EE4-62
EE9-CS7-28
EE9-SO25

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2014

MSc and EEE/EIE PART IV: MEng and ACGI

Corrected Copy

MACHINE LEARNING FOR COMPUTER VISION

Monday, 28 April 10:00 am

Time allowed: 3:00 hours

There are FOUR questions on this paper.

Answer ALL 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

© Imperial College London

1. Object Detection and Boosting

a) (Viola-Jones object detector, 2001)

(i) Treat the boosting algorithm like a black-box and explain the inputs and outputs of the boosting algorithm for object detection:

- the data x_n and labels t_n, $n = 1, ..., N$,
- the feature pool of size M' (explain the motivations on the use of the particular type of features),
- the examples of weak classifiers selected, $y_m(x), m = 1, ..., M << M'$ (explain how to form a binary weak classifier $y_m(x)$ using the feature above).

[5]

(ii) Describe the evaluation procedure for given an image of $W \times H$ pixels in Figure1.1 using the learnt boosting classifier above: explain the scale search, scanning windows, way to evaluate a certain window on the integral image by the example boosting classifier, and way to do final detection.

[5]

b) Below is the AdaBoost algorithm.

Initialise ...
For $m = 1, ..., M$
(1) Learn a classifier $y_m(x)$ s.t. ...
(2) Evaluate $\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$ and set $\alpha_m = \ln\left\{\frac{1-\varepsilon_m}{\varepsilon_m}\right\}$
(3) Update the data weights $w_n^{(m+1)} = ...$
Make predictions using $Y_M(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(x)\right)$

(i) Complete the step (1) above and derive the required formulation using $E = \sum_{n=1}^{N} \exp\{-t_n f_m(x_n)\} = \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(x_n)\right\}$, where $w_n^{(m)} = \exp\{-t_n f_{m-1}(x_n)\}$.

[5]

(ii) Complete the step (3) and derive the formulation. [5]

c) (Real-time evaluation) Discuss the challenges in real-time object detection as to the number of data points to evaluate (for an $W \times H$ pixel image), the required test time per data point, and the required precision-recall rate when considering the number of positive and negative class samples. Explain the principles how the boosting cascade further accelerates the run time.

[5]

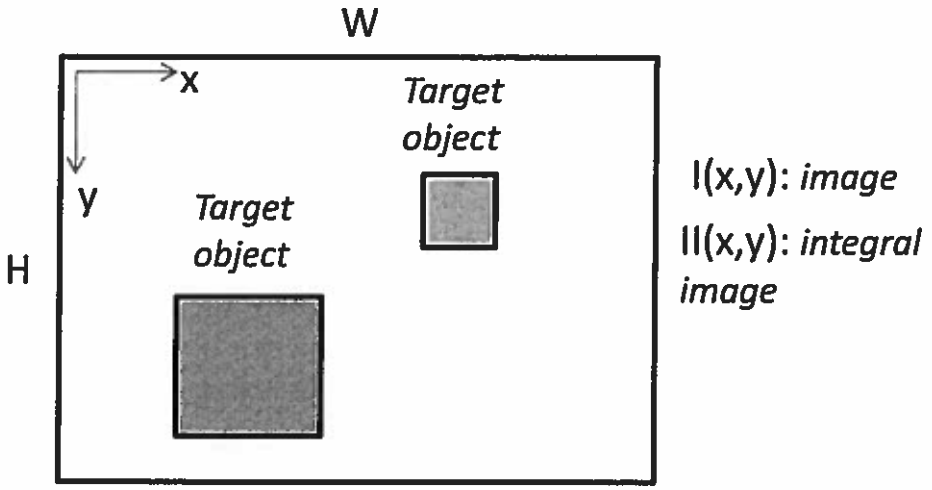

Figure 1.1 Object detection

2. Bag of Words, K-means and GMM

a) (Bag of Words)

(i) Describe the process of the visual dictionary learning: treat the K-means algorithm like a black-box and explain the inputs and outputs of the K-means. Assume that we have M images and extract M' features per image, explain the data acquisition procedure, meaning of K, typical way to determine the value of K, and consequences of setting K too small or too large.

[3]

(ii) Explain how to represent an image using the obtained codebook of K. Discuss the time complexity of this process as to K, M' (the number of features in an image), and D (the dimension of visual words): the similarity between two visual words x_1, x_2 is computed by $\sqrt{\sum_{d=1}^{D}(x_1^d - x_2^d)^2}$, taking $O(D)$.

[4]

b) (K-means and GMM)

The objective function of the K-means is $J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}||x_n - \mu_k||^2$.

(i) Explain the meaning of J and r_{nk} (1-of-K coding). Describe the inputs and outputs of the K-means algorithm, including any parameters and variables munally set or initialised. Indicate what variables are optimised by J.

[4]

Below is the K-means algorithm.

Iterate the two steps until convergence.
Step 1: minimise J with respect to ..., while keeping ... fixed.

$$r_{nk} = \begin{cases} 1, & \text{if } k = \arg\min_j ||x_n - \mu_j||^2 \\ 0, & \text{otherwise} \end{cases}$$

Step 2: minimise ... with respect to ..., while keeping ... fixed.

(ii) Complete the step 1 above, and derive the given formulation for r_{nk}.

[5]

(iii) Complete the step 2, and derive the required formulation for μ_k.

[4]

(iv) Give the example stopping criteria, and answer whether or not the algorithm provides a convergence proof, a global optimal solution and it depends on initialisations.

[2]

(v) Explain the GMM (Gaussian Mixture Model) and its parameters in comparison with the K-means.

[3]

3. Polynomial curve fitting, Gaussian Process

a) See Figure 3.1. Given a training set $\mathbf{x} \equiv (x_1, ..., x_N)^T$ and their target values $\mathbf{t} \equiv (t_1, ..., t_N)^T$, our goal is to discover the underlying function to predict the value $\widehat{t}$ of some new value $\widehat{x}$. We fit the data to a polynomial function $y(x, \mathbf{w})$.

(i) Write the expression for the function $y(x, \mathbf{w})$ with the order M.

[2]

(ii) Write the error function $E(\mathbf{w})$.

[2]

(iii) When M=1, find the optimal solution $\mathbf{w}^*$, by setting the derivative of the error function w.r.t. $\mathbf{w}$ to zero.

[6]

(iv) In Figure 3.1, draw example fitted functions with M=0,1,3,9. Discuss the underfitting/overfitting in each case.

[3]

(v) To relieve the overfitting above, write the modified error function $E(\mathbf{w})$, and explain the meaning of the regularsed term.

[2]

b) In the Gaussian Process (GP), the goal is to predict the target value t_{N+1} for a new input x_{N+1}, given a set of training data $x_1, ..., x_N$ and $\mathbf{t}_N = (t_1, ..., t_N)^T$. It requires the evaluation of $p(t_{N+1}|\mathbf{t}_N)$, where we omit the data vectors for notational simplicity. The joint distribution is given by $p(\mathbf{t}_{N+1}) = \mathcal{N}(\mathbf{t}_{N+1}|\mathbf{0}, \mathbf{C}_{N+1})$ where $\mathbf{C}_{N+1}$ is an $(N+1) \times (N+1)$ covariance matrix s.t.

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix}$$

where the vector $\mathbf{k}$ has elements $k(x_n, x_{N+1})$ for $n = 1, ..., N$, and the scalar $c = k(x_{N+1}, x_{N+1}) + \beta^{-1}$.

(i) Compute the mean and covariance of $p(t_{N+1}|\mathbf{t}_N)$ by the following theorem.

[5]

Given

$$x = \begin{bmatrix} x_a \\ x_b \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}$$

, we have

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b)$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

---

c) Show an example predictive distribution in Figure 3.1, and explain the advantage(s) of GP over Polynomial curve fitting.

[5]

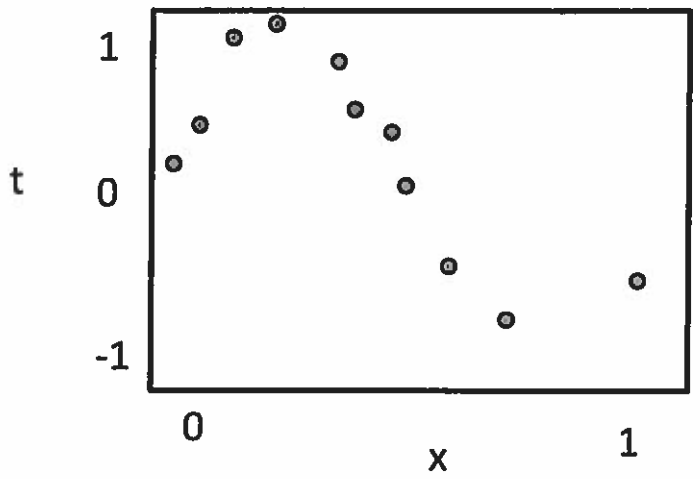

Figure 3.1 Regression.

4. Sparse Kernel Machine, PCA, and their applications

Design a face detector and recogniser: For a given image of $W \times H$ pixels, we first apply the Support Vector Machine (SVM) to locate (or detect) all faces in the image, then the Principal Component Analysis (PCA) i.e. the eigenfaces on the detected faces to recognise face identities.

a) (SVM for face detection)

(i) Training: Describe the training data acquistion process, and discuss learing a decision boundary by linear and nonlinear SVMs.

[5]

(ii) Testing: Write the SVM evaluation equation, explain how we apply this to an image of $W \times H$ pixels. Discuss the time complexity of linear and nonlinear SVMs.

[7]

b) (Eigenfaces for face recognition) Assume we are given two face identities, ID1 and ID2, we learn an eigen-subspace per each ID, and use them for face identification.

(i) Given the eigen-components $\mathbf{u}_i, i = 1, ..., M$, write the expression for the projection coefficients of a face image x.

[3]

(ii) Write the expression for the reconstruction of an image x.

[3]

(iii) Describe the overall process of face identification using the above.

[7]