# Q1-1: Select the correct statement.

A. *Support vector machines are able to produce non-linear decision boundaries by, in a sense, transforming low-dimensional inputs into a high-dimensional space, then performing classification in that high-dimensional space. This usually works because high-dimensional data is much more likely to be linearly separable than low-dimensional data.*

B. *"Kernel trick" refers to first applying the above transformation and then computing the dot products between the transformed data points.*

1. Both the statements are TRUE.
2. Statement A is TRUE, but statement B is FALSE.
3. Statement A is FALSE, but statement B is TRUE.
4. Both the statements are FALSE.

# Q1-1: Select the correct statement.

A. *Support vector machines are able to produce non-linear decision boundaries by, in a sense, transforming low-dimensional inputs into a high-dimensional space, then performing classification in that high-dimensional space. This usually works because high-dimensional data is much more likely to be linearly separable than low-dimensional data.*

B. *"Kernel trick" refers to first applying the above transformation and then computing the dot products between the transformed data points.*

1. Both the statements are TRUE.
2. Statement A is TRUE, but statement B is FALSE. ←
3. Statement A is FALSE, but statement B is TRUE.
4. Both the statements are FALSE.

Since the training and classification only require the dot products between data points, it "cheats" by using the Kernel function instead. This cheating is referred to as the "kernel trick".

Q1-2: Consider the polynomial kernel $k(x', x') = (xx' + 1)^3$, for $x \in \mathbf{R}$ (i.e., a one-dimensional feature space). Give an explicit expression for the corresponding feature map $\phi(x)$ such that $k(x, x') = \phi(x)^T \phi(x')$.

1. $\phi(x)^T = [x^3, \sqrt{3}\, x^2, \sqrt{3}\, x, 1]$
2. $\phi(x)^T = [x^3, \sqrt[3]{3}\, x^2, \sqrt[3]{3}\, x, 1]$
3. $\phi(x)^T = [x^3, x^2, x, 1]$
4. $\phi(x)^T = [x^3, \sqrt{3}\, x^2, \sqrt{3}\, x]$

Q1-2: Consider the polynomial kernel $k(x', x') = (xx' + 1)^3$, for $x \in \mathbf{R}$ (i.e., a one-dimensional feature space). Give an explicit expression for the corresponding feature map $\phi(x)$ such that $k(x, x') = \phi(x)^T \phi(x')$.

1. $\phi(x)^T = [x^3, \sqrt{3}\, x^2, \sqrt{3}\, x, 1]$ ←
2. $\phi(x)^T = [x^3, \sqrt[3]{3}\, x^2, \sqrt[3]{3}\, x, 1]$
3. $\phi(x)^T = [x^3, x^2, x, 1]$
4. $\phi(x)^T = [x^3, \sqrt{3}\, x^2, \sqrt{3}\, x]$

$$\begin{aligned} k(x, x') &= (xx' + 1)^3 \\ &= (xx')^3 + 3(xx')^2 + 3xx' + 1 \\ &= \begin{bmatrix} x^3 & \sqrt{3}x^2 & \sqrt{3}x & 1 \end{bmatrix} \begin{bmatrix} (x')^3 \\ \sqrt{3}(x')^2 \\ \sqrt{3}x' \\ 1 \end{bmatrix} \end{aligned}$$

Q2-1: We know that $k(x, x') = \phi(x)^T\phi(x')$. For $k(x, x') = (x^Tx' + c)^3$; $x,x' \in \mathbf{R}^2$, how many terms are there in $\phi(x)$?

1. 6
2. 9
3. 10
4. 12

Q2-1: We know that $k(x, x') = \phi(x)^T\phi(x')$. For $k(x, x') = (x^Tx' + c)^3$; $x,x \in \mathbf{R}^2$, how many terms are there in $\phi(x)$?

1. 6
2. 9
3. 10 ⬅
4. 12

#terms in $\phi(x)$ = # terms in $(x_1x_1' + x_2x_2' + c)^3$. Denote $a = x_1x_1'$ , $b = x_2x_2'$ for simplicity.

#terms in $(a + b + c)^3$:
#Single element cubed: $a^3$, $b^3$, $c^3$ = 3

+

#involving 2 elements - one term squared and one single: $a^2b$, $ab^2$, $b^2c$, $bc^2$, $c^2a$, $ac^2$ = 6 +
#involing all 3 elements: abc = 1
= 3 + 6 + 1 = 10

# Q2-2: You are training an RBF SVM with $\gamma$ = 1/2$\sigma^2$ (where $\sigma^2$ is the variance of the RBF kernel). Which of the following is correct?

1. To avoid overfitting, $\gamma$ should be reduced.
2. To avoid overfitting, $\gamma$ should be increased.
3. $\gamma$ has no predictable effect on overfitting
4. When I decrease $\gamma$, the number of support vector reduces. This indicates that the chosen value likely correspond to overfitting.

Q2-2: You are training an RBF SVM with $\gamma = 1/2\sigma^2$ (where $\sigma^2$ is the variance of the RBF kernel). Which of the following is correct?

1. To avoid overfitting, $\gamma$ should be reduced.
2. To avoid overfitting, $\gamma$ should be increased.
3. $\gamma$ has no predictable effect on overfitting
4. When I decrease $\gamma$, the number of support vector reduces. This indicates that the chosen value likely correspond to overfitting.

One way to test for overfitting: When we decrease γ the training accuracy reduces, but validation accuracy increases.

# Q3-1: Which of the following might be valid reasons for preferring an SVM over a neural network?

A. *An SVM can effectively map the data to an infinite-dimensional space, a neural net cannot.*
B. *The transformed (basis function) representation constructed by an SVM is usually easier to visualize/interpret than for a neural net.*
C. *An SVM would not get stuck in local minima, unlike a neural net.*

1. A, B
2. B, C
3. A, C
4. A, B, C

# Q3-1: Which of the following might be valid reasons for preferring an SVM over a neural network?

A. *An SVM can effectively map the data to an infinite-dimensional space, a neural net cannot.*
B. *The transformed (basis function) representation constructed by an SVM is usually easier to visualize/interpret than for a neural net.*
C. *An SVM would not get stuck in local minima, unlike a neural net.*

1. A, B
2. B, C
3. A, C ←
4. A, B, C

A: True using RBF kernel
B: Not necessarily
C: True (convex optimization problem in SVM)

Q3-2: Let $K_1$ and $K_2$ be $R^n \times R^n$ kernels and $c \in R^+$ be a positive constant. $\phi_1 : R^n \rightarrow R^d$ and $\phi_2 : R^n \rightarrow R^d$ are feature mappings of $K_1$ and $K_2$ respectively. How to use $\phi_1$ and $\phi_2$ to obtain the feature mapping for the kernel:

$$K(x, z) = c(K_1(x, z)+K_2(x, z))$$

1. [c $\phi_1(x)$, c $\phi_2(x)$]
2. [sqrt(c) $\phi_1(x)$, sqrt(c) $\phi_2(x)$]
3. [sqrt(c $\phi_1(x)$), sqrt(c $\phi_2(x)$)]
4. [c sqrt($\phi_1(x)$), c sqrt($\phi_2(x)$)]

Q3-2: Let $K_1$ and $K_2$ be $R^n \times R^n$ kernels and $c \in R^+$ be a positive constant. $\phi_1 : R^n \rightarrow R^d$ and $\phi_2 : R^n \rightarrow R^d$ are feature mappings of $K_1$ and $K_2$ respectively. How to use $\phi_1$ and $\phi_2$ to obtain the feature mapping for the kernel:

$$K(x, z) = c(K_1(x, z)+K_2(x, z))$$

1. $[c\ \phi_1(x),\ c\ \phi_2(x)]$
2. $[\text{sqrt}(c)\ \phi_1(x),\ \text{sqrt}(c)\ \phi_2(x)]$
3. $[\text{sqrt}(c\ \phi_1(x)),\ \text{sqrt}(c\ \phi_2(x))]$
4. $[c\ \text{sqrt}(\phi_1(x)),\ c\ \text{sqrt}(\phi_2(x))]$

Let $K'(x, z) = K_1(x, z)+K_2(x, z)$

Using kernel algebra, feature mapping $\phi'(x)$ corresponding to $K'(x, z)$: $\phi'(x) = [\phi_1(x)\ \ \phi_2(x)]$.

Now, since $K(x, z) = c\ K'(x, z)$, feature mapping $\phi(x)$ corresponding to $K(x, z)$: $\phi(x) = \text{sqrt}[c]\ \phi'(x) = [\text{sqrt}(c)\ \phi_1(x),\ \text{sqrt}(c)\ \phi_2(x)]$.