UNIVERSITY OF BRISTOL

January 2022 (Mock exam) Examination Period

Department of Computer Science

3rd Year Examination for the Degrees of
Bachelor in Computer Science
Master of Engineering in Computer Science

COMS30033
Machine Learning (Mock exam)

TIME ALLOWED:
2 Hours
plus 30 minutes to allow for collation and uploading of answers.

Answers

Other Instructions

1. THIS IS A MOCK EXAM!!
2. Instruction 1: The exam is divided into two parts (Part 1 and Part 2). The first contains 10 short questions worth 5 marks each and the second 5 long questions worth 10 marks each. Both parts cover the full material taught in the unit.
3. Instruction 2: Note that sharing information with colleagues is strictly forbidden and that we have a set of measures in place to identify cases of plagiarism.
4. Instruction 3: This is NOT a open book exam.

Part 1: Short questions (5 marks each)

Question 1 (5 marks)
Which form(s) of learning requires a explicit target?

Solution: Supervised learning is the clear answer as supervised learning requires by default a explicit target to which we compare the model output.

Question 2 (5 marks)
What is the key difference between parametric and non-parametric models?

Solution: In non-parametric models the number of parameters grows with the data, whereas parametric models assume a fixed number of parameters.

Question 3 (5 marks)
Figure 1 shows a Bayesian network structure (i.e. directed acyclic graph). Write down all pairs of variables which are independent conditional on F.

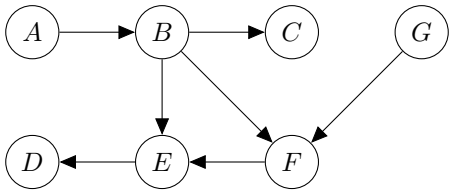

Figure 1: Directed acyclic graph for Question 3

Solution: There are no variables which are independent given F. One can see this by just considering every possible pair and seeing if there is at least one unblocked path. For example, between D and G there are two paths: D, E, B, F, G and D, E, F, G. The second is blocked given F but the first is not since F is a collider on that path.

Question 4 (5 marks)
Let $(1, 5, 2)$ and $(1, 3, 1)$ be the first two principal components computed from some dataset. Let $x_1 = (2, 6, 0)$ and $x_2 = (-6, 1, 2)$ be two datapoints. Compute a 2-d approximation for each of x_1 and x_2 using the two principal components.

Solution: If $\mathbf{x}_n$ is some D-dimensional vector and $\mathbf{u}_i$ is the ith principal component then we have $\mathbf{x}_n = \sum_{i=1}^{D} (\mathbf{x}_n^T \mathbf{u}_i)\mathbf{u}_i$ (this is (12.9) from Bishop). A reasonable 2-d approximation is just to use $(\mathbf{x}_n^T \mathbf{u}_1)$ and $(\mathbf{x}_n^T \mathbf{u}_2)$. (If we were given the mean of the data then we could get a better 2-d approximation, see Bishop 12.1.3, but we are not given the mean, and moreover this better approximation was not covered in the lectures.) For $x_1 = (2, 6, 0)$ the 2-d approximation is $(2\times1+6\times5+0\times2, 2\times1+6\times3+0\times1) = (32, 20)$. For $x_2 = (-6, 1, 2)$ the 2-d approximation is $(-6\times1+1\times5+2\times2, -6\times1+1\times3+2\times1) = (3, -1)$.

Question 5 (5 marks)

Suppose you were using EM to estimate the parameters of a mixture of 3 Gaussians. Let x_i be a training datapoint where $\mathcal{N}(x_i|\mu_1, \Sigma_1) = 0.2$, $\mathcal{N}(x_i|\mu_2, \Sigma_2) = 0.1$, $\mathcal{N}(x_i|\mu_3, \Sigma_3) = 0.4$. μ_j and Σ_j are the current parameter values for the jth Gaussian. Let $\gamma_{i1} = 0.4$, $\gamma_{i2} = 0.2$, $\gamma_{i3} = 0.4$ be the current *responsibilites* for x_i. What are the current values for the mixing coefficients.

Solution: The equation for computing responsibilities from mixing coefficients and Gaussian is:

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

But here we have to go 'the other way' and get the π_k values. Abbreviate the denominator of the above equation to M, so $M\gamma(z_{nk}) = \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Plugging in the values we have been given produces: $0.4M = 0.2\pi_1$, $0.2M = 0.1\pi_2$, $0.4M = 0.4\pi_3$. Since we must have $\pi_1 + \pi_2 + \pi_3 = 1$ it is then easy to deduce that $\pi_1 = 0.4$, $\pi_2 = 0.4$ and $\pi_3 = 0.2$.

Question 6 (5 marks)

When using MCMC what is *burn-in* (2 marks) and we do we use it (3 marks)?

Solution: *Burn-in* is the process of throwing away early samples produced by MCMC. We do this since (even if MCMC is working well) MCMC samples from a *sequence* of probability distributions which converge to the target distribution. So typically, samples which are early in the sequence will be sampled from distributions which may not be close to the target distribution.

Question 7 (5 marks)

(a) Must kernel function always return non-negative values? If yes, explain why. If no, find values of x, y and k such that $k(x, y) < 0$, and where k is a valid kernel function.

(b) Explain why kernels must symmetric: i.e. $k(x, y) = k(y, x)$.

Solution:

Kernels can have negative values. For example with the linear kernel $k(x, y) = x^T y$ we could choose, say $x = (1, 0)^T$, $y = -(1, 0)^T$.

If k is a kernel then $k(x, y) = (\Phi(x))^T \Phi(y)$ where Φ is some feature map. But inner product is symmetric so $k(x, y) = (\Phi(x))^T \Phi(y) = (\Phi(y))^T \Phi(x) = k(y, x)$

Question 8 (5 marks)

This question is about decision trees. We have a dataset containing three types of clothing with three features:

Data point index	X	Y	Z	Class Label
1	0	0	0	+1
2	1	0	1	-1
3	1	1	0	-1
4	1	1	1	+1

(a) What is the minimum depth of tree that would give zero training error? Explain your reasoning.

(b) In general, are there any problems that might arise if we grow a decision tree until there is zero training error? Explain your reasoning and give some ways to avoid this problem with CART trees.

Solution: The minimum depth for zero training error is 2. No single feature splits the data into the two classes, but several combinations are possible: split on X, then Z; split on Y then Z; split on Z then Y.

As with any model, reaching zero training set error often means we overfitted to the training set, and the model may not perform as well on the test data. The general solution is to simplify the tree by excluding decision nodes that are not important. With CART, we can remove nodes after training using pruning, or we can stop growing the tree once a maximum depth or minimum number of samples per leaf has been reached.

Question 9 (5 marks)

Explain in words how the Adaboost method applies weights to training data instances, and why it does this?

Solution: Adaboost trains an ensemble of classifiers sequentially. After training each classifier, Adaboost identifies the training instances that were incorrectly classified by the current classifier. It then increases the weights of these instances so that they are given more importance when training the next classifier. The aim is for the next classifier to classify these to classify these instances correctly. Thereby, Adaboost produces an ensemble of classifiers that make different errors, which is required to reduce the expected error of the ensemble.

Question 10 (5 marks)

Labelled data is important for supervised machine learning and evaluating machine learning systems. A common solution for obtaining large labelled datasets is crowdsourcing. Briefly explain two disadvantages or challenges of using crowdsourcing to obtain labelled data.

Solution:

Crowdworkers are not usually trained experts, and their background is usually unknown. This can lead to lower quality data than expert annotators, with more frequent labelling errors. In contrast, expert annotators may be better able to label difficult instances.

Usually, multiple crowdworkers annotated each data point. This means we need an additional step to aggregate the labels, i.e., to choose the correct answer when the workers disagree.

Spammers who do not provide informative labels also add noise to the data. A method is needed to select only reliable workers or to filter out spam labels after annotation is complete.

Part 2: Long questions (10 marks each)

Question 11 (10 marks)

(a) Which of the following models overfit, underfit and provide an adequate fit to the data? (3 marks)

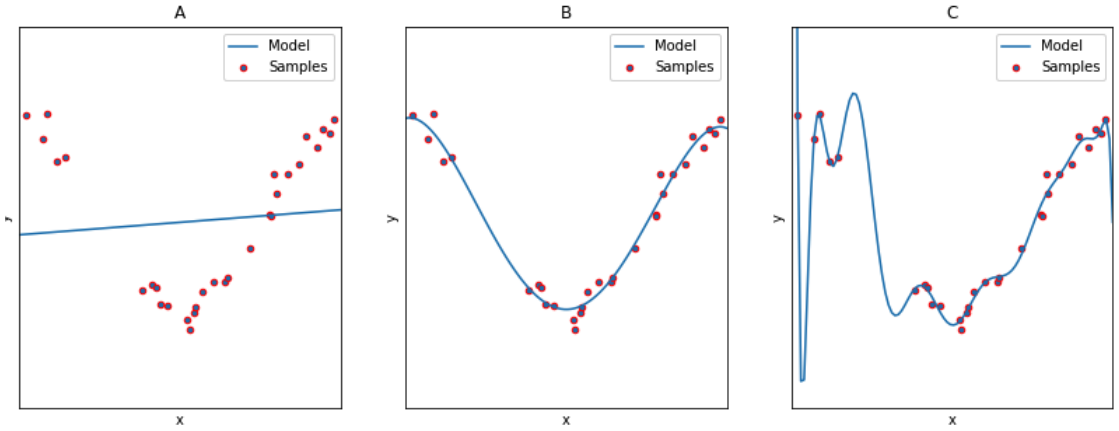

Figure 2: Examples of data (red scatter plot) with three different models (A,B,C; blue solid line).

Solution: Model A underfits the data, whereas the model in C ia case of overfitting. Model B provides a good fit to the data.

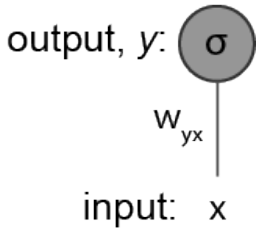

Figure 3: Schematic of a simple artificial neural network.

(b) Given a very simple neural network with only one input x and one output neuron y with a linear activation f function, where $x = 0.5$, the target $t = 1$ and a standard mean squared error ($E = \frac{1}{2}(y - x)^2$) what is the exact value for Δw_{yx} ? Assume that the current $w_{yx} = -0.5$ and that there are no bias. Your answer only needs to be approximate (e.g. 0.06). You should use the chain rule as discussed in the lectures. (7 marks)

Solution: Using the chain rule you should obtain: $\frac{\partial E}{\partial w_{yx}} = (y-t)f'(w_{yx}x)x$, where $f'(x) = 1$ is the derivative of the linear activation function. The exact solution for

the change in the weight is given by $\Delta w_{yx} = -\frac{\partial E}{\partial w_{yx}}$ is : $-(f(-0.5 \times 0.5) - 1) \times f'(-0.5 \times 0.5) \times -0.5) = -0.625$.

Question 12 (10 marks)

(a) Consider running k-means on the following datapoints: $x_1 = (0, -1)$, $x_2 = (1, 2)$, $x_3 = (1, 1)$, $x_4 = (2, 1)$. Suppose $k = 2$ and assume that initially x_1 and x_2 are assigned to cluster 1 and x_3 and x_4 are assigned to cluster 2. After running one iteration of the k-means algorithm to which cluster are the datapoints assigned? (5 marks)

(b) Give two advantages of using Gaussian mixtures over k-means for clustering and one disadvantage. (5 marks)

Solution:

The mean for cluster 1 is $(0 + 1, -1 + 2)/2 = (0.5, 0.5)$ and the mean for cluster 2 is $(1 + 2, 1 + 1)/2 = (1.5, 1)$. So we just assign each datapoint to whichever cluster has the nearest mean. For x_1 the squared distances are $0.5^2 + 1.5^2$ versus $1.5^2 + 2^2$, so it is assigned to cluster 1. For x_2 the squared distances are $0.5^2 + 1.5^2$ versus $0.5^2 + 1^2$, so it is assigned to cluster 2. For x_3 the squared distances are $0.5^2 + 0.5^2$ versus $0.5^2 + 0^2$, so it is assigned to cluster 2. For x_4 the squared distances are $1.5^2 + 0.5^2$ versus $0.5^2 + 0^2$, so it is assigned to cluster 2.

Gaussian mixtures allow soft clustering of datapoints which better reflects any uncertainty over which is the right cluster for a datapoint. (It can always be easily converted to a hard clustering by just choosing the most probable Gaussian for each datapoint.) Also, Gaussian mixtures allow more flexibility in the shape of clusters since we not only have the mean of each Gaussian but also its covariance matrix. A disadvantage is that fitting a Gaussian mixture is more computationally demanding than running k-means.

Question 13 (10 marks)

(a) Explain what *slack variables* in support vector machines (SVMs) are. (5 marks)

(b) Is it correct to call SVMs a *nonparametric* method? Explain your answer. (5 marks)

Solution:

A slack variable ξ measures how far on the 'wrong' side of the margin a datapoint is. Datapoints on or inside the correct margin boundary have a value of $\xi = 0$. Those on the incorrect side of the margin but on the correct side of the decision boundary have $\xi > 0$ but $\xi < 1$. Those on the wrong side of the decision boundary have $\xi > 1$.

SVMs are nonparametric. In a parametric method a fixed number of parameters are learned from the data: the number of parameters does not change with the size of the data. However with SVMs, the number of support vectors (and thus dual parameters) will increase (in general) as the dataset size increases.

Question 14 (10 marks)

This question is about hidden Markov models (HMM). Consider a hidden Markov model (HMM) with the following transition matrix and initial state probability estimates:

$$A = p(z_{n+1}|z_n) = \begin{matrix} 0.3 & 0.7 \\ 0.2 & 0.8 \end{matrix} \tag{1}$$

$$\pi = p(z_1) = [0.3, 0.7] \tag{2}$$

Rows in A correspond to values of z_n and columns to values of z_{n+1}.

The observations are discrete and can take values X, Y or Z. The model also has the following emission probabilities for the observations:

$$p(x_n = i|z_n = j) = \begin{matrix} x_n = i & z_n = 1 & z_n = 2 \\ \text{X} & 0.4 & 0.1 \\ \text{Y} & 0.1 & 0.5 \\ \text{Z} & 0.5 & 0.4 \end{matrix}$$

We observe the sequence X, Y.

(a) Use the parameters to compute the probability distribution $p(x_1 = X, x_2 = Y, z_2 = 1)$.

(b) Suppose we want to compute the probability that the next state is $z_3 = 1$. Briefly state explain the first order Markov assumption and how it applies to this computation.

Solution:

(a) $p(x_1 = X, x_2 = Y, z_2 = 1)$

$= p(z_1 = 1|\pi)p(x_1 = X|z_1 = 1)p(z_2 = 1|z_1 = 1)p(x_2 = Y|z_2 = 2) + p(z_1 = 2|\pi)p(x_1 = X|z_1 = 2)p(z_2 = 1|z_1 = 2)p(x_2 = Y|z_2 = 2)$

$= 0.005$

(b) The first order Markov assumption assumes that future states depend only on the current state, not on its predecessors. Here, this means we need to consider only $p(z_3 = 1|z_2)$ rather than $p(z_3 = 1|z_1, z_2)$.

Question 15 (10 marks)

This question is about linear dynamical systems. Suppose you are using a linear dynamical system to predict a continuous state variable, z_n. The model parameters have already been learned using expectation maximisation. For a new time-step, n, you have a Gaussian prior over z_n with mean 0 and variance 100. You observe a noisy sensor measurement, x_n, then use it to obtain a posterior distribution over z_n. The observation $x_n = 10$ with noise variance 1.

(a) What kind of distribution is the posterior distribution over z_n and which method can you use to compute it?

(cont.)

(b) Will the posterior mean of z_n be closer to 0 or 10 and why?

(c) If we now observe x_{n+1} and want to update the posterior over z_n, what method do we need to use and why?

Solution:

(a) The posterior is Gaussian and can be computed using the Kalman filter.

(b) The posterior mean is closer to 10, as this is the observation value x_n, which had a relatively small noise variance, whereas 0 is the prior mean and the prior variance is relatively large.

(c) We use a Kalman smoother to pass information back along the sequence from observation x_{n+1}. Since the LDS is a Markov model, the next state and therefore the next observation depend on the current state, so observing the next observation provides information about the current state.

END OF PAPER