Machine Learning
Solutions

Mock Exam

Note

Answer ALL questions. Each question will be marked out of 20. The paper will be marked out of 60, which will be rescaled to a mark out of 100.

The end of the paper has an appendix with some formulas and definitions that you may find useful.

Question 1 Core Concepts

(a) Explain what is a supervised learning algorithm, including its core goal. Please give your answer formally, making use of appropriate mathematical symbols and terminology whenever relevant. **[5 marks]**

(b) Answer the following questions regarding feature transformations in the context of machine learning:

- What is a non-linear feature transformation? Please provide a detailed definition.
- When could it be useful to adopt a non-linear feature transformation?

[5 marks]

(c) Question covering Jian's part of the module to be added here, but the format / style of these questions will be similar to the kind of questions asked above.

[5 marks]

(d) Question covering Jian's part of the module to be added here, but the format / style of these questions will be similar to the kind of questions asked above.

[5 marks]

Model answer / LOs / Creativity:

(a) A supervised learning algorithm is an algorithm that takes as input a training set containing pairs of inputs and target outputs

$$\mathcal{T} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

[1 mark]

where $(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y}$ are drawn i.i.d. (independently and identically distributed) from a fixed albeit unknown joint probability distribution $p(\mathbf{x}, y)$, $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. [1 mark]

It then learns a function $g : \mathcal{X} \to \mathcal{Y}$ with the goal of being able to predict (generalise to) unseen (test) examples of the same probability distribution $p(\mathbf{x}, y)$. [3 marks]

PS: some supervised learning algorithms will instead learn a function g that approximates the probability distribution $p(\mathbf{x}, y)$ itself. So, an answer stating this would also be correct. When marking this question, you would get partial marks for each component of the answer as indicated above. Around half of these partial marks would come from appropriate use of mathematical notation. Note also that there are different ways to define the same thing. The wording and mathematical symbols does not necessarily need to match the ones above for the question to be marked as correct.

(b)

- A non-linear feature transformation is a vector function that transforms the input space of the problem using at least one non-linear function. In particular, a feature transformation is a vector function Φ that receives an input vector $\mathbf{x}$, where $\mathbf{x} \in R^d$, and $d \geq 1$ is the dimensionality of the input space of the problem. In a non-linear feature transformation, at least one of the functions $\phi(\mathbf{x})_i \in \Phi(\mathbf{x})$ is a non-linear function. [3 marks]

  PS: the vector function can be of any dimensionality $d' \geq 1$, i.e., it is not a requirement to satisfy $d = d'$.

  PS2: when marking this question, you would get half marks if you only provide something like the first sentence above. Full marks would require a detailed definition. Please note that there are different ways to define the same concept. So, the wording in your answer would not need to match the wording above exactly.

- It could be useful to adopt a non-linear transformation when the problem is non-linear but we wish to use a linear model to solve it. In particular, a non-linear feature transformation could potentially transform the problem into a feature space where the problem is linear, such that a linear model on the feature space could be used to solve it. [2 marks]

  PS: the answer above is the key idea that we've learned in the module. However, there might be different possible answers. If a student gave a different but correct answer, it would also receive full marks for this question.

(c)

(d)

Question 2 Classification

(a) Consider a machine learning problem with two parameters to be learned (w_1 and w_2) and the following loss function:

$$E(\mathbf{w}) = w_1^2 + 0.001 w_2^2,$$

where $w_1 \in R$ and $w_2 \in R$.

- Explain in detail why Gradient Descent could be inefficient to minimise this function. **[5 marks]**
- Explain in detail why standardisation of the input variables (e.g., by deducting the mean from each input variable and then dividing each input variable by the standard deviation) could be potentially helpful to improve the efficiency of Gradient Descent for this problem. **[5 marks]**

(b) In the dual representation of the Support Vector Machines, it could happen that a training example is on the margin, but is associated to a Lagrange multiplier of zero. Despite being on the margin, this training example is not considered as a support vector, as it would not contribute towards the predictions made by the model. Explain in detail why an example that is on the margin could possibly have a Lagrange multiplier of zero.

Hint: you can explain that by reflecting about the steps to go from the primal to the dual representation. **[10 marks]**

Model answer / LOs / Creativity:

(a)
- This function has a much larger scalar being multiplied by w_1 than by w_2. If one were to plot it, it would form an elliptical bowl where the gradient in the direction of w_1 is much steeper than in the direction of w_2, i.e., $\frac{\partial g}{\partial w_1} > \frac{\partial g}{\partial w_2}$.
  As a result, given the Gradient Descent weight update rule $\mathbf{w} = \mathbf{w} - \eta \nabla E(\mathbf{w})$ and a fixed learning rate η, the size of the weight update for w_1 will be larger than that for w_2.
  The larger weight update for w_1 may result in the algorithm jumping across the optimum in the direction of w_1, while the slower weight update for w_2 would result in the algorithm giving small steps. This could result in a long time to find the optimum.
  PS: this is similar to what we've seen in the example given in lecture 2a, though the elipse is in the vertical direction in this question. An answer would get 5 marks if it's entirely correct, and 3 marks if it has correct statements but with omissions or small mistakes, and 1 mark if it is mostly wrong.

- The reason why standardisation could help is related to the reason why the scalar multiplying w_1 is larger than the one multiplying w_2. In particular, the different sizes of the scalars could be due to a different scale for the input variables x_1 and x_2.

  This is because the loss function is obtained by calculating the error on the training examples, which in turn is based on the predictions. For instance, assume that the predictions are given based on a function $h(\mathbf{x}) = x_1 w_1^2 + x_2 w_2^2$. A x_1 that has a larger scale than x_2 could result in a larger scalar being multiplied by w_1 than by w_2 in the loss function.

  Standardising the input variables will result in them being in the same scale, such that the scalars multiplying each weight would hopefully be more similar, resulting in more similar gradients. The more similar gradients would in turn reduce the problem mentioned in the previous item of this answer.

  PS: An answer would get 5 marks if it's entirely correct, 3 marks if it has correct statements but with omissions or small mistakes, and 1 mark if it is mostly wrong.

(b) Note the exam's annex. The dual representation is created by starting with the primal representation and then creating a penalty function $g(\mathbf{w}, b)$ to deal with the constraints.

When a training example n is on the margin, the constraint $y^{(n)} h(\mathbf{x}^{(n)}) \geq 1$ is satisfied with the equality. As a result, the term $1 - y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}) + b)$ in the penalty function g is equal to zero.

In such situation, the value of $a^{(n)}$ does not matter to maximise $g(\mathbf{w}, b)$. The value of $a^{(n)}$ can be any value $a^{(n)} \geq 0$. If $a^{(n)} > 0$ the training example n is a support vector (it will contribute to predictions). However, as $a^{(n)}$ can take any value ≥ 0, the training example n could also be associated to a value of $a^{(n)} = 0$, which is the case considered in this question.

PS: An answer would get 5 marks if it's entirely correct, 3 marks if it has correct statements but with omissions or small mistakes, and 1 mark if it is mostly wrong.

Question 3

(a) Question to be added based on Jian's content. **[10 marks]**

(b) Question to be added based on Jian's content. **[10 marks]**

Model answer / LOs / Creativity:

(a)

(b)

Appendix

Primal representation of hard margin support vector machines

$$\underset{\mathrm{w},b}{\operatorname{argmin}} \left\{ \tfrac{1}{2} \|\mathbf{w}\|^2 \right\}$$

$$\text{subject to} \quad y^{(n)} h(\mathbf{x}^{(n)}) \geq 1, \quad \forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T},$$

where $\mathbf{w}$ and b are the parameters to be learned, $\mathbf{x}^{(i)} \in R^d$ are the input variables of example i, d is the number of input variables, $y^{(i)} \in \{-1, 1\}$ is the output label of example i, $\mathcal{T}$ is the training set, $h(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$, and $\phi(\mathbf{x})$ is a feature embedding. When $\phi(\mathbf{x}) = \mathbf{x}$, no embedding is being used.

Intermediate step between primal and dual hard margin support vector machines

$$\operatorname{argmin}_{\mathrm{w},b} \left\{ \tfrac{1}{2} \|\mathbf{w}\|^2 + g(\mathbf{w}, b) \right\}$$

$$\text{where} \qquad g(\mathbf{w}, b) = \max_{\mathrm{a}} \textstyle\sum_{n=1}^{N} a^{(n)} (1 - y^{(n)} (\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b))$$

$$\text{subject to} \quad a^{(n)} \geq 0, \ \forall n \in \{1, \cdots, N\}$$

where $a^{(i)}$ is the Lagrange multiplier associated to training example i, N is the number of training examples, and $\mathbf{x}^{(i)} \in R^d$ are the input variables of example i.

Dual representation of hard margin support vector machines

$$\underset{\mathrm{a}}{\operatorname{argmax}} \tilde{L}(\mathbf{a}) = \textstyle\sum_{n=1}^{N} a^{(n)} - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a^{(n)} a^{(m)} y^{(n)} y^{(m)} k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)})$$

$$\text{subject to} \quad a^{(n)} \geq 0, \ \forall n \in \{1, \cdots, N\} \text{ and } \textstyle\sum_{n=1}^{N} a^{(n)} y^{(n)} = 0,$$

where $k(\cdot, \cdot)$ is the kernel function.