DEPARTMENT / INSTITUTE / SCHOOL / CENTRE OF EIE

SOLUTION & MARKING SCHEME (2017/ 2018)

COURSE: EIE6207 YEAR: 6
SUBJECT: Theoretical Fundamental and Engineering Approaches for Intelligent Signal and Information Processing

	SUBJECT EXAMINER	INTERNAL MODERATOR / ASSESSOR	EXTERNAL EXAMINER	
	M.W. Mak			

Q1 (a) The margin of separation is the projection of $(\mathbf{x}_1 - \mathbf{x}_2)$ onto the direction orthogonal (perpendicular for 2-D cases) to the decision boundary. Therefore, we have

$$\begin{aligned} d &= (\mathbf{x}_1 - \mathbf{x}_2) \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} \\ &= \frac{\mathbf{w} \cdot \mathbf{x}_1 - \mathbf{w} \cdot \mathbf{x}_2}{\|\mathbf{w}\|} \\ &= \frac{(1-b) - (-1-b)}{\|\mathbf{w}\|} \\ &= \frac{2}{\|\mathbf{w}\|}. \end{aligned}$$

(5 marks, K)

(b) (i) Based on Q1(a), maximizing d is equivalent to minimizing $\|\mathbf{w}\|^2$. The two given inequality constraints can be combined into one because

- when $y_i = 1$, $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 \Rightarrow \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$, which satisfies the first inequality constraint, and
- when $y_i = -1$, $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 \Rightarrow -(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 \Rightarrow \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$, which satisfies the second constraint.

(10 marks, KA)

(ii) The constraints are

- $\alpha_i \geq 0$,
- $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0$, and
- $\alpha_i[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1] = 0$

where $i = 1, \ldots, N$. (10 marks, AE)

(iii) Setting

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \{\alpha_i\}) = 0 \quad \text{and} \quad \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \{\alpha_i\}) = 0,$$

subject to the constraint $\alpha_i \geq 0$, we have

$$\sum\nolimits_{i=1}^{N} \alpha_i y_i = 0 \quad \text{and} \quad \mathbf{w} = \sum\nolimits_{i=1}^{N} \alpha_i y_i \mathbf{x}_i.$$

	SUBJECT EXAMINER	INTERNAL MODERATOR / ASSESSOR	EXTERNAL EXAMINER	
	M.W. Mak			

Substituting these results back into the Lagrangian function, we have

$$\begin{aligned} L(\mathbf{w}, b, \{\alpha_i\}) &= \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) - \sum_{i=1}^{N} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w}) - \sum_{i=1}^{N} \alpha_i y_i b + \sum_{i=1}^{N} \alpha_i \\ &= \frac{1}{2} \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \cdot \sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \cdot \sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j + \sum_{i=1}^{N} \alpha_i \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j). \end{aligned}$$

This results in the Wolfe dual formulation.

(15 marks, AE)

COURSE: EIE6207 YEAR: 6
SUBJECT: Theoretical Fundamental and Engineering Approaches for Intelligent Signal and Information Processing

	SUBJECT EXAMINER	INTERNAL MODERATOR / ASSESSOR	EXTERNAL EXAMINER	
	M.W. Mak			

Q2 (a) To find $\mathcal{X}_k$, we compute the derivative of E with respect to $\boldsymbol{\mu}_k$ and set the result to 0. Specifically,

$$\begin{aligned}\frac{\partial E}{\partial \boldsymbol{\mu}_k} &= \sum_{k=1}^{K}\sum_{\mathbf{x}\in\mathcal{X}_k}\frac{\partial}{\partial \boldsymbol{\mu}_k}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathsf{T}}(\mathbf{x}-\boldsymbol{\mu}_k)\\ &= \sum_{k=1}^{K}\sum_{\mathbf{x}\in\mathcal{X}_k}\frac{\partial}{\partial \boldsymbol{\mu}_k}(\mathbf{x}^{\mathsf{T}}\mathbf{x}-2\boldsymbol{\mu}_k^{\mathsf{T}}\mathbf{x}+\boldsymbol{\mu}_k^{\mathsf{T}}\boldsymbol{\mu}_k)\\ &= \sum_{\mathbf{x}\in\mathcal{X}_k}(-\mathbf{x}+\boldsymbol{\mu}_k)\\ &= 0.\end{aligned}$$

Therefore, we have

$$N_k\boldsymbol{\mu}_k = \sum_{\mathbf{x}\in\mathcal{X}_k}\mathbf{x} \Rightarrow \boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{\mathbf{x}\in\mathcal{X}_k}\mathbf{x}.$$

(6 marks, K)

(b) The K-means algorithm can only use hyperplanes to partition the data in the input space. For doughnut-shape clusters, any hyperplanes will cut through the data such that there will be many samples very close to or even overlap with the boundaries. This is undesirable because we want the clusters to be clearly separable from each other.

(7 marks, A)

(c)

$$\begin{aligned}E_{\boldsymbol{\phi}} &= \sum_{k=1}^{K}\sum_{\mathbf{x}\in\mathcal{X}_k}\left\|\boldsymbol{\phi}(\mathbf{x})-\frac{1}{N_k}\sum_{\mathbf{z}\in\mathcal{X}_k}\boldsymbol{\phi}(\mathbf{z})\right\|^2\\ &= \sum_{k=1}^{K}\sum_{\mathbf{x}\in\mathcal{X}_k}\left(\boldsymbol{\phi}(\mathbf{x})-\frac{1}{N_k}\sum_{\mathbf{z}\in\mathcal{X}_k}\boldsymbol{\phi}(\mathbf{z})\right)^{\mathsf{T}}\left(\boldsymbol{\phi}(\mathbf{x})-\frac{1}{N_k}\sum_{\mathbf{z}\in\mathcal{X}_k}\boldsymbol{\phi}(\mathbf{z})\right)\\ &= \sum_{k=1}^{K}\sum_{\mathbf{x}\in\mathcal{X}_k}\left[\boldsymbol{\phi}(\mathbf{x})^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{x})+\frac{1}{N_k^2}\sum_{\mathbf{z}\in\mathcal{X}_k}\sum_{\mathbf{z}'\in\mathcal{X}_k}\boldsymbol{\phi}(\mathbf{z})^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{z}')-\frac{2}{N_k}\sum_{\mathbf{z}\in\mathcal{X}_k}\boldsymbol{\phi}(\mathbf{z})^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{x})\right]\end{aligned}$$

Because the first term is independent of how the dataset is partitioned, the new

objective function is

$$E'_{\boldsymbol{\phi}} = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{X}_k} \left[\frac{1}{N_k^2} \sum_{\mathbf{z} \in \mathcal{X}_k} \sum_{\mathbf{z}' \in \mathcal{X}_k} \boldsymbol{\phi}(\mathbf{z})^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{z}') - \frac{2}{N_k} \sum_{\mathbf{z} \in \mathcal{X}_k} \boldsymbol{\phi}(\mathbf{z})^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{x}) \right].$$

(10 marks, A)

(d) We may use a kernel function to replace the dot products as follows

$$E'_{\boldsymbol{\phi}} = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{X}_k} \left[\frac{1}{N_k^2} \sum_{\mathbf{z} \in \mathcal{X}_k} \sum_{\mathbf{z}' \in \mathcal{X}_k} K(\mathbf{z}, \mathbf{z}') - \frac{2}{N_k} \sum_{\mathbf{z} \in \mathcal{X}_k} K(\mathbf{z}, \mathbf{x}) \right],$$

where $K(\mathbf{x}, \mathbf{y})$ is a nonlinear kernel such as polynomial or RBF kernels.

Optional: Note that unlike K-means, the kernel K-means cannot compute the means in the feature space because $\frac{1}{N_k} \sum_{\mathbf{x} \in \mathcal{X}_k} \boldsymbol{\phi}(\mathbf{x})$ is either un-implementable or too expansive to evaluate.

(7 marks, E)

COURSE: EIE6207 YEAR: 6
SUBJECT: Theoretical Fundamental and Engineering Approaches for Intelligent Signal and Information Processing

	SUBJECT EXAMINER	INTERNAL MODERATOR / ASSESSOR	EXTERNAL EXAMINER	
	M.W. Mak			

Q3 (a) The diagonal elements are the eigenvalues of the covariance matrix, which represent the variances of the projected components.

(5 marks, K)

(b) Instead of finding the eigenvectors of $\mathbf{X}\mathbf{X}^\mathsf{T}$, we solve the eigen-problem:

$$\mathbf{X}^\mathsf{T}\mathbf{X}\boldsymbol{\psi}_i = \lambda_i \boldsymbol{\psi}_i.$$

Then, we pre-multiply both sides this equation by $\mathbf{X}$ to obtain

$$\mathbf{X}\mathbf{X}^\mathsf{T}(\mathbf{X}\boldsymbol{\psi}_i) = \lambda_i(\mathbf{X}\boldsymbol{\psi}_i).$$

This means that if $\boldsymbol{\psi}_i$ is an eigenvector of $\mathbf{X}^\mathsf{T}\mathbf{X}$, then $\boldsymbol{\phi}_i = \mathbf{X}\boldsymbol{\psi}_i$ is an eigenvector of $\mathbf{X}\mathbf{X}^\mathsf{T}$. So, all we need is to compute the $N-1$ eigenvectors of $\mathbf{X}^\mathsf{T}\mathbf{X}$, which has size $N \times N$.

(10 marks, AE)

COURSE: EIE6207 YEAR: 6
SUBJECT: Theoretical Fundamental and Engineering Approaches for Intelligent Signal and Information Processing

	SUBJECT EXAMINER	INTERNAL MODERATOR / ASSESSOR	EXTERNAL EXAMINER	
	M.W. Mak			

Q4 (a) The softmax function should be used. Denote $a_k^{(L)}$ as the linear activation of k-th output node, the output at node k is given by

$$y_k = \frac{\exp(a_k^{(L)})}{\sum_{j=1}^{K} \exp(a_j^{(L)})}$$

(5 marks, KA)

(b) Denote $\mathbf{y}$ as the output vector, $\mathbf{W}^{(l)}$ as the weight matrix (including the bias terms) at layer l, and $\mathbf{a}^{(l)}$ as the linear weight sums at layer l. Also, denote $f(\mathbf{a})$ as the activation function that can be applied element-wise to the vector $\mathbf{a}$. Then, we may express $\mathbf{y}$ in terms of the input vector $\mathbf{x}$ as follows:

$$\begin{aligned}
\mathbf{y} &= f\left(\mathbf{W}^{(L)}\mathbf{a}^{(L)}\right) \\
&= f\left(\mathbf{W}^{(L)} f(\mathbf{W}^{(L-1)}\mathbf{a}^{(L-1)})\right) \\
&= f\left(\mathbf{W}^{(L)} f(\mathbf{W}^{(L-1)} f(\cdots f(\mathbf{W}^{(1)}\mathbf{x})))\right).
\end{aligned}$$

If f is linear, then we have $f(\mathbf{a}) = \mathbf{a}$. The above equation becomes

$$\begin{aligned}
\mathbf{y} &= \mathbf{W}^{(L)}\mathbf{a}^{(L)} \\
&= \mathbf{W}^{(L)}\mathbf{W}^{(L-1)}\mathbf{a}^{(L-1)} \\
&= \mathbf{W}^{(L)}\mathbf{W}^{(L-1)}\cdots\mathbf{W}^{(1)}\mathbf{x} \\
&= \mathbf{W}\mathbf{x},
\end{aligned}$$

where $\mathbf{W} = \mathbf{W}^{(L)}\cdots\mathbf{W}^{(1)}$. As a result, the output is linearly related to the input. So, the network is linear and is not very useful.

(10 marks, AE)