EE4-62
EE9SO25

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2012

MSc and EEE/ISE PART IV: MEng and ACGI

# MACHINE LEARNING FOR COMPUTER VISION

Monday, 14 May 2:30 pm

Time allowed: 3:00 hours

**There are FIVE questions on this paper.**

**Answer FOUR questions.**

*All questions carry equal marks.*

**Any special instructions for invigilators and information for candidates are on page 1.**

| | | |
|---|---|---|
| Examiners responsible | First Marker(s) : | T-K. Kim |
| | Second Marker(s) : | C. Ling |

1. (Probability Theory and Graphical Model)

a) Consider two image classes each containing three features. The random variable of image class denoted by $c$ can take either $C_1$ or $C_2$. The feature denoted by $f$ can take one of the three possible values $F_1$, $F_2$ or $F_3$. Suppose $p(c = C_1) = 0.3$ and $p(c = C_2) = 0.7$, and $p(F_1|C_1) = 0.2$, $p(F_2|C_1) = 0.4$, $p(F_3|C_1) = 0.4$, $p(F_1|C_2) = 0.6$, $p(F_2|C_2) = 0.2$, $p(F_3|C_2) = 0.2$.

i) What is the overall probability of having the feature 1 i.e. $p(f = F_1)$?

[ 7 ]

ii) Given that we have a feature $F_1$, what is the probability that the image class we have is $C_2$ and $C_1$, i.e. $p(c=C_2|f=F_1)$ and $p(c=C_1|f=F_1)$?

[ 8 ]

b) For the directed probabilistic graph as shown in Figure 1.1(left), give the expression of joint distribution.

[ 5 ]

c) If $p(a,b|c) = p(a|c)p(b|c)$, the variables $a, b$ are statistically independent given $c$. Does the conditional independence hold or not in the graph in Figure 1.1(right)? Prove your answer.

[ 5 ]

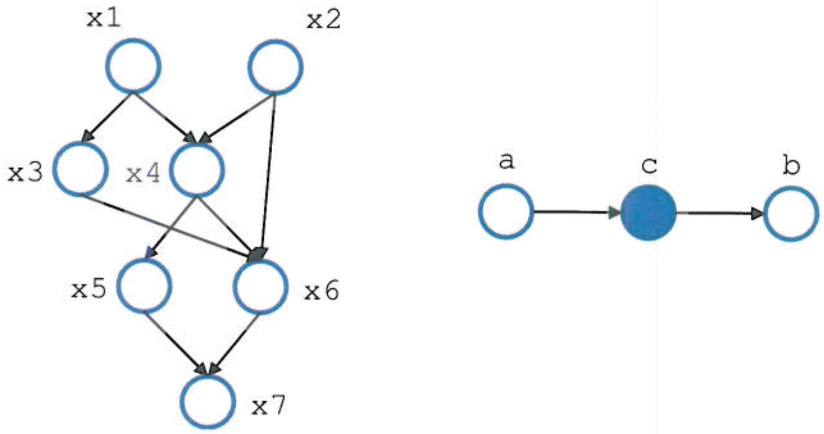

Figure 1.1

2. (Boosting and Object Detection)

$$
\begin{aligned}
E &= e^{-\alpha_m/2} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\alpha_m/2} \sum_{n \in \mathcal{M}_m} w_n^{(m)} \\
&= \left(e^{\alpha_m/2} - e^{-\alpha_m/2}\right) \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}) \neq t_n) + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}
\end{aligned}
$$

a) By differentiating the error function above with respect to $\alpha_m$, show that the parameters $\alpha_m$ in the AdaBoost algorithm are updated using

$$\alpha_m = \ln\left\{\frac{1-\varepsilon_m}{\varepsilon_m}\right\}$$

in which

$$\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}.$$

[ 10 ]

b) Explain the procedure of designing, learning and evaluating the robust real-time object detector (of Viola and Jones, 2001), by mentioning

i) the strong/weak classifiers and integral image,

ii) Adaboost,

iii) and the sliding window approach.

[ 15 ]

3. (Sparse Kernel Machine)

a) Linear discriminant function takes the form of $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$ where $\mathbf{w}$ is called a *weight vector* and $w_o$ a *bias*. Use Figure 3.1.

i) For a point $\mathbf{x}$ on the decision surface, show that the normal distance from the origin to the decision surface is $-\frac{w_0}{||\mathbf{w}||}$.

[ 5 ]

ii) For an arbitrary point $\mathbf{x}$ and its orthogonal projection onto the decision surface $\mathbf{x}_\perp$ so that $\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{||\mathbf{w}||}$, show $r = \frac{y(\mathbf{x})}{||\mathbf{w}||}$.

[ 7 ]

b) SVM for the two-class classification problem takes the form of $y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$ where $\phi(\mathbf{x})$ denotes a feature space transformation, $b$ the bias parameter. Suppose that we have training vectors $\mathbf{x}_n, n = 1, \ldots$ and their target values $t_n, n = 1, \ldots$ where $t_n \in \{-1, 1\}$. Using the notion of margin maximisation and the formulation given in 3-a)-ii), show the SVM solution is found by

$$\arg\max_{\mathbf{w},b}\left\{\frac{1}{||\mathbf{w}||}\min_n[t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)]\right\}.$$

[ 8 ]

c) In order to classify a new data point $\mathbf{x}$ using the trained model, we evaluate the sign of $y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b$, where $a_n$ are lagrange multipliers and $k$ is a kernel function. Discuss the complexity of the linear and nonlinear SVMs.

Linear kernel: $k(\mathbf{x}, \mathbf{x}_n) = \mathbf{x}^T\mathbf{x}_n$

Gaussian kernel: $k(\mathbf{x}, \mathbf{x}_n) = \exp\left(-||\mathbf{x} - \mathbf{x}_n||^2/2\sigma^2\right)$

[ 5 ]

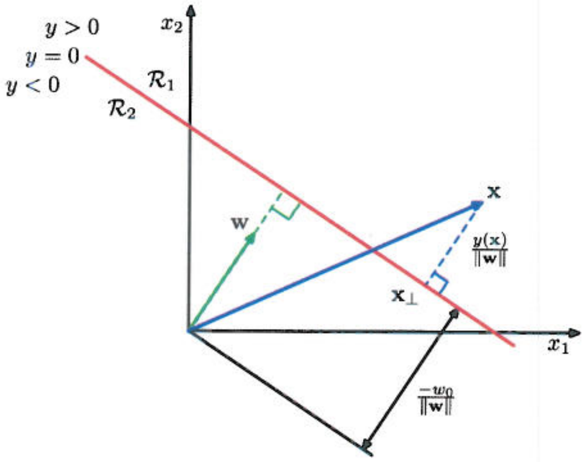

Figure 3.1

4. (Manifold Learning and Face Recognition) Explain the *Eigenface* method by addressing the followings. For each of the followings, use the notations below, in order to show formulations, indicate the dimensions of data vectors and matrices used, and explain their meanings.

$\mathbf{x}$ : face image vector, $\mathbf{X}$ : data matrix, $\mathbf{C}$ : covariance matrix

$\mathbf{U}$ : eigen vector matrix, $\Lambda$ : eigen value matrix, $\mathbf{Z}$ : project coefficient matrix

$\widetilde{\mathbf{X}}$ : reconstrcuted data matrix, $n$ : the number of images

a) Conversion of face images of $w \times h$ pixels into column vectors.

[ 5 ]

b) Construction of the covariance matrix.

[ 5 ]

c) Eigenvector and eigenvalue analysis.

[ 5 ]

d) Projection of data onto the eigen-subspace.

[ 5 ]

e) Reconstruction.

[ 5 ]

5. (Gaussian Process and Pose Estimation) In the Gaussian Process for regression, the observed target values are given by $t_n = y_n + \varepsilon_n$, where $\varepsilon_n$ is a random noise variable and $y_n = y(\mathbf{x}_n) = \mathbf{w}^T \phi(\mathbf{x}_n)$. From given a data set $\mathbf{x}_1, ..., \mathbf{x}_N$, we denote the vector $\mathbf{y} = (y_1, ..., y_N)^T$. Then we have $\mathbf{y} = \boldsymbol{\Phi}\mathbf{w}$ where $\boldsymbol{\Phi}$ is the matrix with elements $\Phi_{nk} = \phi_k(\mathbf{x}_n)$.

a) For a given prior distribution over $\mathbf{w}$ by an isotropic Gaussian of the form $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$, show that the marginal distribution $p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K})$ where $\mathbf{K}$ a Gram matrix (i.e. show $\mathbb{E}[\mathbf{y}] = \mathbf{0}$ and $\text{cov}[\mathbf{y}] = \mathbb{E}[\mathbf{y}\mathbf{y}^T] = \mathbf{K}$).

[ 5 ]

b) The joint distribution of the target values $\mathbf{t} = (t_1, ..., t_N)^T$ conditioned on $\mathbf{y} = (y_1, ..., y_N)^T$ is an isotropic Gaussian of the form $p(\mathbf{t}|\mathbf{y}) = \mathcal{N}(\mathbf{t}|\mathbf{y}, \beta^{-1}\mathbf{I}_N)$. Using $p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K})$ and the theorem below, derive $p(\mathbf{t})$.

Theorem: Given

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mu, \Lambda^{-1})$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}),$$

the marginal distribution becomes

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mu + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\Lambda^{-1}\mathbf{A}^T).$$

[ 5 ]

c) Explain the meaning of placing the prior distribution $p(\mathbf{w})$ w.r.t. generalisation and in relevance to the margin term in the maximum margin classifier.

[ 7 ]

d) Explain the post estimation problem in computer vision by answering each of the following questions.

i) What are typical input and output?

ii) Is this a classification problem or regression problem? Justify your answer.

iii) Exemplify the multi-valued problem.

[ 8 ]

# Exam Answers (2012): EE462, EE9SO25 Machine Learning for Computer Vision

1-a)-i)

$$p(F_1) = p(F_1|C_1)p(C_1) + p(F_1|C_2)p(C_2) = 0.2 \times 0.3 + 0.6 \times 0.7 = 0.48$$

1-a)-ii) Reversing the conditional probability by Bayes' theorem gives

$$p(C_2|F_1) = \frac{p(F_1|C_2)p(C_2)}{p(F_1)} = \frac{0.6 \times 0.7}{0.48} = \frac{7}{8}$$

$$p(C_1|F_1) = 1 - 7/8 = 1/8$$

1-b) The joint distribution is given as

$$p(x_1)p(x_2)p(x_3|x_1)p(x_4|x_1,x_2)p(x_5|x_4)p(x_6|x_2,x_3,x_4)p(x_7|x_5,x_6).$$

1-c) We obtain

$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(c|a)p(b|c)}{p(c)} = p(a|c)p(b|c)$$

So, we get the conditional independence property $a \perp b|c$.

2-a) If we differentiate the error function above w.r.t. $\alpha_m$ we obtain

$$\frac{\partial E}{\partial \alpha_m} = \frac{1}{2}\left( (e^{\alpha_m/2} + e^{-\alpha_m/2}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n) - e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)} \right).$$

Setting this equal to zeros and rearranging, we get

$$\frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}} = \frac{e^{-\alpha_m/2}}{e^{\alpha_m/2} + e^{-\alpha_m/2}} = \frac{1}{e^{\alpha_m} + 1}.$$

Using $\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$, we can rewrite this as

$$\frac{1}{e^{\alpha_m} + 1} = \varepsilon_m,$$

which can be further rewritten as

$$e^{\alpha_m} = \frac{1 - \varepsilon_m}{\varepsilon_m},$$

from which the update equation for $\alpha_m$ follows directly.

2-b) The strong boosting classifier is given as the linear weighted sum of weak classifier responses: $H(x) = \sum_m \alpha_m h_m(x)$. Each weak classifier is Haar-basis like function using a rectangle filter response, whose computation is accelerated on an integral image.

Adaboost, a sequential learning algorithm, is used to learn the set of weak classifiers. It chooses the best weak classifiers sequentially, and the number of weak classifier is set manually or using a a stop-criterion such as error rate on train data.

Given a test image, we typically evaluate sub-windows at every possible pixels and scales. For each sub-window, we apply the learnt boosting algorithm to tell if it contains or not an object of interest.

3-a)-i) For a point $\mathbf{x}$ on the decision surface, $y(\mathbf{x}) = 0$. So the normal distance from the origin to the decision surface is

$$\frac{\mathbf{w}^T\mathbf{x}}{||\mathbf{w}||} = -\frac{w_0}{||\mathbf{w}||}.$$

3-a)-ii) Multiplying both sides by $\mathbf{w}^T$ and adding $w_0$, we have $\mathbf{w}^T\mathbf{x} + w_0 = \mathbf{w}^T\mathbf{x}_\perp + w_0 + r\frac{\mathbf{w}^T\mathbf{w}}{||\mathbf{w}||}$. Making use of $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$ and $y(\mathbf{x}_\perp) = \mathbf{w}^T\mathbf{x}_\perp + w_0 = 0$, we have

$$y(\mathbf{x}) = 0 + r||\mathbf{w}|| \qquad \rightarrow \qquad r = \frac{y(\mathbf{x})}{||\mathbf{w}||}$$

3)-b) The perpendicular distance of a point $\mathbf{x}$ from a hyperplane $y(\mathbf{x}) = 0$ is $|y(\mathbf{x})|/||\mathbf{w}||$. As we assumed $t_n y(\mathbf{x}_n) > 0$ for all $n$, the distance of a point $\mathbf{x}_n$ to the decision surface is

$$\frac{t_n y(\mathbf{x}_n)}{||\mathbf{w}||} = \frac{t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)}{||\mathbf{w}||}$$

The margin is the minimum perpendicular distance, and we wish to find $\mathbf{w}$ and $b$ that maximises the margin. The solution is therefore given as

$$\arg\max_{\mathbf{w},b}\left\{\frac{1}{||\mathbf{w}||}\min_n[t_n(\mathbf{w}^T\phi(\mathbf{x}) + b)]\right\}$$

3)-c) In the case of using a linear kernel,

$$\begin{aligned} y(\mathbf{x}) &= \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b = \sum_n a_n t_n \mathbf{x}^T\mathbf{x}_n + b \\ &= \mathbf{x}^T\sum_n a_n t_n \mathbf{x}_n + b = \mathbf{x}^T\mathbf{z} + b \end{aligned}$$

where $\mathbf{z}$ is pre-computable. Therefore, for a new data point, we need to take the dot-product between $\mathbf{x}$ and $\mathbf{z}$, which takes $O(d)$ where $d$ is the vector dimension. In the nonlinear SVM, those computations in the exponential can not be taken out thus requiring the dot-product between $\mathbf{x}$ and every $\mathbf{x}_n$, which takes $O(dN)$.

4-a) All pixels are raster-scanned and concatenated into a column vector $\mathbf{x} \in \mathbf{R}^m$ where $m = w \times h$. Thus, the data matrix whose columns are the image vectors is $\mathbf{X} \in \mathbf{R}^{m \times n}$ where $n$ is the number of face images.

4-b) $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$ where $\mathbf{X} = [\mathbf{x}_1 - \mu, ...\mathbf{x}_n - \mu]$, $\mu$ is the mean vector. $\mathbf{C} \in \mathbf{R}^{m \times m}$

4-c) $\mathbf{CU} = \Lambda\mathbf{U}$, where $\mathbf{U} \in \mathbf{R}^{m \times d}$ and $d < n$. Each eigenvector has the same image dimension i.e. $m = w \times h$.

4-d) $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$, $\mathbf{Z} \in \mathbf{R}^{d \times n}$ and typically $d << m$ thus the dimension reduction occurs.

4-e) $\widetilde{\mathbf{X}} = \mathbf{UZ}$, $\widetilde{\mathbf{X}} \in \mathbf{R}^{m \times n}$.

5-a)

$$\mathbb{E}[\mathbf{y}] = \Phi\mathbb{E}[\mathbf{w}] = \mathbf{0}$$

$$\text{cov}[\mathbf{y}] = \mathbb{E}[\mathbf{y}\mathbf{y}^T] = \Phi\mathbb{E}[\mathbf{w}\mathbf{w}^T]\Phi^T = \frac{1}{\alpha}\Phi\Phi^T = \mathbf{K}$$

where $\mathbf{K}$ is the Gram matrix that has

$$K_{nm} = k(\mathbf{x}_n, \mathbf{x}_m) = \frac{1}{\alpha}\phi(\mathbf{x}_n)^T\phi(\mathbf{x}_m).$$

5-b) By analogy,

$$\mu = 0, \quad \Lambda^{-1} = \mathbf{K}, \quad \mathbf{A} = \mathbf{I}, \quad \mathbf{b} = 0, \quad \mathbf{L}^{-1} = \beta^{-1}\mathbf{I}.$$

Thus,

$$p(\mathbf{t}) = \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{C})$$

where

$$C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1}\delta_{nm}.$$

5-c) By placing an isotropic Gaussian prior distribution, all elements of $\mathbf{w}$ are encouraged to take small values near to zero. The margin term to maximise for good generalisation is proportional to $1/||\mathbf{w}||$, thus we need to minimise $||\mathbf{w}||$.

5-d) Typical input is image silhouette feature $\mathbf{z}$ and output is a set of joint angles $\theta$. As the output has continuous variables, it is a regression problem. Often we have multiple $\theta$ values for a given $\mathbf{z}$, i.e. many-to-one mapping, due to ambiguities of silhouettes in the projection of 3D objects into 2D image planes.