mathbf{x} + \exp w_2^T \mathbf{x} + \exp w_3^T \mathbf{x}} \end{pmatrix}$$

2. **Task (5 Points)** Sketch the decision boundaries (pairwise between all classes) into the figure below. Justify the shape of your sketch.

Solution:



Let $(\theta_1, \theta_2, \theta_3)$ be the logits in a logistic regression problem. For the decision boundaries there are three cases to distinguish:

- $\theta_2, \theta_3 \geq \theta_1$:

$$1 = \frac{\frac{\exp\theta_2}{\sum \exp\theta_i}}{\frac{\exp\theta_3}{\sum \exp\theta_i}} = \frac{\exp\theta_2}{\exp\theta_3}$$

- $\theta_1, \theta_3 \geq \theta_2$
- $\theta_1, \theta_2 \geq \theta_3$

for each case the decision boundary is **linear**.

- 1.5 points if the sketched boundaries are linear in the area where the contour lines already suggest linearity (0.5 points for each case)
- 1.5 points for linearity in the area $[-2.5, 2.5] \times [-2.5, 2.5]$.
- 2 points if there is also an explanation with the log-odds

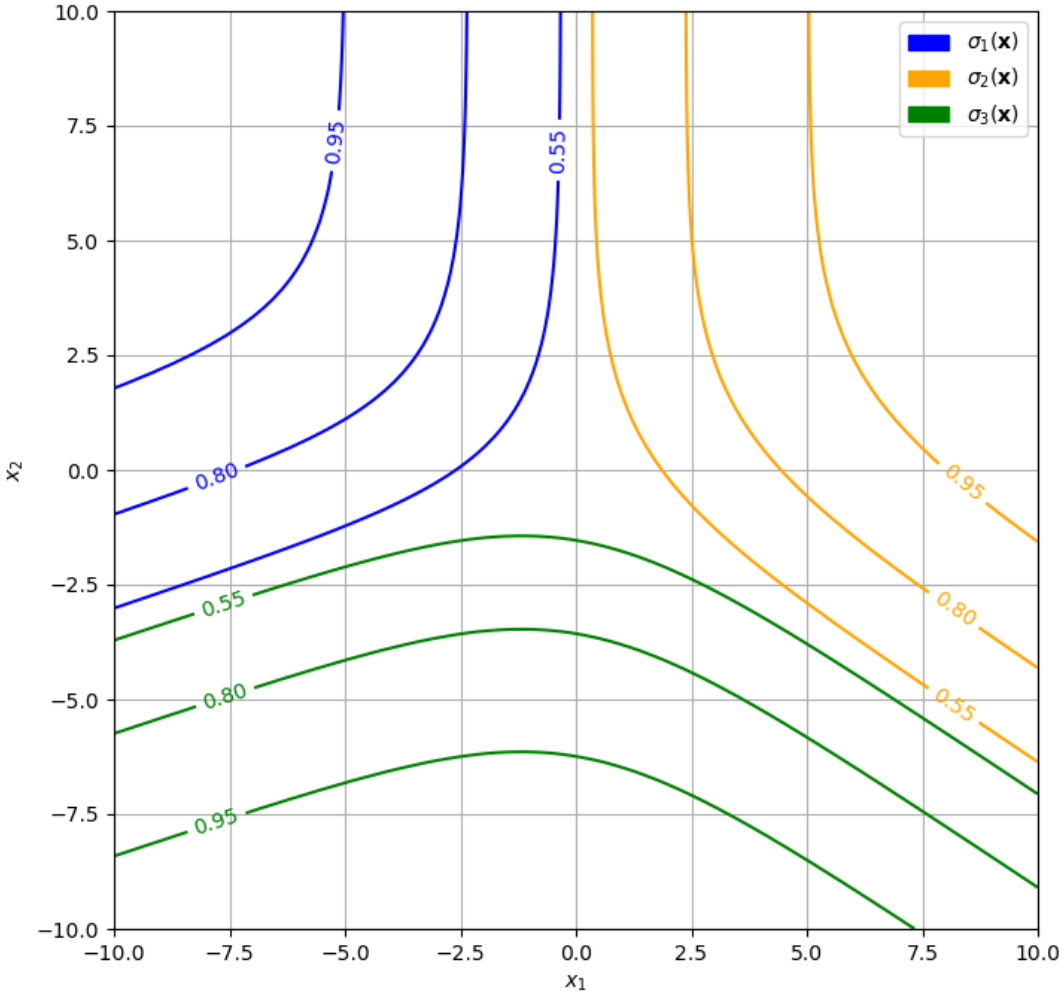


Figure 2 A logistic regression problem with three classes.

Task 6: SVM (10 Points)

You are given a training dataset, as shown in Fig 3. For this problem, assume that we are training a soft-margin SVM with a quadratic kernel. Recall that the soft-margin SVM problem is:

$$\min \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{n}\xi_i \tag{1}$$

$$\text{subject to } \forall i = 1, \cdots, n\text{:}$$

$$\xi_i \geq 0 \tag{2}$$

$$(\mathbf{w}\cdot\mathbf{x}_i + b)y_i - (1-\xi_i) \geq 0. \tag{3}$$

1. **Task (3 Points)** Which curve best represents the decision boundary when $C = \infty$? Why? The decision boundaries are shown in Fig 3. Note: 1 point for the choice of the decision boundary and 2 points for the justification.

Solution:
The decision boundary is the curve (a). When $C = \infty$, it becomes a hard margin SVM, hence the decision boundary must separate the two classes. We can conclude that the decision boundary is like curve (a) as it enforces all constraints (page 29 - SVM slides).

2. **Task (3 Points)** Which curve best represents the decision boundary for a very small value of C, but not zero? Why? The decision boundaries are shown in Fig 3. Note: 1 point for the choice of the decision boundary and 2 points for the justification.

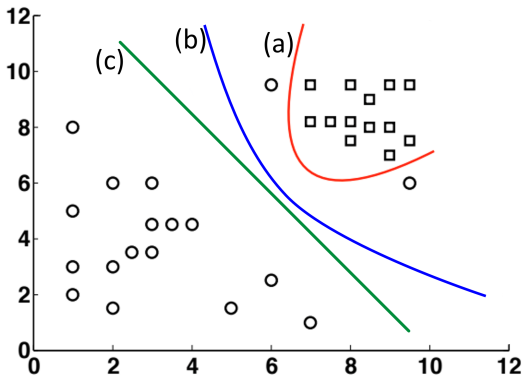


Figure 3 Training dataset for SVM

Solution:
In the figure, the most appropriate decision boundary is the curve (b). When the value of C is small but not zero, we would expect the decision boundary to balance, maximizing the margin and permitting some misclassification.

3. **Task (4 Points)** Consider a support vector machine whose input space is $\mathbb{R}^2$, and in which the inner products are computed by means of the kernel.

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^2 - 1$$

Show that the mapping to feature space that is implicitly defined by this kernel is the mapping to $\mathbb{R}^5$ given by:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix}$$

Solution:

The kernel function is given by:

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^2 - 1$$

Expand the kernel function:

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^2 - 1 = (\mathbf{x} \cdot \mathbf{y})^2 + 2(\mathbf{x} \cdot \mathbf{y}) + 1 - 1 = (\mathbf{x} \cdot \mathbf{y})^2 + 2(\mathbf{x} \cdot \mathbf{y})$$

To determine the feature mapping $\phi(\mathbf{x})$, consider:

$$\phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix}$$

Compute the inner product $\phi(\mathbf{x}) \cdot \phi(\mathbf{y})$:

$$\phi(\mathbf{x}) \cdot \phi(\mathbf{y}) = x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 + 2x_1y_1 + 2x_2y_2$$

This simplifies to:

$$\phi(\mathbf{x}) \cdot \phi(\mathbf{y}) = (x_1y_1 + x_2y_2)^2 + 2(x_1y_1 + x_2y_2) = (\mathbf{x} \cdot \mathbf{y})^2 + 2(\mathbf{x} \cdot \mathbf{y})$$

Thus, we have shown that the feature mapping $\phi(\mathbf{x})$ indeed transforms the input space such that the kernel $k(\mathbf{x}, \mathbf{y})$ corresponds to the inner product in this transformed feature space. The given kernel function can be represented as the dot product in the 5-dimensional feature space defined by:

$$\phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix}$$

Task 7: Feed-forward neural network (28 Points)

The figure below depicts a feedforward neural network designed for regression. The network includes the following components:

- Input Layer: Three input features x_1, x_2, and x_3.
- Hidden Layers: The first hidden layer consists of a single node representing a scalar v. The second hidden layer h consists of two nodes, each representing a scalar, i.e., h_1 and h_2. The activation function is ReLU.
- Output Layer: A single output node y.

Each node in the hidden and output layers has an associated bias term.

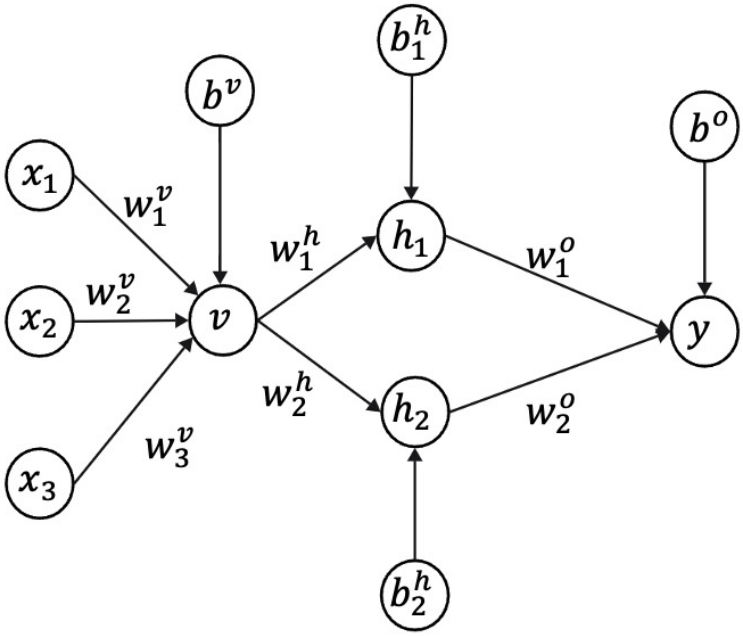


Figure 4 Feed-forward neural network.

Suppose we initialize this network with the following parameters:

$$W^v = \begin{bmatrix} w_1^v & w_2^v & w_3^v \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$b^v = \begin{bmatrix} 1 \end{bmatrix}$$

$$W^h = \begin{bmatrix} w_1^h \\ w_2^h \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$b^h = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$W^o = \begin{bmatrix} w_1^o & w_2^o \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$b^o = 1$$

1. **Task (3 Points)** Write down the output formula for v, h and y.

Solution:

$$v = \text{ReLU}(W^v x + b^v)$$
$$h = \text{ReLU}(W^h v + b^h)$$
$$\hat{y} = W^o h + b^o$$

2. **Task (5 Points)** Calculate the forward pass for $x = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}^T$. Show each computation step.

Solution:

$$v = \text{ReLU}(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + 1)$$
$$= 4$$
$$h = \text{ReLU}(\begin{bmatrix} 1 \\ -1 \end{bmatrix} \times 4 + \begin{bmatrix} 1 \\ 1 \end{bmatrix})$$
$$= \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$
$$\hat{y} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \end{bmatrix} + 1$$
$$= 6$$

3. **Task (2 Points)** Choose an appropriate loss function $L(\hat{y}, y)$ for this task and provide a formal definition.

Solution:
Mean square error.

$$L(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$$

4. **Task (1 Point)** Compute $L(\hat{y}, y)$ given the true output $y = 5$.

Solution:

$$\begin{aligned} L(\hat{y}, y) &= (\hat{y} - y)^2 \\ &= (6 - 5)^2 \\ &= 1 \end{aligned}$$

5. **Task (1 Point)** Calculate $\frac{\partial L(\hat{y}, y)}{\partial \hat{y}}$.

Solution:

$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y) = 2$$

6. **Task (1 Point)** Mark the gradient descent path in Figure 4 if we want to adjust the weight w_2^v. The

gradient descent path refers to the sequence of neurons or layers in the neural network through which the error is propagated backward to update the weights.

Solution:

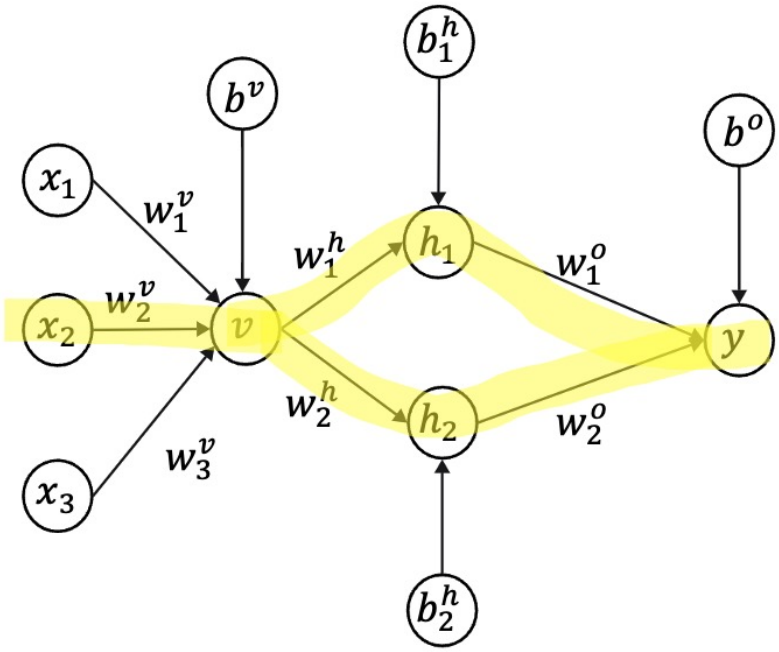


Figure 5 Feed-forward neural network.

7. **Task (6 Points)** Calculate $\frac{\partial h}{\partial v}$.

Solution:

$$\frac{\partial h}{\partial v} = \frac{\partial \text{ReLU}(W^h v + b^h)}{\partial (W^h v + b^h)} \cdot \frac{\partial (W^h v + b^h)}{\partial v}$$

$$z = W^h v + b^h = \begin{bmatrix} 1 \\ -1 \end{bmatrix} 4 + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ -3 \end{bmatrix}$$

$$\frac{\partial h}{\partial v} = \frac{\partial \text{ReLU}(z)}{\partial z} \cdot W^h$$

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

8. **Task (4 Points)** Given $\frac{\partial \hat{y}}{\partial h} = W^o = \begin{bmatrix} 1 & 1 \end{bmatrix}$, $\frac{\partial v}{\partial w_2^v} = x_2 = 1$, calculate the gradient of w_2^v with respect to $L(\hat{y}, y)$

Solution:

$$\frac{\partial L}{\partial w_2^v} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} \cdot \frac{\partial h}{\partial v} \cdot \frac{\partial v}{\partial w_2^v}$$

$$= 2 \times \begin{bmatrix} 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \times 1$$

$$= 2$$

9. **Task (2 Points)** Adjust the parameters w_2^v using gradient descent and a learning rate of $\alpha = 0.1$.

Solution:

$$w_2^v := w_2^v - \alpha \times \frac{\partial \mathcal{L}}{\partial w_2^v} = 2 - 0.1 \times 2 = 1.8$$

10. **Task (3 Points)** What could happen if the input $x = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix}^T$? How could you solve the problem?

Solution:

The ReLU activation function outputs the value itself if it is positive, otherwise zero. If we feed this input into the given neural network, neurons will output zeros through layers, and thus, the weights can not be updated. To solve this problem, we can modify the initialization or use another activation function like Leaky ReLU that allows a small gradient when the unit is inactive.