The University of Melbourne

School of Computing and Information Systems

COMP90051
Statistical Machine Learning
2021 Semester 2 – Final Exam

Identical examination papers: None

Exam duration: 120 minutes

Reading time: 15 minutes

Upload time: 30 minutes additional to exam + reading; upload via Canvas

Late submissions: -1 against final subject mark per minute late, starting 120+15+30 minutes after exam start, up to 30 minutes late maximum. Late submissions permitted by provided OneDrive upload link.

Length: This paper has 6 pages including this cover page.

Authorised materials: Lecture slides, workshop materials, prescribed reading, your own projects.

Calculators: permitted

Instructions to students: The total marks for this exam is 120, corresponding to the number of minutes available. The mark will be scaled to compute your final exam grade.

This paper has three parts, A-C. You should attempt all the questions.

This is an open book exam (see authorised materials above). You should enter your answers in a Word document or PDF, which can include typed and/or hand-written answers. You should answer each question on a separate page, i.e., start a new page for each of Questions 1–7 – parts within questions do not need new pages. Write the question number clearly at the top of each page. You have unlimited attempts to submit your answer during the course of the exam, but only your last submission is used for marking.

You must not use materials other than those authorised above. You should not use private tutor notes, nor use materials off the Internet. You are not permitted to communicate with others for the duration of the exam, other than to ask questions of the teaching staff via the Exam support tool in Canvas (BigBlueButton). Your computer, phone and/or tablet should only be used to access the authorised materials, enter or photograph your answers, and upload these files.

Library: This paper is to be lodged with the Baillieu Library.

Continued overleaf . . .

COMP90051 Statistical Machine Learning Final Exam

Semester 2, 2021

Total marks: 120

Students must attempt all questions

Section A: Short Answer Questions [40 marks]

Answer each of the questions in this section as briefly as possible. Expect to answer each question in 1-3 lines.

Question 1: [40 marks]

(a) Suppose you have trained a *soft-margin support vector machine (SVM) with a RBF kernel*, and the performance is very good on the training set while very poor on the validation set. How will you change the hyperparameters of the SVM to improve the performance of the validation set? List two strategies. [5 marks]

(b) Compared with the *Root Mean Square Propagation (RMSProp)* algorithm, what's the main drawback of the *Adaptive Gradient (AdaGrad)* algorithm? [5 marks]

(c) What strategy makes the *Variational Autoencoder (VAE)* capable of applying gradient descent through the samples of latent representation z to the encoder? [5 marks]

(d) For the *recurrrent neural network (RNN)* that takes a sequence as the input, explain why we need to use *backpropogation through time* to update weights. [5 marks]

(e) Let E be the set of all *extremum estimators*, L be the set of all *maximum-likelihood estimators*, and M be the set of all *M-estimators*. Fill in the blanks in your answers with E, L, M to make the following expression correct: ___ $\subset$ ___ $\subset$ ___. [5 marks]

(f) We've seen that the *square-loss risk of a parameter estimate* $\hat{\theta}$ is $\mathbb{E}_\theta[(\theta - \hat{\theta})^2] = [B(\hat{\theta})]^2 + \mathrm{Var}(\hat{\theta})$ while the *square-loss risk of a supervised regression predictor* is $\mathbb{E}_{X,Y}[(Y - \hat{f}(X))^2] = (\mathbb{E}[Y] - [\hat{f}(X)])^2 + \mathrm{Var}(\hat{f}(X)) + \mathrm{Var}[Y]$ where the last term is known as the *irreducible error*. Note that in these risks both $\hat{\theta}$ and $\hat{f}$ implicitly depend on a random training set. Why is there no irreducible error term in the first square-loss risk, even though it appears in the second risk? [5 marks]

(g) What *objective function* is optimised during training of the *Gaussian mixture model (GMM)*? Give your answer as a mathematical expression. [5 marks]

(h) Explain why only one iteration of *Newton-Raphson* is required to train *linear regression*. [5 marks]

 Continued overleaf . . .

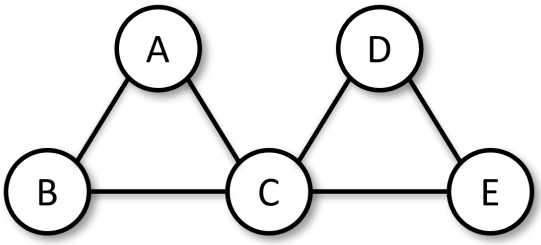
Section B: Method & Calculation Questions [60 marks]

In this section you are asked to demonstrate your conceptual understanding of methods that we have studied in this subject, and your ability to perform numeric and mathematical calculations.

Question 2: Probabilistic Graphical Models [14 marks]

Consider the following *undirected probabilistic graphical model (U-PGM)* on five Boolean-valued variables, where the *clique potentials* $f(A, B, C)$ and $G(C, D, E)$ are shown below.

A	B	C	f(A,B,C)
T	T	T	9
T	T	F	3
T	F	T	4
T	F	F	3
F	T	T	5
F	T	F	3
F	F	T	2
F	F	F	3



C	D	E	g(C,D,E)
T	T	T	0
T	T	F	0
T	F	T	0
T	F	F	0
F	T	T	2
F	T	F	4
F	F	T	4
F	F	F	1

(a) Using the given *clique potential tables*, calculate the *normalising constant* (aka *partition function*) for the joint distribution on A, B, C, D, E. [8 marks]

(b) Calculate $\Pr(A = F, B = F, C = F)$. You may leave your answer as a fraction. (If you were unable to answer the previous part, leave the constant as Z in your workings here.) [6 marks]

Question 3: Frequentist Parameter Estimation [12 marks]

Consider an i.i.d. sequence of random variables $X_1, X_2, \ldots$ coming from some distribution with mean θ. Consider a simple estimator $\hat{\theta}_n = \hat{\theta}(X_1, \ldots, X_n) = X_1$ of the mean. That is, use the first observation X_1 as the estimate and ignore the rest.

(a) Why is $\hat{\theta}_n$ an *unbiased estimator* of θ in general? [3 marks]

(b) Is the estimator $\hat{\theta}_n$ *consistent*? [3 marks]

(c) Explain why your answer to the previous part is correct. [6 marks]

Question 4: VC Dimension [14 marks]

In class we saw that the family of linear classifiers in $\mathbb{R}^2$ (the 2D plane) have Vapnik Chervonenkis (VC) dimension 3. This question asks you to consider *VC dimension* of an unrelated family of classifiers also *in the plane*. Consider the family $\mathcal{F}$ of *rectangle classifiers* parametrised by reals a, b, c, d where $a < b$ and $c < d$: for any instance $\mathbf{x} \in \mathbb{R}^2$ we have the classifier,

$$f_{abcd}(\mathbf{x}) \quad = \quad \begin{cases} +1\ , & \text{if } a \le x_1 \le b \text{ and } c \le x_2 \le d \\ -1\ , & \text{otherwise} \end{cases} ,$$

or in words, a rectangle classifier predicts $+1$ if the instance lies within the rectangle, and -1 if the instance falls outside the rectangle.

(a) What is the *VC dimension of* $\mathcal{F}$. You do not need to justify your answer in this part—you will do so in the remaining parts. [2 marks]

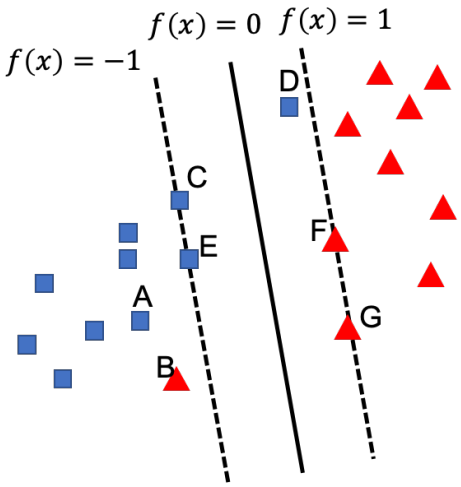
 Continued overleaf . . .

(b) If you answered some number d to Part (a), prove that $VC(\mathcal{F}) \geq d$ in this part by drawing d points that can be shattered by $\mathcal{F}$. Demonstrate how each of the 2^d labellings is possible for these points: provide a new drawing for each labelling consisting of the points and a rectangle overlapping some of the points. [10 marks]

(c) If you answered some number d to Part (a), prove that $VC(\mathcal{F}) < d+1$ in this part by arguing why there can be no set of $d+1$ points that can be labelled in all ways by classifiers in $\mathcal{F}$. [2 marks]

Question 5: Support Vector Machines [12 marks]

Assume that the following figure is an illustration of the *decision boundary* learned by a *linear soft-margin SVM* on a two-class data set, with points from one class labelled with blue squares and points from the other class labelled with red triangles.



The *Lagrangian function* of the *soft-margin SVM* is:

$L(w, b, \lambda, \beta, \xi) = \frac{\|w\|^2}{2} + C\sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \lambda_i g_i(w, b, \xi) + \sum_{i=1}^{n} \beta_i(-\xi_i)$
$g_i(w, b, \xi) = 1 - \xi_i - y^{(i)}(w^T x^{(i)} + b)$

where $f(x) = w^T x + b$, λ_i, β_i are *Lagrange multipliers* and ξ_i is the *slack variable* of the i^{th} data point, C is the *slack penalty*.

Given the seven points A,B C, D, E, F, and G as shown in the figure,

(a) Identify the data point(s) with $\lambda_i = C$ (by index A-G. Note: wrong points in will be deducted marks). [4 marks]

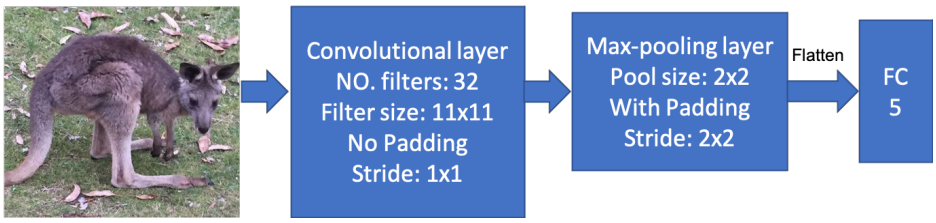
(b) Identify the data point(s) with $\xi_i = 0$ (by index A-G. Note: wrong points in will be deducted marks). [4 marks]

(c) If Point A is removed from the data set, will the decision boundary change? Explain why. [4 marks]

Continued overleaf . . .

Question 6: Convolutional Neural Network [8 marks]

Consider the following simple *Convolutional Neural Network*. The input is a *RGB image (3 channels)* with width and height equal to 224. The setting of the layers are shown in the figure. There are 32 filters in the convolutional layer and each filter has a size of 11×11.



(a) Compute the *number of parameters* in the convolutional layer (show the formula and give the final result) [3 marks]

(b) In order to reduce the number of parameters of the convolutional layer in the figure, while *keeping the receptive field the same*, one possible method is to stack one 3×3 convolutional layer and one 9×9 convolutional layer. Describe five different possible stackings, i.e., different combinations of multiple (can be two or more than two) convolutional layers. (Note: changing the order of the multiple convolutional layers is counted as the same stack, e.g., "Stacking a 9×9 and a 3×3 convolutional layer" is counted as the same stack as "Stacking a 3×3 and a 9×9 convolutional layer".) [5 marks]

Continued overleaf . . .

Section C: Design and Application Questions [20 marks]

In this section you are asked to demonstrate that you have gained a high-level understanding of the methods and algorithms covered in this subject, and can apply that understanding. Answers should be about 1 page in length for the question.

Question 7: Food Delivery Recommendations [20 marks]

Your task is to design an automatic *food delivery recommendation* system for UberEats. (You do not need to be familiar with UberEats to answer this question: UberEats is an app that shows users nearby restaurants, users can choose one shown or search for one, then select a meal to be cooked and delivered to their home.) Because UberEats makes money (a 'commission') whenever a user orders food on its platform, it wants to learn which restaurants to display to the user when the UberEats app is opened. The choice of restaurants displayed on the app front page may make it more likely for a user to order food on the platform, and ultimately how much money UberEats makes—its revenue. UberEats wants to maximise its revenue using multi-armed bandits.

(a) Formulate the *restaurant recommendation task* as a *contextual multi-armed bandit problem*. Explain what your *arms* are, what *rounds* correspond to, what are your *rewards*, and what *context* vectors you would use based on what data UberEats might possess. [5 marks]

(b) Describe what data UberEats should *collect*, so that you could *evaluate* the effectiveness of multiple bandit learners you might try. [5 marks]

(c) It is common for users to browse for a restaurant on the app, but then ultimately not order their food from UberEats. Discuss whether and how MLinUCB (LinUCB with *missing rewards*) from project 2 could be applied in this setting. For example, should all such "browse but no order" events be used? What might be limitations of this approach? [5 marks]

(d) UberEats has another recommendation to make: a budget of coupons (a free meal worth up to $25) every day. UberEats wants to use bandits to *learn which users to give coupons to*. Compare this application to the above one—are there any key differences? [5 marks]

— End of Exam —