Elements of Machine Learning, WS 2021/2022
Prof. Dr. Isabel Valera and Prof. Dr. Jilles Vreeken
EXAM, FEBRUARY 24TH, 2022, SOLUTION SHEET

PROBLEM 1 (LINEAR REGRESSION AND REGULARIZATION) **(10 points)**

1. Analyze, using linear regression, the data from the following table.

X_1	2	2.3	2.4	2.6	2.8	3
Y	14	14.6	14.8	15.2	15.6	16

Recall that linear regression takes the form $Y = \beta_0 + \beta_1 X_1$, but that it is often convenient to formulate it as $\mathbf{X}\beta = \mathbf{Y}$ with $\beta = [\beta_0\ \beta_1]^\top$ and $\mathbf{X} = [\mathbb{1}\ \mathbf{X_1}]$.

(a) Estimate the coefficients β_0 and β_1 using the following result.

$$(\mathbf{X}^\mathbf{T}\ \mathbf{X})^{-1} = \begin{bmatrix} 9.93 & -3.88 \\ -3.88 & 1.54 \end{bmatrix} .$$

Explain and *justify* each step you take. (2 points)

(b) Based on the value of β_1: What can you say of the relationship between X_1 and Y? (1 point)

(c) If we know that the data was generated as $Y = \beta_1 X_1 + \beta_0 + \epsilon$, with $\mathbb{E}[\epsilon] = 0$ and $Var(\epsilon) = 0.5$, can you tell, 95%-confidently, that the trend given in (b) holds? Why? (2 points)

2. Assume that we have a one-dimensional dataset for which we perform ridge regression, where we fix $\beta_0 = 0$. That is, we solve the following optimization problem.

$$\min_{\beta_1} \sum_{n=1}^{N} (y_n - x_n\beta_1)^2 + \lambda\beta_1^2 .$$

(a) Derive, step by step, a closed-form solution for β_1. Make sure that the obtained expression is a minimizer of the optimization problem. (3 points)

(b) Intuitively, what effect does λ have on the *bias* of the estimate of β_1? And on its *variance*? (2 points)

Solution.

1. We have that $X \in \mathbb{R}^{n\times 2}, \beta \in \mathbb{R}^{2\times 1}$. In matrix form: $\mathbf{Y} = \mathbf{X}\beta = \mathbb{1}\beta_0 + X\beta_1$ with $\beta = [\beta_0 \ \beta_1]^\top$ and $\mathbf{X} = [\mathbb{1} \ \mathbf{X_1}]$.

   (a) We can easily check that $\text{rank}(\mathbf{X}) = 2$, and thus it is left-invertible. That is, there exists a $\mathbf{X}^+ \in \mathbb{R}^{2\times n}$ such that $\mathbf{X}^+\mathbf{X} = \mathbf{I}_2$. This is the Moore-Penrose pseudo-inverse and can be written as $\mathbf{X}^+ = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$.
   Therefore, we can estimate β by simply solving the linear system in the matrix form as
   $$\mathbf{X}^T\mathbf{Y} = \mathbf{X}^+\mathbf{X}\beta$$
   using the given approximation for $(\mathbf{X}^\top\mathbf{X})^{-1}$. In particular, we have the following.
   $$\begin{aligned}\beta &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}\\ &= (\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top\mathbf{Y})\\ &= \begin{bmatrix} 9.93 & -3.88 \\ -3.88 & 1.54 \end{bmatrix}\begin{bmatrix} 90.2 \\ 228.3 \end{bmatrix} = \begin{bmatrix} 9.882 \\ 1.606 \end{bmatrix}\end{aligned}$$
   So we have that $\beta_0 = 9.882$ and $\beta_1 = 1.606$.

   (b) The value of β_1 is approximately 1.606, which means that there is a positive correlation between X_1 and Y. That is, the bigger X_1 is, the bigger Y becomes. Furthermore, $\beta_1 = 1.606$ suggests that with an increase of 1 unit in X_1, Y increases by 1.606 units on average.

   (c) In order to know the value of β_1 with a confidence of 95%, we need to compute the confidence interval of the parameter. This interval can be written as $[\beta_1 - 2\cdot SE(\beta_1), \beta_1 + 2\cdot SE(\beta_1)]$, where
   $$SE(\beta_1)^2 = \frac{Var(\epsilon)}{\sum_{n=1}^{N}(x_n - \overline{x})^2}$$
   as mentioned in slides. Since the noise has $Var(\epsilon) = 0.5$, we have that $SE(\beta_1) \approx 0.87$, and thus the interval is approximately $[-0.15, 3.36]$. Therefore, we cannot ensure with high confidence that the positive correlation between X_1 and Y holds, since the interval $[-0.15, 3.36]$ has non-positive values.
   In other words, if we draw 100 datasets from the model as in the problem, we are *not* guaranteed to acquire $\beta_1 > 0$ in at-least 95% of the datasets.

2. (a) We want to solve
   $$\min_{\beta_1} L = \min_{\beta_1} \sum_{n=1}^{N}(y_n - x_n\beta_1)^2 + \lambda\beta_1^2 \ .$$
   We start by differentiating w.r.t β_1,
   $$\partial_{\beta_1} L = \partial_{\beta_1}\left(\sum_{n=1}^{N}\left[y_n^2 + x_n^2\beta_1^2 - 2y_nx_n\beta_1\right] + \lambda\beta_1^2\right) =$$

$$= 2\sum_{n=1}^{N}\left(x_n^2\beta_1 - y_n x_n\right) + 2\lambda\beta_1.$$

Then we set the derivative to 0 and solve for β_1,

$$\frac{1}{2}\partial_{\beta_1}L = \sum_{n=1}^{N}\left(x_n^2\beta_1 - y_n x_n\right) + \lambda\beta_1 = 0$$
$$\left(\sum_{n=1}^{N} x_n^2 + \lambda\right)\beta_1 = \sum_{n=1}^{N} y_n x_n$$
$$\beta_1 = \frac{\sum_{n=1}^{N} y_n x_n}{\sum_{n=1}^{N} x_n^2 + \lambda} = \frac{\langle Y, X_1\rangle}{\langle X_1, X_1\rangle + \lambda}\ .$$

We need to make sure that this expression is a minimizer.

- **Option 1.** Take the second derivative and check that it is positive.

$$\partial^2_{\beta_1}L = 2\partial_{\beta_1}\left(\left[\sum_{n=1}^{N} x_n^2 + \lambda\right]\beta_1\right) = 2\sum_{n=1}^{N} x_n^2 + 2\lambda > 0$$

- **Option 2.** L is a quadratic function w.r.t β_1, and its first coefficient is $\sum_{n=1}^{N} x_n^2 + \lambda > 0$, thus it opens upwards (convex function).

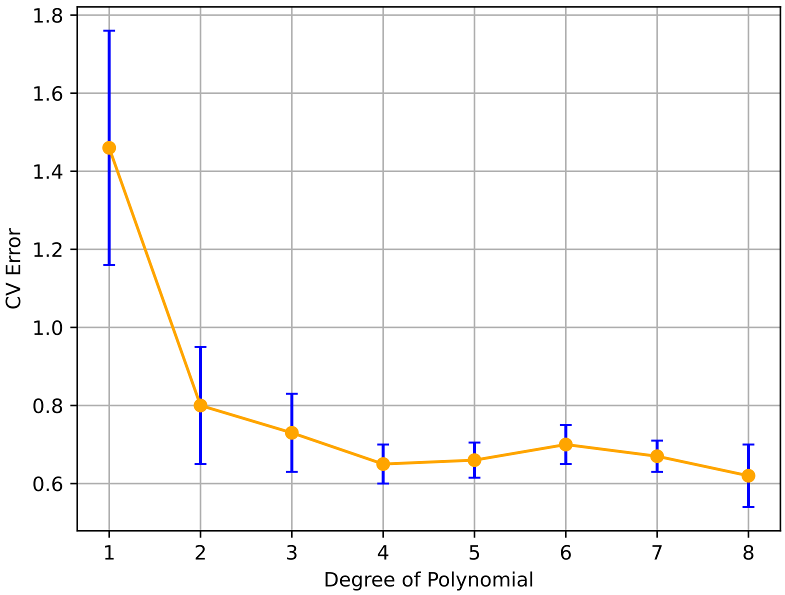
(b) Increasing the regularization strength λ increases the bias, and decreases the variance, of β_1, by preferring the estimated β_1 being closer to 0. As a result, data has less influence on the value of β_1. In other words, the higher the λ value, the less fluctuation of the estimated β_1 we observe between randomly generated datasets. On the other hand, when we decrease the regularization strength λ, we effectively decrease the bias but increase the variance of β_1.

Problem 2 (Non-Linear regression and errors) (10 points)

1. We have been requested to build a model that can properly model some complex data. To this end, we have decided to use polynomial regression.

   (a) Describe in your own words the main idea behind polynomial regression. What is its main advantage over linear regression? How can we estimate its parameters? (2 points)

   (b) Given the cross-validation error as a function of the degree of the polynomial in the figure below (mean in yellow, standard deviation in blue): Which polynomial degree would you use? Justify your answer. (1 point)



2. We want to use a tree-based approach to learn to predict a target (Y) from a predictor (X) such as the one shown in the image below. For this task, we consider a simple regression tree, where the tree is created following the greedy *recursive binary splitting* algorithm seen in the lectures. Recall that, at each step, the algorithm chooses the predictor X_j and cut point s, creating two new regions $R_1(j,s) = \{X|X_j < s\}$ and $R_2(j,s) = \{X|X_j \geq s\}$ solving

$$\min_{j,s} \sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2, \tag{2.1}$$

where $\hat{y}_{R_k}$ denotes the mean response for the training observations in $R_k(j,s)$.

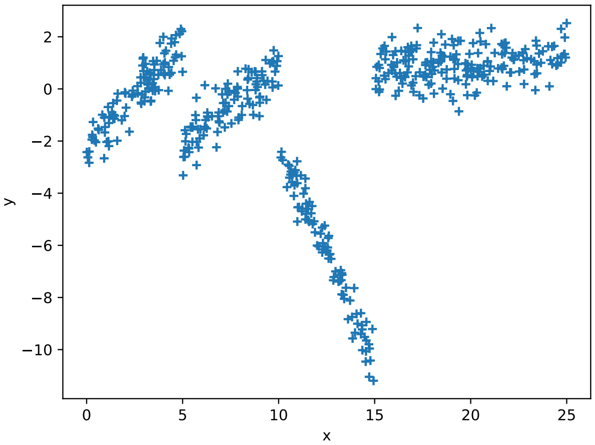
(a) Is such a tree a good model for the data shown below? Why (not)? (2 points)

(b) How would you reduce the training error to 0? And, in contrast, how would you avoid this from happening? Why would you be interested in avoiding zero training error? (2 points)

We decide to slightly change our approach and use, instead, a Linear Model Tree. This model differs from a standard regression tree in that $\hat{y}_{R_k}$ is replaced (in Eq. 2.1) by the prediction of a linear model fitted using the data from that region, that is, $\hat{y}_{R_k}$ is replaced by $\hat{y}_{i,R_k} = x_i a_{R_k} + b_{R_k}$.

(c) Is this a better model for our data? Why (not)? (1 point)

(d) Explain the steps to predict the response for a new data point. (2 points)



Solution.

1. (a)
   - The main idea behind polynomial regression is to include higher order powers X_i^n of the predictors X_i to predict a target Y. The main advantage of using polynomial regression over linear regression is that we can model non-linear relationships between the predictors X_i and the target Y.
   - We can estimate the parameters with the least squares, as in linear regression, by treating each higher order power X_i^n as a separate predictor. As a simple illustrative example, assuming we have a single predictor $X \in \mathbb{R}$ for which we wish to include a quadratic and a cubic polynomial to predict a target Y, we estimate the parameters $\beta = (\beta_0, \dots, \beta_3) \in \mathbb{R}^4$ as

   $$\min_{\beta} \mathbb{E}[(Y - \beta_0 - \beta_1 X - \beta_2 X^2 - \beta_3 X^3)^2].$$

   (b) Polynomial with degree 4. While 8 has a slightly lower CV mean error, in this case we prefer a simpler model as it has lower variance than degree 8 while having roughly the similar CV mean error (and thus insignificant difference in the bias between the two models). Following the one standard error rule, the mean error of 4 degree model is within the standard deviation of 8 degree model. Hence, we pick a polynomial of degree 4.

2. (a)
   - **Positive answer.** A regression tree will fit the data fairly well, the sudden jumps of Y correspond well to splits with our tree. However, to model the linear relationship within these regions a decision tree is suboptimal, but will still produce decent predictions.
   - **Negative answer.** A regression tree will not fit the data well. While it can split the 4-distinguished regions well, it would fail at regressing within each region. For each point in the region, it will always output the average response, which can be quite off.

   (b) A decision tree can easily achieve zero training loss by splitting until each data point has its own region. Achieving zero training error is a clear sign that we overfitted our model and hence badly generalize to the true underlying relation between input and target. To avoid overfitting, we can: i) prune a tree to a smaller subtree with weakest link pruning; ii) set a threshold on the minimum number of nodes per region (see ISLR).

   (c) This is a better model for the given data, as the target Y seems to follow a linear relationship with the predictor X (i.e. $Y = mX$ for some m not necessarily equating to 0) in each of the 4 groups. We can split the data in 4 regions and within each region model the target Y using X via the linear least squares, to exploit the linear relationship that the data exhibits.

   (d) For an unseen data-point we first apply the trained decision tree until we reach a leaf node. Recall that for each leaf node, we have a corresponding linear model for predictions. Now, for the leaf node we arrive at for the unseen data-point, we use the corresponding linear model to get our final prediction.

Problem 3 (Linear classification) **(10 points)**

Our highly competent research team is dealing with a classification problem in which they want to predict the type of monkey from an NFT, out of K different monkey classes. However, the GPUs are broken, and their non-deep-learning skills are a bit rusty. As an external advisor, they demand your expertise in linear classifiers to make a decision on which model to use.

Ian Badfellow seeks for perfection, and claims that they should use the *ideal* classifier.

(a) Explain him in your own words what is the Bayes Classifier, in which sense it is ideal, and why we do not use it in practice so often. (2 points)

Jürgen Schüber, who complains that Bayes did not properly cite the work of Leibniz, claims that we need some assumptions if we want to solve the problem. To this end, he kindly reminds you that according to the Bayes's theorem for each class k we have

$$\Pr(Y = k \mid X = x) = \frac{\pi_k \cdot f_k(x)}{\sum_{l=1}^{K} \pi_l \cdot f_l(x)} ,$$

where $f_k(x)$ denotes the density function of X for the k-th class, $\Pr(X = x \mid Y = k)$.

(b) What is π_k in the above expression? How would you estimate π_k for a given dataset? (1 point)

(c) Assume that f_k is provided. How can you use the Bayes' theorem to predict the class label of a new data point? (1 point)

Unfortunately, f_k is unknown in practice, and our experts really need your help to decide.

(d) Give the specific form of f_k, and list the extra assumptions made for: (2 points)

i) Multinomial Logistic Regression.
ii) Linear Discriminant Analysis.
iii) Quadratic Discriminant Analysis.

(e) Assume $K = 2$. What is the odds ratio? And the discriminant function? How do they relate? (2 points)

José Bengio is tired of talking and wants to step in. To keep funding coming, the team is interested in knowing whether the monkey is valuable ($Y = k$) or not ($Y \neq k$) (i.e., binary classification).

(f) In order to assess the best method, José wants to compute the ROC curve for all the previous methods. What are the axes of the plot, and how do you generate the curve (that is, which value would you change to generate the curve)? (2 points)

Solution.

(a) Given a specific data-point x, the Bayes classifier outputs the class $\arg\max_k P(Y = k \mid X = x)$. The classifier is ideal in the sense that it predicts the most probable class for the data-point x, and thus minimizes the misclassification probability. In practice, however, we cannot use the Bayes classifier, as we do not know either the desired conditional distribution $P(Y \mid X)$, or the distributions $P(X \mid Y)$ and $P(Y)$ to compute the desired conditional distribution $P(Y \mid X)$.

(b) π_k is the prior probability $P(Y = k)$ for observing an instance of class k. We can approximate $P(Y = k)$ counting the instances of class k in our data and dividing it by the total number of instances in our data.

(c) Given that $P(Y = k \mid X = x) \propto \pi_k \cdot f_k(x)$. We plug in our calculated π_k and the given $f_k(x)$ in to the equation of the Bayes Theorem, to compute the score for the class k. Finally, by taking $\arg\max_k \pi_k \cdot f_k(x)$, we approximate the Bayes optimal decision.

(d) Summary:

Method	f_k	Assumptions
Logistic	$\exp(\beta_0 + \sum_p \beta_p x_p)$	Logits are linear in X.
LDA	$\mathcal{N}(\mu_k, \Sigma)$	Densities are Gaussian with *shared* variances.
QDA	$\mathcal{N}(\mu_k, \Sigma_k)$	Densities are Gaussian with *exclusive* variances.

(e) (Solution for general K) The odds ratio between class k and k' is $\frac{\Pr(Y=k|X=x)}{\Pr(Y=k'|X=x)}$. The discriminant function for the class k is $\delta_k = \log \Pr(Y = k|X = x)$.

Therefore, the odds ratio can be written as $e^{\delta_k}/e^{\delta_{k'}}$.

(f) To generate the plot, we draw in the y-axis the true positive rate (TPR) (aka. sensitivity), and the false positive rate (FPR) (aka. one minus the specificity) in the x-axis.

We generate the ROC curve by modifying the threshold α for which we choose the positive class, that is, the rule for which we say that x is of the positive class if $\Pr(Y = +|X = x) \geq \alpha$.

PROBLEM 4 (NON-LINEAR CLASSIFICATION) **(10 points)**

1. The greedy algorithm seen in the course to build classification trees does not allow for partitions such as the one in Fig 2.

   (a) Draw the partition produced by the decision tree shown in Fig 1. Is the partition unique? Why? (2 points)

   (b) Why cannot we build a tree that produces the partition in Fig. 2? How would you change the model to allow such a partitioning? You do not have to explain how the changed model is fitted. (2 points)

   (c) Is the misclassification error a good loss function to generate a tree? Why? Justify your answer with an example. (2 points)

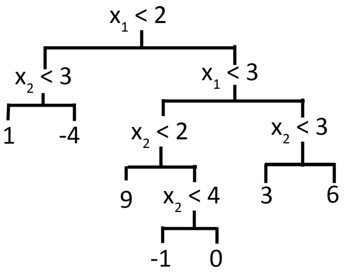


Fig 1. Decision tree for exercise (a).

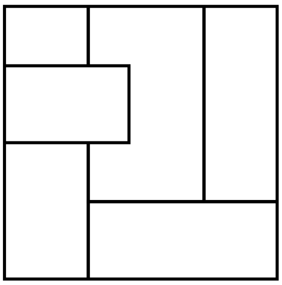


Fig 2. Partition for exercise (b).

2. Recall that the Support Vector Machine is defined as follows:

$$\begin{aligned} &\underset{\substack{M \\ \beta_0,\dots,\beta_p \\ \xi_1,\dots,\xi_N}}{\text{maximize}}\ M \\ &\text{subject to } \|\beta\| = 1 \\ &\qquad y_i f(x_i) \geq M(1-\xi_i) \quad \text{for } i = 1,\dots,N \\ &\qquad \xi_i \geq 0, \sum_{i=1}^{N} \xi_i \leq C \end{aligned}$$

   (d) How does an SVM differ from a maximal margin classifier? (1 point)

   (e) Explain the purpose of the variable C in the optimization problem above. (1 point)

   (f) How does the kernel trick help an SVM classify non-linearly related data? What is the main advantage of this approach? (2 points)

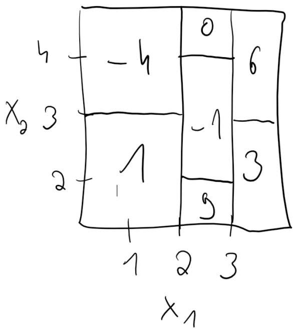
Solution.

1. (a) Yes, partitions are always unique. Proof by contradiction: assume there are two different partitions that both represent the same tree. Because they are different, we have two cases:

    i. There is a coordinate tuple where the two partitions return different values. In this case, the same tree would give two outputs for the same input, which is impossible because decision trees are deterministic.

    ii. Both partitions return the same value for every input, but one has more dividers than the other. Since we have as many dividers in the partition as inner nodes in the tree, this cannot be the case either.

    Thus, the two partitions cannot be different if they represent the same decision tree.



   (b) Because the conditions on the internal nodes are too simple. In particular, each internal node considers a single predictor. On the other hand, to create the partition as in Fig. 2, we would need to combine conditions on multiple predictors, e.g. by an AND operator.

   (c) No, misclassification is not a good loss function to generate a tree, as it is relatively more insensitive to node inbalances, compared to other metrics e.g. the Gini Index. An illustrative example is provided in the book "The Elements of Statistical Learning", section 9.2.3.

2. (d) The SVM allows for some points to lay in the margin or even on the wrong side of the hyperplane, whereas the maximum margin classifier does not allow such a flexibility. In other words, in the maximum margin classifier, we force all the slack variables $\xi_i = 0$; as a result, the constraint $y_i f(x_i) \geq M(1 - \xi_i)$ in the SVM boils down to $y_i f(x_i) \geq M$.

   (e) The variable C controls the misclassified points. If $C = 0$, then we ensure all the slack variables $\xi_i = 0$, resulting in a maximum margin classifier as mentioned above. However, when the data is not linearly separable, we allow for some errors (i.e. points on the incorrect side of the hyperplane), but we penalize these errors. Here, the C guarantees that we have at most C misclassified points.

   (f) Via the kernel trick, we transform the data through a nonlinear transformation to an alternate feature space, such that the SVM can be applied to find a linear

decision boundary in the transformed feature space. As a result, the SVM then effectively capture non-linear relationships in the input space.
The main advantage of using the kernel trick is that it requires only dot products of representations in the feature space, without explicitly requiring the representations in the transformed feature space. And computing the former can be significantly faster than computing the latter if the transformed feature space is very high dimensional.

PROBLEM 5 (UNSUPERVISED LEARNING) **(10 points)**

1. Given the following set of points:

$$\mathbf{x}_1 = (7,\ \ 0);\quad \mathbf{x}_2 = (5,-3);\quad \mathbf{x}_3 = (1,\ \ 6);$$
$$\mathbf{x}_4 = (6,-1);\quad \mathbf{x}_5 = (5,\ \ 3);\quad \mathbf{x}_6 = (2,-3);$$

   Compute two full iterations of k-means clustering (Lloyd's algorithm) with initial clusters $\mu_1 = (-1, 2)$ and $\mu_2 = (3, 5)$. Use $d(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$ as distance. Make sure to write down the necessary distances, explain the steps you follow, and to describe the resulting clusters (centroid and points) *at the end of both iterations.* (3 points)

2. Draw the dendrogram for the following dataset, using single linkage hierarchical clustering with the Manhattan distance, $d(\boldsymbol{a}, \boldsymbol{b}) = |a_1 - b_1| + |a_2 - b_2|$. Make sure to indicate the distances, the height at which each fusion occurs, as well as the observations corresponding to each leaf in the dendrogram. (3 points)

Name	X_1	X_2
A	5	2
B	3.5	1
C	-3	2
D	2	4
E	7	-3
F	3	3.5

3. Suppose we have a dataset which has too many features, and thus we wish to perform dimensionality reduction on the dataset by applying PCA:

   (a) Does PCA perform feature selection? Why (not)? (2 points)

   (b) Say we use PCA to reduce the data dimensionality to a single feature. Give two different ways we can interpret the resulting feature. (2 points)

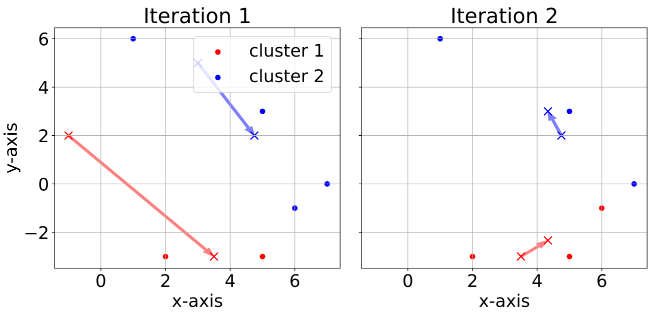


Figure 1: Illustration of K-Means algorithm. Crosses represent centroids of the clusters of respective color. Arrows denote the update of cluster centroid after the iterations.

Solution.

1. **Iteration 1**. We first compute the distances between the datapoints $\mathbf{x}_i$ and the cluster centroids μ_1 and μ_2, which is as follows.

$d(\mathbf{a}, \mathbf{b})$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$
μ_1	68	61	20	58	37	34
μ_2	41	68	5	45	8	65

We now select, for each data point $\mathbf{x}_i$, a cluster j such that the euclidean distance between the datapoint $\mathbf{x}_i$ and the cluster centroid μ_j is minimized, giving us the following cluster assignments.

Points	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$
Cluster Assignments	2	1	2	2	2	1

Finally, we update the means: $\mu_1 = (3.5, -3)$, $\mu_2 = (4.75, 2)$. Figure 1 (left) illustrates the steps.

Iteration 2. We repeat the process as before, although now with updated means μ_1 and μ_2 from Iteration 1. Namely, we first compute the distances as

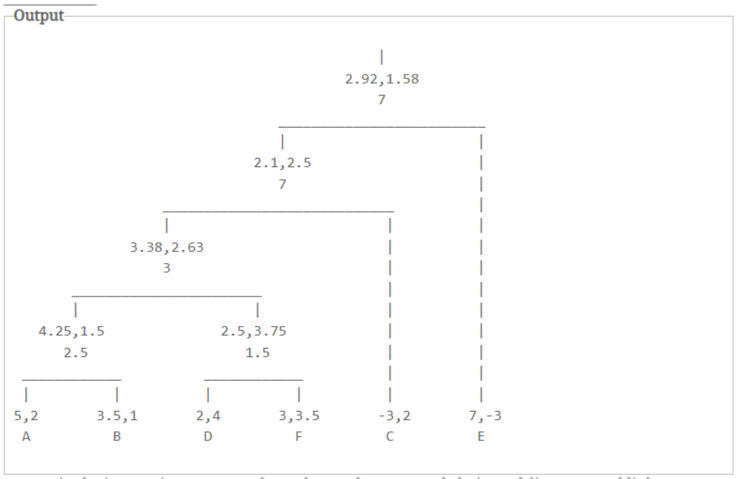
$d(\mathbf{a}, \mathbf{b})$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$
μ_1	21.2	2.6	87.2	10.2	28.2	2.2
μ_2	9.1	25.1	30.1	10.6	1.1	32.6

This gives us the updated cluster assignments as

Points	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$
Cluster Assignments	2	1	2	1	2	1

Finally, we have the updated means as $\mu_1 = (4.3, -2.3)$, $\mu_2 = (4.3, 3)$. Figure 1 illustrates the 2nd iteration.

2. Dendrogram:



```
                                             |
                                         2.92,1.58
                                             7
                                __________________________
                                 |                       |
                             2.1,2.5                     |
                                 7                       |
                  _____________________________          |
                   |                          |          |
              3.38,2.63                       |          |
                   3                          |          |
       ________________________               |          |
       |                      |               |          |
   4.25,1.5              2.5,3.75             |          |
      2.5                   1.5               |          |
 _____________         _____________          |          |
 |          |           |          |          |          |
5,2       3.5,1       2,4       3,3.5        -3,2       7,-3
 A          B           D          F          C          E
```

3. (a) No! We do not select features to keep and delete others. In PCA, we get the principal components, which are in a sense new features computed from the old ones. Feature selection would mean selecting some of the original features and discarding others.

   (b) The resulting feature is a linear combination of the already existing features, such that the variance of this resulting feature is maximized. Alternatively, we can also interpret the resulting feature such that the (sum of) euclidean distance between the original data-points and the respective transformed data-points in this feature space is minimized.