The Australian National University
School of Computing

COMP3670/6670 Introduction to Machine Learning
Semester 2, 2021

Final Exam

- Write your name and UID on the first page (you will be fine if you forget to write them).
- This is an open book exam. You may bring in any materials including electronic and paper-based ones. Any calculators (programmable included) are allowed. No communication devices are permitted during the exam.
- Reading time: 30 minutes
- Writing time: 180 minutes
- For all the questions, write your answer CLEARLY on papers prepared by yourself.
- There are totally 9 pages (including the cover page)
- Points possible: 100
- This is not a hurdle.
- When you are asked to provide a justification to your answer, if your justification is incorrect, you will get 0.

- **Section 1. Linear Algebra and Matrix Decomposition** (13 points)

1. (6 points) Show that it is impossible to have a set of 3 orthogonal **non-zero** vectors $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ in $\mathbb{R}^2$.
(Hint: Write $\mathbf{v}_1 = [x_1, y_1]^T$ and first assume that $x_1 = 0$. Then, prove it for the case where $x_1 \neq 0$, using the first case to help you.)
Solution. We write
$$\mathbf{v}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad \mathbf{v}_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \quad \mathbf{v}_3 = \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$$
As the hint suggests, we first assume that $x_1 = 0$. Then since $\mathbf{v}_1 \cdot \mathbf{v}_2 = 0$, we have $y_1 y_2 = 0$. Since y_1 cannot be zero (otherwise $\mathbf{v}_1 = \mathbf{0}$) we must have $y_2 = 0$. Since $\mathbf{v}_1 \cdot \mathbf{v}_3 = 0$, we have $x_1 x_3 + y_1 y_3 = y_1 y_3 = 0$, and $y_1 \neq 0$, so $y_3 = 0$. Since $\mathbf{v}_2 \cdot \mathbf{v}_3 = 0$, we have $x_2 x_3 + y_2 y_3 = x_2 x_3 = 0$. Now, $y_2 = 0$, so $x_2 \neq 0$ (else $\mathbf{v}_2 = \mathbf{0}$), so $x_3 = 0$. We have that $x_3 = 0$ and $y_3 = 0$, so $\mathbf{v}_3 = \mathbf{0}$, a contradiction.
Now, if any of the $x_1, x_2, x_3, y_1, y_2, y_3$ were zero, this would also result in a contradiction in the same way, as by symmetry we can reorder the vectors so the zero was somewhere in the first vector, and since $x_1 y_1 + x_2 y_2 = y_1 x_1 + y_2 x_2$, we could swap the x and y values in all vectors so that $x_1 = 0$, from which the contradiction follows by above.
Now, assume that $x_1 \neq 0$. $\mathbf{v}_1 \cdot \mathbf{v}_2$ gives
$$\begin{aligned} \mathbf{v}_1 \cdot \mathbf{v}_2 &= 0 \\ x_1 x_2 + y_1 y_2 &= 0 \\ x_2 &= \frac{-y_1 y_2}{x_1} \end{aligned}$$
and similarly
$$\begin{aligned} \mathbf{v}_1 \cdot \mathbf{v}_3 &= 0 \\ x_1 x_3 + y_1 y_3 &= 0 \\ x_3 &= \frac{-y_1 y_3}{x_1} \end{aligned}$$
Then $\mathbf{v}_2 \cdot \mathbf{v}_3 = 0$, so
$$\begin{aligned} x_2 x_3 + y_2 y_3 &= 0 \\ \left(\frac{-y_1 y_2}{x_1}\right)\left(\frac{-y_1 y_3}{x_1}\right) + y_2 y_3 &= 0 \\ \frac{y_1^2 y_2 y_3}{x_1^2} + y_2 y_3 &= 0 \\ y_2 y_3 \left(\frac{y_1^2}{x_1^2} + 1\right) &= 0 \end{aligned}$$
Then either $y_2 = 0$ (contradiction), or $y_3 = 0$ (contradiction) or $\frac{y_1^2}{x_1^2} + 1 = 0$, giving $y_1^2 = -x_1^2$. Since the left hand side is non-negative, and the right hand side is non-positive, the only way to satisfy this equation is $y_1 = x_1 = 0$, a contradiction.

2. (7 points) Consider an arbitrary matrix $\mathbf{A} \in \mathbb{R}^{2\times 2}$
$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$
We would like $\mathbf{A}$ to satisfy the following properties:
(a) $\mathbf{A}$ is non-invertible.
(b) $\mathbf{A}$ is symmetric.

(c) All the entries of $\mathbf{A}$ are positive.

(d) $\mathbf{A}$ has a positive eigenvalue $\lambda = p > 0$.

Find the set of all matrices $\mathbf{A}$ (as a function of p) that satisfy all the above constraints.

Solution. $\mathbf{A}$ is symmetric, so $b = c$. We can eliminate c and consider only matricies of the form

$$\mathbf{A} = \begin{bmatrix} a & b \\ b & d \end{bmatrix}.$$

$\mathbf{A}$ is non-invertible, so $ad - b^2 = 0$. Also, $a, b, d > 0$. We have that $p > 0$ is an eigenvalue, so we find all eigenvalues using the characteristic polynomial.

$$\begin{aligned} (a-\lambda)(d-\lambda) - b^2 &= 0 \\ ad - a\lambda - d\lambda + \lambda^2 - b^2 &= 0 \\ -a\lambda - d\lambda + \lambda^2 &= 0 \\ \lambda(\lambda - a - d) &= 0 \end{aligned}$$

So $\lambda = 0$ or $\lambda = a + d$. Since 0 cannot be positive, we have that $p = a + d$, or $d = p - a$. We can eliminate d.

$$\mathbf{A} = \begin{bmatrix} a & b \\ b & p-a \end{bmatrix}$$

Finally, since $ad - b^2 = 0$, we have $a(p-a) = b^2$, and hence $b = \pm\sqrt{a(p-a)}$. Since all entries need to be positive, we take the positive root, $b = \sqrt{a(p-a)}$. For the square root to be defined (and not zero), we need $a(p-a) > 0$, which implies that $0 < a < p$. Hence, the set of all possible $\mathbf{A}$ is given by

$$\left\{ \begin{bmatrix} a & \sqrt{a(p-a)} \\ \sqrt{a(p-a)} & p-a \end{bmatrix} : 0 < a < p \right\}$$

as required.

- **Section 2. Analytic Geometry and Vector Calculus** (12 points)

1. (6 points) Find all matrices $\boldsymbol{T} \in \mathbb{R}^{2\times 2}$ such that for any $\boldsymbol{v} \in \mathbb{R}^2$,

$$\boldsymbol{v}^T\boldsymbol{v} = (\boldsymbol{T}\boldsymbol{v})^T\boldsymbol{v} = (\boldsymbol{T}\boldsymbol{v})^T\boldsymbol{T}\boldsymbol{v}$$

Solution. Consider $\boldsymbol{v}\cdot\boldsymbol{v} = \boldsymbol{T}(\boldsymbol{v})\cdot\boldsymbol{v}$. Let $\boldsymbol{v} = [x, y]^T$. Then

$$\begin{aligned}
\boldsymbol{v}\cdot\boldsymbol{v} &= \boldsymbol{T}(\boldsymbol{v})\cdot\boldsymbol{v} \\
x^2 + y^2 &= x(ax+by) + y(cx+dy) \\
x^2 + y^2 &= ax^2 + (b+c)xy + dy^2
\end{aligned}$$

By choosing $x = 0, y = 1$, we obtain $d = 1$, and by choosing $x = 1, y = 0$ we obtain $a = 1$. Then, we choose $x = 1$ and leave y arbitrary for the moment. Substituting, we obtain

$$1 + y^2 = 1 + (b+c)y + y^2$$

from which we obtain $(b+c)y = 0$ for all y, so $c = -b$. We now consider the other equation

$$\begin{aligned}
\boldsymbol{v}\cdot\boldsymbol{v} &= \boldsymbol{T}(\boldsymbol{v})\cdot\boldsymbol{T}(\boldsymbol{v}) \\
x^2 + y^2 &= (ax+by)^2 + (cx+dy)^2 \\
x^2 + y^2 &= (a^2+c^2)x^2 + (b^2+d^2)y^2 + 2(ab+cd)xy \\
x^2 + y^2 &= (1+b^2)x^2 + (1+b^2)y^2 + 2(b-b)xy \\
x^2 + y^2 &= (1+b^2)(x^2+y^2) \\
1 &= 1 + b^2 \\
b &= 0
\end{aligned}$$

So $a = 1, b = 0, c = -b = 0, d = 1$. So the only matrix that satisfies this property is the identity.

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2. (6 points) Let $\boldsymbol{x}, \boldsymbol{a} \in \mathbb{R}^{n\times 1}$, and define $f : \mathbb{R}^{n\times 1} \to \mathbb{R}$ as

$$f(\boldsymbol{x}) = 3\exp(2\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x})$$

where $\exp(x) := e^x$, the exponential function. Compute $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$.

Solution. We apply the chain rule. Note that $f(\boldsymbol{x}) = g(h(\boldsymbol{x}))$, where $g : \mathbb{R} \to \mathbb{R}$, $g(h) = 3\exp(2h)$ and $h : \mathbb{R}^n \to \mathbb{R}$, $h(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$. We have by lectures/assignment that $\nabla_{\mathbf{x}} h(\mathbf{x}) = \mathbf{x}^T(\mathbf{A} + \mathbf{A}^T)$, and by calculus we have that $\nabla_h g(h) = 6\exp(2h)$. Therefore

$$\frac{df}{d\mathbf{x}} = \frac{dg}{dh}\frac{dh}{d\mathbf{x}} = 6\exp(2h)\mathbf{x}^T(\mathbf{A}+\mathbf{A}^T) = 6\exp 2(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x})\mathbf{x}^T(\mathbf{A}+\mathbf{A}^T)$$

- **Section 3. Probability** (15 points)

  Consider the following scenario. I have two boxes, box A and box B. In their initial configuration, Box A contains 3 red balls and 3 white balls; Box B contains 6 red balls.

  We swap the balls in boxes via the following procedure.

  Procedure 1: *We draw a ball x_A uniformly at random from box A, and draw a ball x_B uniformly at random from box B. We place ball x_A into box B and place ball x_B into box A.*

  Let R_A and R_B be random variables representing the number of **red balls** in box A and box B respectively.

  1. (3 points) Describe the probability mass functions $p(R_a = n)$ and $p(R_b = n)$ for all integers $n \geq 0$ for the initial configuration of the boxes.
     Solution. Clearly, box A contains 3 red balls, so
     $$p(R_A = n) = \begin{cases} 1 & n = 3 \\ 0 & n \neq 3 \end{cases}$$
     and box B contains 6 red balls, so
     $$p(R_B = n) = \begin{cases} 1 & n = 6 \\ 0 & n \neq 6 \end{cases}$$

  2. (3 points) After performing Procedure 1, what is the new joint probability mass functions $p(R_A = n_a, R_B = b_n)$?
     Solution. We are guaranteed to move a red ball from box B into box A, it is equally likely that the ball x_A chosen from A is red. So, we have a 0.5 probability of nothing changing. and a probability of 0.5 of adding a red ball to box A and loosing a red ball from box B. Hence,
     $$p(R_A = n_a, R_B = n_b) = \begin{cases} 0.5 & (n_a, n_b) = (3, 6) \\ 0.5 & (n_a, n_b) = (4, 5) \\ 0 & \text{else} \end{cases}$$

  We now describe a new method of swapping balls between boxes.

  Procedure 2: *We draw a ball x_A uniformly at random from box A, and place it in box B. Then, we draw a ball x_B uniformly at random from box B, and place it in box A.*

  3. (2 points) Is Procedure 2 equivalent to Prodecure 1? Why or why not? Use no more than 3 sentences to explain your answer.
     Solution. No, as we place x_A into Box B before selecting a ball from box B. This means it is possible (though unlikely) that the ball drawn from B was the same ball moved there from A, so it is possible that x_B is black (which was impossible in the previous case).
  4. (3 points) We reset the experiment to the initial configuration, and then perform Procedure 2.
     (a) Compute the probability x_A is white.
     (b) Compute the probability that x_B is white.

     Solution. Box A has the same number of red and white balls, so $p(x_A = \text{white}) = 1/2$. Now, the outcome of x_B depends on what x_A was, so we use the sum and product rules.

     $$p(x_B = \text{white}) = p(x_B = \text{white}|x_A = \text{white})p(x_A = \text{white}) + p(x_B = \text{white}|x_A = \text{red})p(x_A = \text{red})$$

If x_A = white, then box B now has 6 red balls and 1 white ball, so $p(x_B = \text{white}|x_A = \text{white}) = 1/7$. If x_A = red, then box B still contains only red balls, so $p(x_B = \text{white}|x_A = \text{red}) = 0$. Hence,

$$p(x_B = \text{white}) = 1/7 \times 1/2 + 0 \times 1/2 = 1/14$$

5. (4 points) We reset the experiment to the initial configuration, and then perform **Procedure 1.** After that, we select either box A or box B by some uniformly random procedure (like flipping a **fair** coin), and draw a ball from the selected box. The ball drawn was **white**. What was the probability that box B was selected?

Solution.

$$p(\text{Box} = B|\text{ball} = w)$$
$$= \frac{p(\text{ball} = w|\text{Box} = B)p(\text{Box} = B)}{p(\text{ball} = w|\text{Box} = A)p(\text{Box} = A) + p(\text{ball} = w|\text{Box} = B)p(\text{Box} = B)}$$

First, consider $p(\text{ball} = w|\text{Box} = B)$. If box B were selected, there is a $1/2$ chance of it containing 5 red balls out of 6, or $1/2$ chance of 6 red balls out of 6. So,

$$p(\text{ball} = w|\text{Box} = B) = 1/2 \times 5/6 + 1/2 \times 1 = 11/12$$

If box A were selected, there is a $1/2$ chance of it containing 3 red balls out of 6, or $1/2$ chance of 4 red balls out of 6.

$$p(\text{ball} = w|\text{Box} = A) = 1/2 \times 3/6 + 1/2 \times 4/6 = 7/12$$

Hence.

$$p(\text{Box} = B|\text{ball} = w)$$
$$= \frac{11/12 \times 1/2}{7/12 \times 1/2 + 11/12 \times 1/2} = 11/18$$

- **Section 4. Clustering and Gaussian Mixture Model (GMM)** (16 points)

  Suppose you have N data points in $\mathbb{R}$: $x_1, x_2, ..., x_n, x_{n+1}, ..., x_N$. Now you want to partition them into two clusters C_1 and C_2. First, you assign $x_1, x_2, ..., x_n$ into C_1 and assign $x_{n+1}, ..., x_N$ into C_2. After that, you want to calculate the new cluster centers μ_1 and μ_2 of C_1 and C_2. For each cluster, your objective is to minimise the averaged squared distances between each data point and its assigned center.

  1. (3 points) Calculate μ_1 and μ_2 that achieve your objective (M-step). (Only showing result without the optimisation process will lead to 0 mark.) Hint: after the assignment, you can only use the first n points to calculate μ_1, and the rest points to calculate μ_2.

     Solution. We denote the two centers as μ_1 and μ_2, respectively. The loss function can be written as:

     $$L = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_1)^2 + \frac{1}{N-n}\sum_{i=n+1}^{N}(x_i - \mu_2)^2.$$

     We then calculate the partial derivatives of L w.r.t μ_1 and μ_2. We have

     $$\frac{\partial L}{\partial \mu_1} = \frac{2}{n}\sum_{i=1}^{n}(\mu_1 - x_i),$$

     $$\frac{\partial L}{\partial \mu_2} = \frac{2}{n}\sum_{i=n+1}^{N}(\mu_2 - x_i).$$

     We set both partial derivatives to 0. We can easily get

     $$\mu_1 = \frac{1}{n}\sum_{i=1}^{n} x_i, \mu_2 = \frac{1}{N-n}\sum_{i=n+1}^{N} x_i.$$

  2. (2 points) Are μ_1 and μ_2 a global minimum solution to this M-step (calculating the means of clusters)? Use no more than 3 sentences to explain your answer.

     Solution. Yes. Reason 1: the assignment strategy (first n points to C_1 and the rest to C_2) is given. Reason 2: the optimization of μ_1 does not depend on μ_2 (does not assume given values of μ_2), and the optimization of μ_2 does not depend on μ_1 (does not assume given values of μ_1).

  Given μ_1 and μ_2 calculated in Question 1 above, you now want to update the assignment (E-step). This time again you aim to minimise the sum of squared distances between each data point and its center.

  3. (3 points) What is your optimal assignment strategy (you can only assign a data point to one cluster, *i.e., hard assignment*)? Why is it optimal for your aim? (You can use whatever is helpful to illustrate, such as figures and maths. Only showing the result without a clear demonstration/proof process with lead to 0 mark)

     Solution. We should assign each point to its closest center.

     To prove, we only have to optimize the assignment of each individual data point, because the data points are independently and identically distributed (i.i.d). For data point x_i, because of hard assignment, the loss can be written as,

     $$L_h = \begin{cases} (x_i - \mu_1)^2, & \text{if } x_i \text{ is assigned to center 1} \\ (x_i - \mu_2)^2, & \text{if } x_i \text{ is assigned to center 2.} \end{cases}$$

     Apparently, L_h is minimised when our assignment strategy allows L to always take the lesser between $(x_i - \mu_1)^2$ and $(x_i - \mu_2)^2$. In other words, our assignment strategy should always satisfy the lesser between $(x_i - \mu_1)^2$ and $(x_i - \mu_2)^2$ is chosen. That is, x_i should be assigned to the center it has a closer distance with.

4. (2 points) Is this assignment a global minimum solution to the E-step under hard assignment? Use no more than 3 sentences to explain your answer.

   Solution. Yes. This is because μ_1 and μ_2 are considered as given. Moreover, hard assignment is used. Under these conditions, the optimization of the assignment strategy gives a global minimum to the E-step.

Following the above questions, we do further explorations. The GMM performs soft assignment, *i.e.,* assigning a data point into multiple clusters, and this assignment is accompanied by responsibilities. Now, we want to explore a similar scheme in k-means. Specifically, we define

$$r_{m2} = \frac{(x_m - \mu_1)^2}{(x_m - \mu_1)^2 + (x_m - \mu_2)^2}, r_{m1} = \frac{(x_m - \mu_2)^2}{(x_m - \mu_1)^2 + (x_m - \mu_2)^2},$$

where r_{m2} denotes how likely x_m belongs to C_2, and r_{m1} denotes how likely x_m belongs to C_1.

1. (2 point) Suppose x_m is assigned to C_1 under *hard assignment*, then under *soft assignment*, which cluster will bear a higher responsibility for x_m? Use two sentences to explain.

   Solution. If x_m is assigned to C_1 under *hard assignment*, it means $(x_m - \mu_1)^2 < (x_m - \mu_2)^2$. Therefore, $rm2 < rm1$. So x_m should be assigned to C_1 with a higher responsibility.

2. (4 points) Given μ_1 and μ_2, when looking at a single data point x_m, does soft assignment have a lower loss value than the hard assignment? Prove it. Hint: your loss function will be the sum of the squared distance from x_m to each center multiplied by the "responsibility" $r_{mk}, k = 1, 2$.

   Solution. No.

   $$L_s = r_{m1}(x_m - \mu_1)^2 + r_{m2}(x_m - \mu_2)^2.$$

   When $(x_m - \mu_1)^2 < (x_m - \mu_2)^2$,

   $$L_h = (x_m - \mu_1)^2.$$

   $$\begin{aligned} L_h - L_s &= (x_m - \mu_1)^2 - r_{m1}(x_m - \mu_1)^2 - r_{m2}(x_m - \mu_2)^2 && (1) \\ &= \frac{(x_m - \mu_1)^4 + (x_m - \mu_1)^2(x_m - \mu_2)^2 - (x_m - \mu_1)^4 - (x_m - \mu_2)^4}{(x_m - \mu_1)^2 + (x_m - \mu_2)^2} && (2) \\ &= \frac{((x_m - \mu_1)^2 - (x_m - \mu_2)^2)(x_m - \mu_2)^2}{(x_m - \mu_1)^2 + (x_m - \mu_2)^2} < 0 && (3) \end{aligned}$$

   Therefore, the soft assignment actually gives a higher loss value than hard assignment.

- **Section 5. Linear Regression** (13 points)

1. (4 points) In least squares linear regression, our empirical risk is

$$\boldsymbol{R} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \boldsymbol{\theta}^T\boldsymbol{x}_n)^2,$$

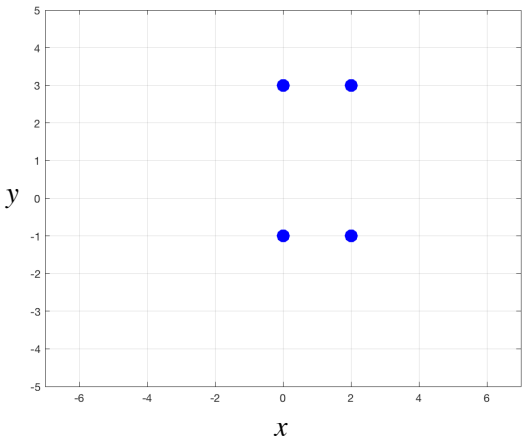
where $\boldsymbol{x}_n \in \mathbb{R}^d$ is a training sample, $y_n \in \mathbb{R}$ is its label. $\boldsymbol{\theta}$ contains the model parameters. Now we use a sigmoid function in this empirical risk, *i.e.,*

$$\boldsymbol{R} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \text{sigmoid}(\boldsymbol{\theta}^T\boldsymbol{x}_n))^2.$$

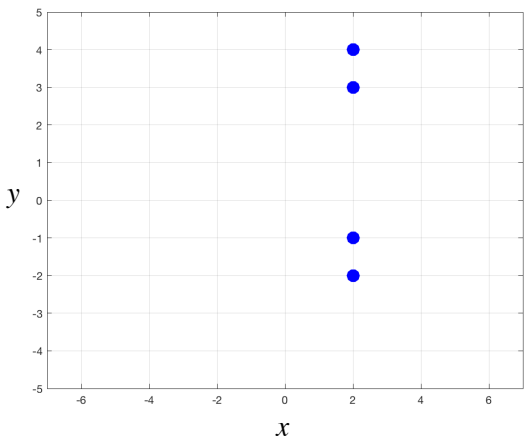
In no more than 4 sentences, state the reason why using sigmoid function in this way is not desirable.

Solution. The sigmoid function can only output values between 0 and 1, while the ground truth y could be other values. Moreover, the sigmoid function gives very similar values when $\boldsymbol{\theta}^T\boldsymbol{x}$ is very large (or very small) and thus will give similar loss values. This property will compromise model prediction performance under very large or very small $\boldsymbol{\theta}^T\boldsymbol{x}$.

2. Now we have four data points shown in the figure below. We use the least square error in regression.



(a)



(b)

1) (3 points) In (a), draw the linear regression output and the first principal component. Are they of the same direction/orientation? Use no more than 3 sentences to explain why.

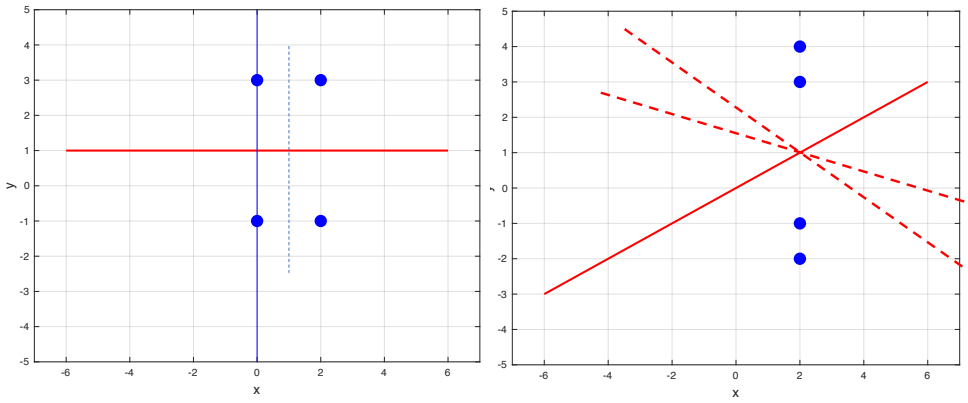
Solution. As shown in figure (a) below, the linear regression line should be $y = 1$. The first principal component is vertical. Students get full marks if the PCA result is vertical.

2) (3 points) In (b), draw the regression output, using language to describe when necessary.

Note: you can use any drawing software (PowerPoint, pdf editor, etc) to directly do it on figure; otherwise, approximately draw the four points on your paper and then draw the regression/PCA lines.

Solution. As shown in figure (b) below, the linear regression output should be any straight line passing the point $(2, 1)$ except the vertical one.

3. (3 points) In the lecture slides, to derive compact formula, we include a dummy feature $x_0 = 1$ into the sample vector $\boldsymbol{x}$, and correspondingly $\boldsymbol{\theta}$ includes the coefficient θ_0 of this dummy feature. Now I want to remove this dummy feature and its coefficient from the



regression model. Will the resulting model give better test accuracy after training (sufficient training samples, no over-fitting)? In 3 sentences justify your answer.

Solution. Removing this dummy dimension from the model will never give better performance. This is because the new model has to be going through the origin - its expressive power is usually lower. The resulting model will have a similar performance with the original model only if the underlying function itself goes through the origin, *i.e.*, θ_0 should be 0 anyway.

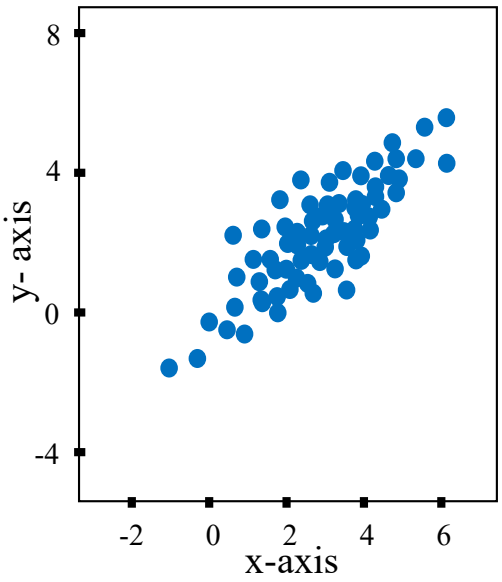
- **Section 6. Principal Component Analysis (PCA) and Linear Regression** (14 points)

1. (4 points) Give the main steps for principal component analysis. For example, given $\boldsymbol{X} \in \mathbb{R}^{D\times N}$, list the steps to reduce dimensions to $D-1$, and calculate a new $\boldsymbol{X} \in \mathbb{R}^{(D-1)\times N}$.

   Solution. 1. Standardise the data. Subtract the mean and divide by the standard deviation.
   2. Perform eigendecomposition of the covariance matrix $\frac{1}{N}XX^T$.
   3. Select the largest $D-1$ eigenvalues, and the corresponding $D-1$ eigenvectors constitute the projection matrix $\boldsymbol{B}$. The project is calculated as $\tilde{\boldsymbol{X}} = \boldsymbol{B}\boldsymbol{B}^T\boldsymbol{X}$.
   4. We undo the standardisation, yielding the dimension-reduced dataset in the original space.

2. (2 points) As shown in the figure below, we generate a 2D dataset with size 100×2. The eigenvalues and corresponding eigenvectors are given under the figure. Compute θ, which is defined as the angle ($\leq 90°$) between the first principal component and $x-$axis.

   Solution. The larger eigenvalue is 1683, so the major axis is $v2 = [-0.71, -0.71]^T$. $\tan(\theta) = \frac{-0.71}{-0.71} = 1. So \theta = 45°$.



$$\lambda_1 = 316, \lambda_2 = 1683$$

$$v_1 = [-0.71, 0.71]^T, v_2 = [-0.71, -0.71]^T$$

3. (2 points) Following the previous question, if we reduce the original data to one-dimensional data using PCA, calculate the percentage of information loss.

   Solution. ratio $= \frac{316}{316+1683} = 15.81\%$.

4. (3 points) Suppose we have sufficient training data. If we use PCA to first project the training data onto a few principal components, *i.e.,* performing dimension reduction, will this lower-dimensional training set give a better linear regression model than the original higher-dimensional training data? In three sentences, justify your answer.

   Solution. No, the new model trained with lower-dimensional data samples is no better than the old one. It is because through PCA there is inevitable some information loss (although not much). Since we have sufficient training data, the higher-dimensional data samples will not have over-fitting problem and, due to no information loss, allow better model training

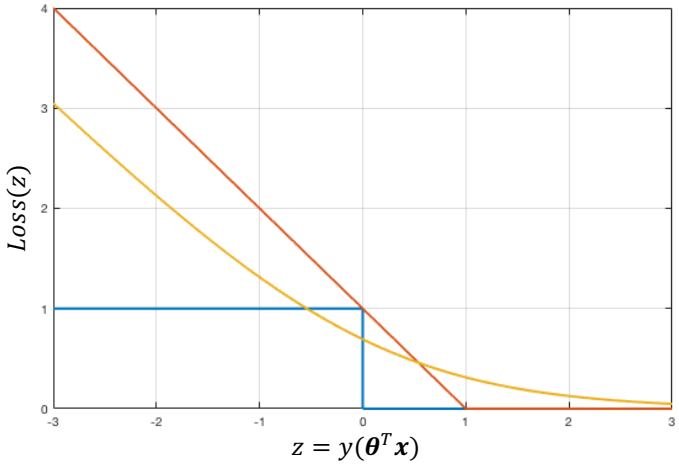
5. (3 points) Suppose you perform the following PCA operations on your d-dimensional data points of which the covariance matrix has d distinct eigenvalues. Operation 1: project the data onto a j-dimensional space, and then project the j-dimensional data onto a

g-dimensional space. Operation 2: project the d-dimensional data directly onto the g-dimensional space. Here $d > j > g$. Are the PCA results of the two operations the same? In no more than 5 sentences explain your answer.

Solution. The results are the same. Because the covariance matrix has distinct eigenvalues, its eigenvectors are orthogonal to each other. After projecting data onto j-dimensional space, eigenvectors associated with the largest $d-j$ eigenvalues remain, while the others are discarded. The discarded ones do not influence the remaining ones, as they are orthogonal. So no matter how many times you perform PCA, as long as the final dimension is the same, the finally selected eigenvectors are the same, so the PCA results are the same.

- **Section 7. Classification** (17 points)

  In your lecture slides, the zero-one loss, hinge loss and logistic loss (with basis e) can be plotted as the following figure. The loss functions are written as the function of $z = y(\boldsymbol{\theta}^T \boldsymbol{x})$. Here, $y \in \{-1, 1\}$ is the label of data sample $x \in \mathbb{R}^d$, where d is the feature dimension. $\boldsymbol{\theta} \in \mathbb{R}^d$ contains the model parameters.



  1. (1 point) What does a small z mean? What does a large z mean? Use one sentence to answer each.

     Solution. When z is large, the classifier makes a correct prediction; when z is small, the classifier makes an incorrect prediction.

  2. (3 points) If we use the least squares loss function to train a classifier, please write down the loss function 1) using $\boldsymbol{\theta}^T \boldsymbol{x}$ as argument, and 2) using z as argument.

     Solution. When using $\boldsymbol{\theta}^T \boldsymbol{x}$ as argument,

$$\text{Loss}_1 = \frac{1}{N} \sum_{n=1}^{N} (y - \boldsymbol{\theta}^T \boldsymbol{x})^2.$$

     Now we use $z = y\boldsymbol{\theta}^T \boldsymbol{x}$ as argument. When $y = 1$,

$$\text{Loss}_2 = \frac{1}{N} \sum_{n=1}^{N} (1 - y\boldsymbol{\theta}^T \boldsymbol{x})^2 = \frac{1}{N} \sum_{n=1}^{N} (1 - z)^2.$$

     When $y = -1$,

$$\text{Loss}_2 = \frac{1}{N} \sum_{n=1}^{N} (-1 + y\boldsymbol{\theta}^T \boldsymbol{x})^2 = \frac{1}{N} \sum_{n=1}^{N} (1 - y\boldsymbol{\theta}^T \boldsymbol{x})^2 = \frac{1}{N} \sum_{n=1}^{N} (1 - z)^2.$$

     Therefore, $\text{Loss}_2 = \frac{1}{N} \sum_{n=1}^{N} (1 - z)^2$

  3. (1 point) In the figure above, draw the least squares loss function in the same figure with the other three loss functions.

     Note: you can use any drawing software (PowerPoint, pdf editor, etc) to directly do it on figure; otherwise, roughly replicate this figure on paper and then add the curve for the least squares loss.

     Solution. This curve should be $L = (1 - z)^2$.

  4. (2 points) From what you have drawn, explain in 2 sentences why least squares is not a good choice for classifier training.

     Solution. Least squares loss gives positive loss values for correctly classified samples ($z > 1$); When z increases from 1 to inf, the loss increases even faster, meaning that the classifier will be updated a lot even under easy samples (correctly classified).

5. (2 points) Following the slides, we now transform z into probability:

$$p = p(y|\boldsymbol{x}) = \text{sigmoid}(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{(-y\boldsymbol{\theta}^T\boldsymbol{x})}}.$$

Now rewrite the logistic loss using p as argument. For your convenience, the logistic loss is $\text{Loss}_L(z) = \log(1+e^{-z})$ when using z as argument.

Solution. It is easy to know that

$$e^{-z} = p^{-1} - 1.$$

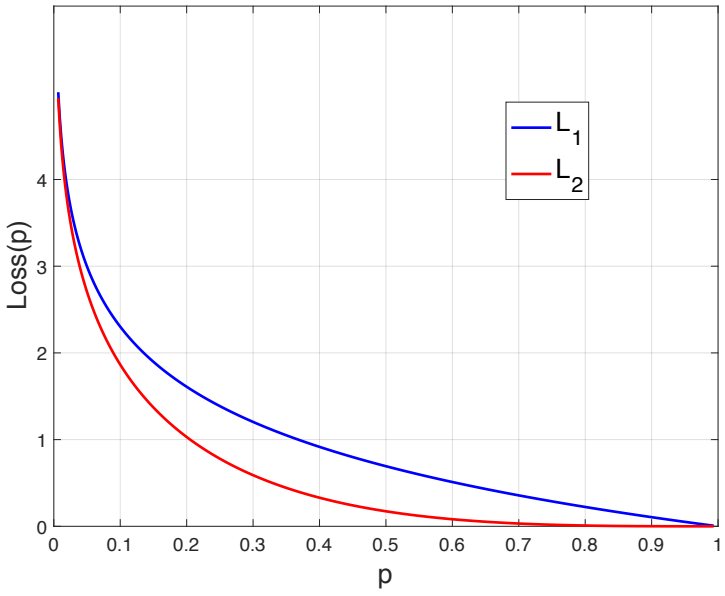
Therefore, $\text{Loss}_L(p) = \log(1+e^{-z}) = \log(1+p^{-1}-1) = \log(p^{-1}) = -\log(p)$.

6. (2 points) In the loss function $\text{Loss}_L(p)$ you just derived, what does a small p mean? what does a large p mean?

Solution. A small p ($p < 0.5$) means a data sample is incorrectly classified; A large p ($p > 0.5$) means a data sample is correctly classified.

7. (2 points) Suppose you are working on a two-way classification task. During classifier training, while some samples can be easily and successfully classified into the correct class, others are rather difficult and oftentimes misclassified. You want to solve this issue by designing a loss function that allows the classifier to focuse more on such difficult and easily misclassified samples. We have two loss functions for you to choose, all expressed as a function of p: $L_1(p)$ and $L_2(p)$. We draw these loss functions in the figure below against p. Which loss function would you like to choose? Use no more than 3 sentences to justify your answer.

Solution. I would like to use the red loss function L_2. In L_2, the loss for easy samples (those correctly classified) is very small when $p > 0.6$; the loss for hard samples (those incorrectly classified) remains large. It is clear from the figure that L_2 pays much less attention to the easy samples while focusing primarily on the hard samples.



8. You want a K-class classifier, where the number of classes $K > 2$. According to the lecture slides, this classifier contains K linear functions

$$g_k(\boldsymbol{x}) = \boldsymbol{\theta}_k^T\boldsymbol{x} + \theta_{k0}, k = 1, ..., K.$$

Here, $g_k(\boldsymbol{x}) \in \mathbb{R}$ characterises how likely sample $\boldsymbol{x}$ belongs to the kth class. Besides, vector $[\boldsymbol{\theta}_k^T, \theta_{k0}]^T$ is also called the prototype of the kth class. If a test sample is closest to the prototype of the ith class, it will be classified into the ith class. Suppose for some reason, that you find it very undesirable to train the classifier using methods like gradient descent.

In other words, you find using what we've learned in the classification lecture results in a poor classifier.

1) (2 points) In two sentences, give a potential reason why the classifier trained by gradient descent gives very poor performance (suppose your make no mistakes in programming and math). Trivial answers (*e.g.,* my computer is down) will receive 0.

Solution. There are too few training samples, which would lead to severe over-fitting if training is undertaken.

2) (2 points) Propose a way to calculate the prototype vectors that could give decent classification performance without training the classifier.

Solution. For each class, calculate the average data vector and use this vector (plus the dummy dimention 1) as the prototype.

——— End of the paper ———