UNIVERSITY OF SOUTHAMPTON COMP6208W1

SEMESTER 2 EXAMINATION 2019/20

ADVANCED MACHINE LEARNING

Duration 120 mins (2 hours)

This paper is a WRITE-ON examination paper.

You ***must*** write your Student ID on this Page and must not write your name anywhere on the paper.

All answers should be written within the designated boxes in this examination paper and sufficient space is provided for each question.

If, for some reason, space is required to complete or correct an answer to a question, use the "Additional Space" provided on the facing or adjacent page to the question. Clearly indicate which question the answer corresponds to.

No credit will be given for answers presented elsewhere and without clear indication of to what question they correspond. Blue answer books may be used for scratch; they will be discarded without being looked at.

Answer all questions. Section A (worth 40 marks) are a series of questions with short answers. Section B (worth 60 marks) involve longer questions

Student ID:

Question	Mark	*Arithmetic checked*	*Double Marked*
Total:			

University approved calculators MAY be used.

A foreign language translation dictionary (paper version) is permitted provided it contains no notes, additions or annotations.

10 page examination paper

Section A

Question A 1

(a) Explain why over expressive machines are likely to generalise poorly when the number of training examples is small. *(5 marks)*

Over expressive machine are like to find spurious rules that are consistent with the training data. They typically have a high variance in that the rules they find vary strongly depending on the training set.

(b) Explain why CNNs capture the structure of typical image datasets. *(5 marks)* 5

CNNs are built from local filters that respond to objects in a translationally invariant way (the same feature maps will be activated by an object irrespective of where it is).

(c) Explain the major ways (i) gradient descent (ii) Newton's method and (iii) quasi-Netwon methods differ in terms of the information they use. *(5 marks)* 5

(i) Gradient descent uses just the gradient $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - r\,\boldsymbol{\nabla} f(\boldsymbol{x}^{(t)})$

(ii) Newton's method uses both the gradient and the Hessian $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \mathbf{H}^{-1}\,\boldsymbol{\nabla} f(\boldsymbol{x}^{(t)})$

(iii) Quasi-Newton methods use some method for approximating the Hessian.

5

(d) In stochastic gradient descent (SGD) explain what are mini-batches and their possible advantages and disadvantages. *(5 marks)*

In SGD we compute the gradient of the loss function for a subset of the training examples (the mini-batch). This is far faster than computing the gradient of the loss function for the whole training set, but it is only an approximation to the true gradient.

(e) Describe the Karush-Kuhn-Tucker (KKT) conditions for constrained optimisation. *(5 marks)*

The KKT conditions are used when we have inequality constraints. They state that either the Lagrange multiplier is zero and the point found satisfies the constraint and would be an optimum regardless of the constraint, or the Lagrange multiplier is non-zero and the point found lies on the constraint.

(f) Show that the Dirichlet distribution given by $\mathrm{Dir}(\boldsymbol{p}|\alpha) = \Gamma(\alpha_0) \prod_{i=1}^{d} \frac{p_i^{\alpha_i - 1}}{\Gamma(\alpha_i)}$, where $\boldsymbol{p} = (p_1, p_2, \ldots, p_d)$, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_d)$ and $\alpha_0 = \sum_{i=1}^{d} \alpha_i$ is a conjugate prior to the multinomial likelihood $\mathrm{Binom}(\boldsymbol{k}|n, p) = n! \prod_{i=1}^{d} \frac{p_i^{k_i}}{k_i!}$, where $\boldsymbol{k}$ is a vector of counts $(k_1, k_2, \ldots, k_d)$ with $\sum_{i=1}^{d} k_i = n$. Derive update equations for the parameters $\boldsymbol{\alpha}'$ of the posterior distribution after observing counts $\boldsymbol{k}$. *(5 marks)*

We only need to consider the functional form with respect to p_i. Thus the posterior is proportional to

$$f(\boldsymbol{p}|\boldsymbol{x}) \propto \prod_{i=1}^{d} \frac{p_i^{k_i}}{k_i!} \prod_{i=1}^{d} \frac{p_i^{\alpha_i - 1}}{\Gamma(\alpha_i)} \propto \prod_{i=1}^{d} p_i^{\alpha_i + k_i - 1} \propto \mathrm{Dir}(\boldsymbol{p}|\boldsymbol{\alpha} + \boldsymbol{k})$$

The updated equation is thus $\boldsymbol{\alpha}' = \boldsymbol{\alpha} + \boldsymbol{k}$.

(g) Prove that the set of positive semi-definite matrices is a convex set. *(5 marks)*

For any two PSD matrices $\mathbf{M}_1, \mathbf{M}_2 \succeq 0$ the convex combination $\mathbf{M}_3 = a\,\mathbf{M}_1 + (1-a)\mathbf{M}_2$ (where $a \in [0,1]$) has a quadratic form with an arbitrary vector $\boldsymbol{v}$ of

$$\boldsymbol{v}^\mathsf{T}\mathbf{M}_3\boldsymbol{v} = \boldsymbol{v}^\mathsf{T}\mathbf{M}_1\boldsymbol{v} + \boldsymbol{v}^\mathsf{T}\mathbf{M}_2\boldsymbol{v}$$

But as $\mathbf{M}_1, \mathbf{M}_2 \succeq 0$ then $\boldsymbol{v}^\mathsf{T}\mathbf{M}_1\boldsymbol{v} \geq 0$ and $\boldsymbol{v}^\mathsf{T}\mathbf{M}_2\boldsymbol{v} \geq 0$. Since $a, (1-a) \geq 0$ it follows that $\boldsymbol{v}^\mathsf{T}\mathbf{M}_3\boldsymbol{v} \geq 0$ or $\mathbf{M}_3 \succeq 0$. Thus all convex combinations of positive semi-definite matrices are positive semi-definite so they form a convex set.

5

(h) Explain what are the hyper-parameters of a Gaussian process and why they are relatively easy to learn. *(5 marks)*

The hyper-parameters are the mean function $m(\boldsymbol{x})$ and the kernel or covariance $k(\boldsymbol{x}, \boldsymbol{y})$ (often the kernel will depend on a scale ℓ that is a hyper-parameter). It is relatively straightforward to find good hyper-parameters as we can compute the evidence $\mathbb{P}(\mathcal{D})$ in closed form.

5

End of question A1

(a) ___/5	(b) ___/5	(c) ___/5	(d) ___/5	(e) ___/5	(f) ___/5	(g) ___/5	(h) ___/5	Total ___/40

Section B

Question B 2

(a) If $\{X_i | i = 1,\, 2,\, m\}$ is a set of correlated random variables such that

$$\langle X_i \rangle = \mu \qquad \langle (X_i - \mu)(X_i - j) \rangle = \begin{cases} \sigma^2 & \text{if } i = j \\ \rho\,\sigma^2 & \text{if } i \neq j \end{cases}$$

show

$$\left\langle \left(\frac{1}{n} \sum_{i=1}^{n} X_i - \mu \right)^2 \right\rangle = \rho\,\sigma^2 + \frac{(1-\rho)\,\sigma^2}{n}$$

(10 marks)

$$\begin{aligned}
\left\langle \left(\frac{1}{n} \sum_{i=1}^{n} X_i - \mu \right)^2 \right\rangle &= \frac{1}{n^2} \left\langle \left(\sum_{i=1}^{n} (X_i - \mu) \right)^2 \right\rangle \\
&= \frac{1}{n^2} \left\langle \sum_{i,j=1}^{n} (X_i - \mu)\,(X_j - \mu) \right\rangle = \frac{1}{n^2} \sum_{i,j=1}^{n} \langle (X_i - \mu)\,(X_j - \mu) \rangle \\
&= \frac{1}{n^2} \sum_{i=1}^{n} \sigma^2 + \frac{1}{n^2} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \rho\,\sigma^2 = \frac{1}{n^2} n\,\sigma^2 + \frac{1}{n^2} n\,(n-1)\,\rho\,\sigma^2 \\
&= \rho\,\sigma^2 + \frac{(1-\rho)}{n} \sigma^2
\end{aligned}$$

$\overline{10}$

(b) Using the result derived in part (a) explain why ensembling many machines can reduce the variance in the bias-variance dilemma if the machine predictions are not heavily correlated. Use this to explain the success of random forest. *(10 marks)*

We can think of X_i as the prediction made by a learning machine. For a finite training set this prediction will vary. If our machines are unbiased $\langle X \rangle_i = \mu$, that is the true output. ρ is a measure of the Pearson correlation between the predictions. By ensembling many machines (high n) we reduce the second term. However, because the machines are correlated our prediction will still vary from the true prediction by $\rho\,\sigma^2$. In random forest we try to make the predictions uncorrelated by basing them on a different set of features and using bootstrapping so they learn on a slightly different training set.

End of question B2

$\frac{}{10}$

(a) $\frac{}{10}$ (b) $\frac{}{10}$ Total $\frac{}{20}$

Question B 3

(a) We can write the loss for ridge regression as

$$L(\boldsymbol{w}) = \|\mathbf{X}\,\boldsymbol{w} - \boldsymbol{y}\|^2 + \eta\,\|\boldsymbol{w}\|^2$$

where $\mathbf{X}$ is the design matrix and $\boldsymbol{y}$ is a vector of target values. Calculate the weights, $\boldsymbol{w}^*$, that minimise the loss. By writing $\mathbf{X} = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^\mathsf{T}$ show that we can write $\boldsymbol{w}^* = \mathbf{V}\,\hat{\mathbf{S}}^+\mathbf{U}^\mathsf{T}\boldsymbol{y}$, where the elements of $\hat{\mathbf{S}}^+$ are zero everywhere except for the diagonal where $\hat{S}^+_{ii} = s_i/(s_i^2 + \eta)$ with s_i being the singular values of $\mathbf{X}$ (i.e. $s_i = S_{ii}$). *(15 marks)*

The gradient of the loss is given by

$$\nabla L(\boldsymbol{w}) = 2\,\mathbf{X}^\mathsf{T}\mathbf{X}\,\boldsymbol{w} - 2\,\mathbf{X}^\mathsf{T}\boldsymbol{y} + 2\,\eta\,\boldsymbol{w}.$$

Setting this to zero and solving for $\boldsymbol{w}$

$$\boldsymbol{w}^* = (\mathbf{X}^\mathsf{T}\mathbf{X} + \eta\,\mathbf{I})^{-1}\,\mathbf{X}^\mathsf{T}\,\boldsymbol{y}.$$

Writing $\mathbf{X} = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^\mathsf{T}$ then

$$\mathbf{X}^\mathsf{T}\mathbf{X} + \eta\mathbf{I} = \mathbf{V}\,\left(\mathbf{S}^\mathsf{T}\mathbf{S} + \eta\mathbf{I}\right)\mathbf{V}^\mathsf{T}.$$

So that

$$\boldsymbol{w}^* = \mathbf{V}\,\left(\mathbf{S}^\mathsf{T}\mathbf{S} + \eta\,\mathbf{I}\right)^{-1}\mathbf{S}\mathbf{U}^\mathsf{T}\,\boldsymbol{y}$$

(note that we used the fact that V and U are orthogonal matrices so that $\mathbf{V}\,\mathbf{V}^\mathsf{T} = \mathbf{I}$ and $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}$).

Thus, $\boldsymbol{w}^* = \mathbf{V}\,\hat{\mathbf{S}}^+\mathbf{U}^\mathsf{T}\,\boldsymbol{y}$ where

$$\hat{\mathbf{S}}^+ = \left(\mathbf{S}^\mathsf{T}\mathbf{S} + \eta\,\mathbf{I}\right)^{-1}\mathbf{S}.$$

Since the elements of S are zero everywhere except on the diagonal then $\mathbf{S}^\mathsf{T}\mathbf{S}$ will also be zero everywhere except for the diagonal which are s_i^2 and consequently $\hat{\mathbf{S}}^+$ will be zero everywhere except on the diagonal where $\hat{S}^+_{ii} = s_i/(s_i^2 + \eta)$.

$\overline{15}$

(b) Use the result that you derive to explain how adding the L_2 regulariser $\eta \|\boldsymbol{w}\|^2$ improves the conditioning of the solution and is likely to improve generalisation.
(5 marks)

The effect of adding a regulariser is to change the usual pseudo-inverse ($S_{ii}^{+} = s_i^{-1}$) to $\hat{S}_{ii}^{+} = s_i/(s_i^2 + \eta)$. If for any singular value $s_i = 0$ the problem would be ill-posed as there would be an infinity of solutions—the pseudo-inverse is not defined. The problem now has a unique solution. That is, $\hat{S}_{ii}^{+} = 0$ if $s_i = 0$. Similarly is s_i is small $S_{ii}^{+} = s_i^{-1}$ would be very large and the pseudo-inverse poorly conditions (very sensitive to the training data—i.e. with a large variance). However, if $s_i < \eta$ then $\hat{S}_{ii}^{+} < 1$ and the regularised solution will be much better conditioned.

End of question B3

5

(a) ___/15 (b) ___/5 Total ___/20

Question B 4

(a) Write a Lagrangian for the linear programming problem of choosing $\boldsymbol{x}$ to minimise $\boldsymbol{c}^{\mathsf{T}}\,\boldsymbol{x}$, subject to the constraints $\mathbf{M}\,\boldsymbol{x} = \boldsymbol{b}$. Show that the Lagrangian can be rewritten to obtain the dual problem where the roles of the variables $\boldsymbol{x}$ and the Lagrange multipliers are exchanged. Write down the dual problem as a maximisation problem plus a new set of constraints. *(5 marks)*

The Lagrangian will be

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}) = \boldsymbol{c}^{\mathsf{T}}\,\boldsymbol{x} - \boldsymbol{\alpha}^{\mathsf{T}}\,(\mathbf{M}\,\boldsymbol{x} - \boldsymbol{b})$$

where α are a set of Lagrange multipliers. Note that the optimum is given by minimising with respect to $\boldsymbol{x}$, but maximising with respect to α. We can rewrite this as

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}) = \boldsymbol{b}^{\mathsf{T}}\boldsymbol{\alpha} - \boldsymbol{x}^{\mathsf{T}}\left(\mathbf{M}^{\mathsf{T}}\boldsymbol{\alpha} - \boldsymbol{c}\right)$$

which we can interpret as a linear programming problem where we choose α to maximise $\boldsymbol{b}^{\mathsf{T}}\boldsymbol{\alpha}$ subject to the constraints $\mathbf{M}^{\mathsf{T}}\boldsymbol{\alpha} = \boldsymbol{c}$.

5

(b) Describe the Wasserstein distance as a linear programming problem and describe the dual problem. Describe how this is used in the Wasserstein GAN. *(15 marks)*

(Not all details are necessary for full marks.)

We can pose the Wasserstein distance between two probability densities $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ as a minimisation problem were

$$W(p,q) = \min_{\gamma(\boldsymbol{x},\boldsymbol{y})\in\Lambda} \int \gamma(\boldsymbol{x},\boldsymbol{y})\, d(\boldsymbol{x},\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y}$$

where $d(\boldsymbol{x},\boldsymbol{y})$ is the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ and Λ is the set of joint probability distributions with marginal given by

$$\int \gamma(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{y} = p(\boldsymbol{x}), \qquad \int \gamma(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{x} = q(\boldsymbol{y}).$$

This can be seen as a linear programme of finding $\gamma(\boldsymbol{x},\boldsymbol{y})$ that minimises a linear objective function (the integral in the definition of $W(p,q)$), subject to constraints (given above). Note that $\gamma(\boldsymbol{x},\boldsymbol{y})$ can be consider a transportation policy transferring probability mass from p at a point $\boldsymbol{x}$ to q at a point $\boldsymbol{y}$.

The Lagrangian can be written as

$$\mathcal{L} = \int \gamma(\boldsymbol{x},\boldsymbol{y})\, d(\boldsymbol{x},\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} - \int \alpha(\boldsymbol{x}) \left(\int \gamma(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{y} - p(\boldsymbol{x}) \right) \mathrm{d}\boldsymbol{y} - \int \beta(\boldsymbol{y}) \left(\int \gamma(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{x} - g(\boldsymbol{x}) \right) \mathrm{d}\boldsymbol{x}$$

$\boxed{15}$

where $\alpha(\boldsymbol{x})$ and $\beta(\boldsymbol{y})$ are Lagrange multipliers and $\gamma(\boldsymbol{x},\boldsymbol{y}) \geq 0$. This can also be written

$$\mathcal{L} = \int \alpha(\boldsymbol{x})\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} + \int \beta(\boldsymbol{x})\, g(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} + \int \gamma(\boldsymbol{x},\boldsymbol{y})\ (d(\boldsymbol{x},\boldsymbol{y}) - \alpha(\boldsymbol{x}) - \beta(\boldsymbol{y}))\ \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y}$$

Leading to a dual problem of finding $\alpha(\boldsymbol{x})$ and $\beta(\boldsymbol{y})$ that maximises $\int \alpha(\boldsymbol{x})\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} + \int \beta(\boldsymbol{x})\, g(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$ subject to the constraint $\alpha(\boldsymbol{x}) + \beta(\boldsymbol{y}) \leq d(\boldsymbol{x},\boldsymbol{y})$. Since $d(\boldsymbol{x},\boldsymbol{x}) = 0$ we have that $\beta(\boldsymbol{x}) = -\alpha(\boldsymbol{x})$. Thus we have to find a function $\alpha(\boldsymbol{x})$ that maximises $\int \alpha(\boldsymbol{x})\ \big(p(\boldsymbol{x}) - g(\boldsymbol{x})\big)\ \mathrm{d}\boldsymbol{x}$ subject to the constraint $\alpha(\boldsymbol{x}) - \alpha(\boldsymbol{y}) \leq d(\boldsymbol{x},\boldsymbol{y})$, that is, $\alpha(\boldsymbol{x})$ is Lipschitz-1.

In a Wasserstein GAN we learn a function $\alpha(\boldsymbol{x})$ that discriminate between samples, $\boldsymbol{x}$, drawn from the true distribution $p(\boldsymbol{x})$ and samples, $\boldsymbol{y}$, drawn from a generator with distribution $g(\boldsymbol{y})$. It does this by maximising $\int \alpha(\boldsymbol{x})\ \big(p(\boldsymbol{x}) - g(\boldsymbol{x})\big)\ \mathrm{d}\boldsymbol{x}$. This is our critic or discriminator network, but it should be Lipschitz-1. We also learn a generator that tries to minimise this objective function (decreasing the Wasserstein distance).

End of question B4

(a) $\frac{}{5}$	(b) $\frac{}{15}$	Total $\frac{}{20}$

END OF PAPER