Stanford CS 229, Fall 2022 Midterm Solutions

The midterm is **open-book** (**no internet**), **closed-collaboration**, and subject to the Honor Code. Please write your name on the first page and also on **every** midterm page.

You may:

- Access any materials or resources, including the course notes and reference material you may have downloaded or printed previously (remember to cite your sources). You can use electronic devices, but cannot connect to the internet.
- Cite without proof any result from lectures or lecture notes, unless otherwise stated.
- Use the back of the exam pages for scratch work. Please make sure all of your answers are clearly written on the front of the pages.

You may not:

- Talk to, consult, or collaborate with anyone about the exam, and you may not consult any human or artificial intelligence about the exam problems. Any such collaboration is a violation of the Honor Code.
- Access any resources from the internet during the midterm. Note: you are allowed to download materials from the internet ahead of the midterm and access them offline during the midterm.

Good luck! We know you've been working hard, and we all want you to succeed! :)

Question	Points
1 True or False	/15
2 Multiple Choice	/15
3 Dirichlet Distribution as Exponential Family	/25
4 Kernel Method for Transfer Learning	/25
5 Recurrent Neural Network	/20
6 Generalized Least-Squares Regression	/20 + 10
Total	/120 + 10

Name of Student: _____

SUNetID: _____@stanford.edu

The Stanford University Honor Code:

I attest that I have not given or received aid in this examination, and that I have done my share and taken an active part in seeing to it that others as well as myself uphold the spirit and letter of the Honor Code.

Signed: _____

1. **[15 points] True or False**

   For each question, answer True or False, and **briefly and convincingly justify your answer** in a sentence or two.

   (a) [3 points] Any probability distribution over continuous random variables belongs to the exponential family.

   Answer:

   False. Examples of common distributions that are not exponential families are Student's t, most mixture distributions, and even the uniform distribution.

   (b) [3 points] When using kernel methods, during test time, we still need to access the training data if we want to make a prediction on a new example x.

   Answer:

   Yes. We need to access the training data since the kernel functions are used to measure the similarity between the training data points and the new data point.

   (c) [3 points] In least squares regression, given a dataset $(X, \boldsymbol{Y})$, where $X \in \mathrm{R}^{n \times d}$ is the feature matrix with each row representing a data instance, and $\boldsymbol{Y} \in \mathrm{R}^n$ is the target vector, we would like to minimize the cost function: $J(\boldsymbol{\theta}) = \frac{1}{2}(X\boldsymbol{\theta} - \boldsymbol{Y})^\top (X\boldsymbol{\theta} - \boldsymbol{Y})$. As long as the rows in the feature matrix X are linearly independent, we can always get the closed-form optimal solution as: $\boldsymbol{\theta} = (X^\top X)^{-1} X^\top \boldsymbol{Y}$.

   Answer:

   False. When the number of examples is fewer than the number of features, $X^\top X$ is also not invertible.

All equivalent statements (using rank, full-rank, etc) will be accepted.

(d) [3 points] Suppose we train a neural network model A and a linear model B on the same classification problem (e.g. the MNIST dataset). If model A has a lower validation error than model B, then model A is always better than model B in terms of test error.

Answer:

False. A lower validation error does not always mean a lower test error because of randomness caused by finite samples.

(e) [3 points] A multi-layer fully-connected neural network with identity activation functions ($\sigma(x) = x$) can perfectly classify a non-linearly separable dataset.

Answer:

False. Multi-layer neural networks with identity activation functions will be equivalent to a linear function, which cannot classify a non-linearly separable dataset.

2. **[15 points] Multiple Choices**

For each question, select ALL of the correct answers. There may be multiple correct answers for each problem. If you select all the correct answers, you get full points. If all your selected answers are correct but you do not select all the correct answers, you get half points. If at least one of your selection is incorrect, you get zero points.

(a) [3 points] Based on the following descriptions, which of the following is(are) hyper-parameters? (Parameters are the variables in your model that are learned and updated by the gradient. Hyper-parameters are variables that are used to control the learning process.)

(1) The learning rate for the stochastic gradient decent algorithm.

(2) The weight W in the linear regression algorithm assuming that the linear regression model is $Wx + b$.

(3) The number of layers of a neural network

(4) The number of hidden units in a given layer of a neural network

(5) The mean vector μ for a Gaussian Discriminant Analysis model

Answer: (1), (3), (4). Reason: (2), (5) are parameters

(b) [3 points] Which of the following statement(s) is(are) true for neural networks?

(1) Neural networks with more layers always fit the training data with lower loss than networks with less layers

(2) A neural network with only fully-connected layers can accept inputs with different dimensions, e.g. can accept both $x_1 \in \mathbb{R}^{d_1}$ and $x_2 \in \mathbb{R}^{d_2}$, where $(d_1 \neq d_2)$.

(3) $f(x) = \begin{cases} 0.1x & x \leq 0 \\ x & x > 0 \end{cases}$ is a nonlinear function.

(4) Back-propagation is not a method to compute the gradient for neural network parameters.

Answer: (3). Reason: (1) A neural network with two layers and a nonlinear function can fit any loss. (2) The input dimension to a fully-connected neural network is fixed. (4) Neural network uses back-propagation to compute the gradient.

(c) [3 points] When we split the dataset into training, validation and test sets, which of the following statement(s) is(are) true?

(1) For a **standard** machine learning problem, training data and validation data are drawn from the same distribution.

(2) Since validation data is also used in the training phase of machine learning, model parameters will overfit to validation data similar to training data.

(3) k-fold cross-validation requires training k models to select hyper-parameters.

(4) The performance of a model always follows this order: performance on training data > validation data > testing data.

Answer: (1)(3). Reason: (2) Validation data are only used to select hyper-parameters but not train parameters. (4) The training performance is usually higher than validation and test performance but the order between validation and test performance cannot be decided.

(d) [3 points] Which of the following statement(s) is(are) true for the kernel?

(1) A valid kernel $k(x, z)$ should satisfy $k(x, z) > 0$

(2) For two valid positive semi-definite kernels $k_1(x, z)$, $k_2(x, z)$, their product $k_1(x, z)k_2(x, z)$ is also a valid kernel.

(3) A valid kernel $k(x, z)$ can always be decomposed into a finite number of inner-products of ϕ, i.e. $k(x, z) = \sum_{i=1}^{n} \phi_i(x)^T \phi_i(z), (n < \infty)$.

(4) Kernel methods can fit a non-linear function.

Answer: (2)(4). Reason: (1) $k(x, z) = x^T z$ is also a valid kernel but sometimes $x^T z < 0$; (2) is the due to the composition rule of kernels. Product of two PSD matrices is PSD. (3) Some kernel functions can be decomposed into infinite number of inner-products of ϕ.

(e) [3 points] Say we have a task to predict the depth of a river, given the climate data from last month. Choose all the model(s) that is(are) suitable for this problem.

(1) Linear Regression

(2) Logistic Regression

(3) Naive Bayes Classifier

(4) Neural Network with Regression Loss

Answer: (1) (4). Reason: (2)(3) are classification models.

3. **[25 points] Dirichlet Distribution as Exponential Family**

In this problem, we will explore the properties of the Dirichlet distribution, which is also a member of the exponential family and a very useful distribution in Bayesian statistics[1].

For $\boldsymbol{y} = (y_1, \ldots, y_k)$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$, Dirichlet distribution $\boldsymbol{y} \sim \text{Dir}(\alpha_1, \ldots, \alpha_k)$ is defined to have the probability density function:

$$p(\boldsymbol{y}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{k} y_i^{\alpha_i - 1} \quad \text{with} \quad B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{k} \alpha_i)}$$

Although $B(\boldsymbol{\alpha})$ seems complicated, it is just the normalizing factor, and the gamma function is defined as $\Gamma(n) = n!$ for any positive integer n. Putting this all together,

$$p(y_1, \ldots, y_k|\alpha_1, \ldots, \alpha_k) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} y_i^{\alpha_i - 1} \tag{1}$$

(a) [8 points] Recall that the general exponential family distribution parametrized by the natural parameter η has probability density function (or probability mass function if discrete):

$$p(\boldsymbol{y}; \boldsymbol{\eta}) = b(\boldsymbol{y}) \exp\{\boldsymbol{\eta}^\top T(\boldsymbol{y}) - a(\boldsymbol{\eta})\},$$

where $b(\boldsymbol{y})$ is called the base measure, $T(\boldsymbol{y})$ is the sufficient statistic, and $a(\boldsymbol{\eta})$ is the log-partition function. Using Eqn. (1) above, show that the Dirichlet distribution is a member of the exponential family. Clearly state the values for $b(\boldsymbol{y})$, $\boldsymbol{\eta}$, $T(\boldsymbol{y})$, $a(\boldsymbol{\eta})$.

Answer:

Solution 1:

$$b(\boldsymbol{y}) = 1$$

$$\boldsymbol{\eta} = \boldsymbol{\alpha} - 1 = \begin{bmatrix} \alpha_1 - 1 \\ \alpha_2 - 1 \\ \vdots \\ \alpha_k - 1 \end{bmatrix}$$

$$T(\boldsymbol{y}) = \log \boldsymbol{y} = \begin{bmatrix} \log y_1 \\ \log y_2 \\ \vdots \\ \log y_k \end{bmatrix}$$

$$a(\boldsymbol{\eta}) = \sum_{i=1}^{k} \log \Gamma(\alpha_i) - \log \Gamma\left(\sum_{i=1}^{k} \alpha_i\right)$$

[1] In probability and statistics, the Dirichlet distribution is often defined over the probability simplex (the space of all probability distributions over k mutually exclusive events) $\Delta_k = \{\boldsymbol{y} \mid \boldsymbol{y} \in \mathbb{R}^k,\ \boldsymbol{y}^\top \mathbf{1} = 1,\ \boldsymbol{y} \geq \mathbf{0}\}$.

Solution 2:

$$b(\boldsymbol{y}) = \frac{1}{\prod_{i=1}^{k} y_i}$$

$$\boldsymbol{\eta} = \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix}$$

$$T(\boldsymbol{y}) = \log \boldsymbol{y} = \begin{bmatrix} \log y_1 \\ \log y_2 \\ \vdots \\ \log y_k \end{bmatrix}$$

$$a(\boldsymbol{\eta}) = \sum_{i=1}^{k} \log \Gamma(\alpha_i) - \log \Gamma \left(\sum_{i=1}^{k} \alpha_i \right)$$

Continue answer 3(a) here.

(b) [9 points] Given a dataset $\mathcal{D} = \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}\}$, we can use maximum likelihood estimation to estimate the parameters of a Dirichlet distribution. Let the log-likelihood of the dataset under $\text{Dir}(\boldsymbol{\alpha})$ be $\log p(\mathcal{D}|\boldsymbol{\alpha}) = \log\left(\prod_{i=1}^{N} p(\boldsymbol{y}^{(i)}|\boldsymbol{\alpha})\right)$. By taking the derivative of the log-likelihood with respect to α_j, derive the gradient ascent update rule for estimating the parameters of a Dirichlet distribution.

For this question, we assume that we know how to compute the **Digamma** function (the logarithm derivative of the Gamma function): $\psi(x) = \frac{\mathrm{d}}{\mathrm{d}x}\log\Gamma(x)$.

Answer:

$$\begin{aligned}
\ell(\boldsymbol{\alpha}) &= \log\left(\prod_{i=1}^{N} p(\boldsymbol{y}^{(i)}|\boldsymbol{\alpha})\right) \\
&= N\left(\log\Gamma\left(\sum_{j=1}^{k}\alpha_j\right) - \sum_{j=1}^{k}\log\Gamma(\alpha_j)\right) + \sum_{i=1}^{N}\sum_{j=1}^{k}(\alpha_j - 1)\log y_j^{(i)} \\
&= N\left(\log\Gamma\left(\sum_{j=1}^{k}\alpha_j\right) - \sum_{j=1}^{k}\log\Gamma(\alpha_j)\right) + \sum_{j=1}^{k}(\alpha_j - 1)\left(\sum_{i=1}^{N}\log y_j^{(i)}\right)
\end{aligned}$$

Using the Digamma function, we can compute the gradient:

$$(\nabla\ell(\boldsymbol{\alpha}))_j = \frac{\partial\ell(\boldsymbol{\alpha})}{\partial\alpha_j} = N\left(\psi\left(\sum_{j=1}^{k}\alpha_j\right) - \psi(\alpha_j)\right) + \sum_{i=1}^{N}\log y_j^i$$

The gradient ascent update rule with learning rate β will be:

$$\boldsymbol{\alpha} := \boldsymbol{\alpha} + \beta \cdot \nabla\ell(\boldsymbol{\alpha})$$

Continue answer 3(b) here.

(c) [8 points] In Bayesian statistics, the Dirichlet distribution is typically used as the prior distribution for Multinomial distribution. The probability mass function of a Multinomial distribution parametrized by $\boldsymbol{y}$ is defined as:

$$p(x_1, \ldots, x_k | y_1, \ldots, y_k) = \frac{(\sum_{i=1}^k x_i)!}{\prod_{i=1}^k x_i!} \prod_{i=1}^k y_i^{x_i}$$

Suppose the prior distribution of $\boldsymbol{y}$ (the parameter of the Multinomial distribution) is $\text{Dir}(\boldsymbol{y}|\boldsymbol{\alpha})$ and the probability of observing a sample $\boldsymbol{x} \in \text{N}^k$ (the k-tuple representing the counts of each event $i \in \{1, \ldots, k\}$) is $\text{Multinomial}(\boldsymbol{x}|\boldsymbol{y})$. Use Bayes' Rule to show that the posterior distribution of $\boldsymbol{y}$ after observing a sample $\boldsymbol{x}$, $p(\boldsymbol{y}|\boldsymbol{x})$, is also a Dirichlet distribution and state the parameters for the posterior Dirichlet distribution.

Hint: You may use the following fact. If a distribution defined over some random variables $(z_1, \ldots, z_k)$ satisfies that $p(z_1, \ldots, z_k)$ is proportional to $\prod_{i=1}^k z_i^{\alpha_i - 1}$, then $p(z_1, \ldots, z_k)$ must be a Dirichlet distribution with $\alpha_1, \ldots, \alpha_k$ being the associated parameters.

Answer:

Note that when deriving $p(\boldsymbol{y}|\boldsymbol{x})$, $\boldsymbol{x}$ and $\boldsymbol{\alpha}$ can be treated as constants. Based on Bayes' Rule, we know that

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x}) &\propto p(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{y}) \\ &\propto \text{Dir}(\boldsymbol{y}|\boldsymbol{\alpha}) \cdot \text{Multinomial}(\boldsymbol{x}|\boldsymbol{y}) \\ &\propto \prod_{i=1}^k y_i^{\alpha_i + x_i - 1} \end{aligned}$$

Therefore, $p(\boldsymbol{y}|\boldsymbol{x}) = \text{Dir}(\boldsymbol{y}|x_1 + \alpha_1, \ldots, x_k + \alpha_k)$.

4. **[20 points] Recurrent Neural Network**

A recurrent neural network (RNN) is a widely-used neural network architecture to perform prediction tasks on sequences.

Consider a sequence prediction problem. We have a training dataset $\{(x_1, x_2, x_3, ..., x_T)\}$ containing a complete sequence with length T. Here, $x_t \in \mathbb{R}^d$ at each time step t is a d-dimensional column vector. We also have a test dataset with $\{(x_1)\}$ with an incomplete sequence, where only the first element of the sequence is given. The goal is to predict the whole sequence with length T. We use a simple RNN architecture to do the task. Note that for this problem, we assume there is only one sequence in the train set and one in the test set dataset.

An RNN unit, at each time step t, does the following computation:

$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b) \tag{2}$$

where

- $W_x \in \mathbb{R}^{n \times d}$, $W_h \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ are all model parameters. Note that here the parameters do not change with respect to time step t.
- $h_t \in \mathbb{R}^n$ is an n-dimensional vector, and called the hidden state. You may regard h_0 as a constant vector for convenience.
- σ is the sigmoid function and applied element wise.

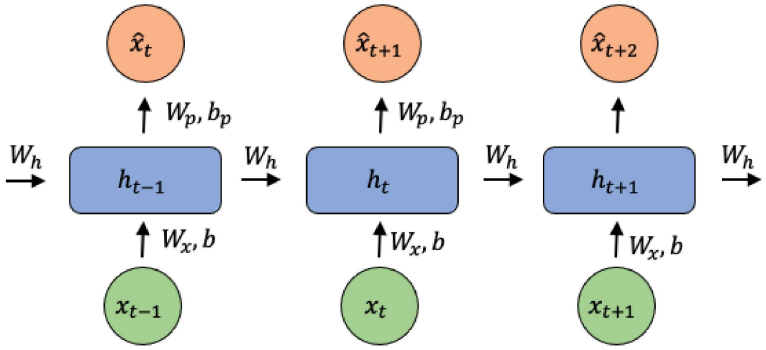

Figure 1: Diagram of RNN

At each time step t, we predict the next time step state x_{t+1} using only a fully-connected neural network, with the hidden state h_t as the input: $\hat{x}_{t+1} = W_p h_t + b_p$, where $W_p \in \mathbb{R}^{d \times n}, b_p \in \mathbb{R}^d$ are parameters. We use the L2 loss as the loss function between the predicted output $\hat{x}_{t+1}$ and the true next state x_{t+1}: $l(\hat{x}_{t+1}, x_{t+1}) = \|x_{t+1} - \hat{x}_{t+1}\|_2^2$.

Question (b) depends on (a) but (c) is independent from (a)(b).

(a) [6 points] Let's first derive $\frac{\partial h_t}{\partial h_{t-1}}$, when given a sample $x_1, ...x_T$.

Answer:

$$\frac{\partial h_t}{\partial h_{t-1}} = \left((1 - \sigma(W_x x_t + W_h h_{t-1} + b)) \odot \sigma(W_x x_t + W_h h_{t-1} + b)\right)^T W_h$$

(b) [8 points] With the data in problem (a) and assuming that $T = 3$, derive the gradient of the total loss $l(x_2, \hat{x}_2) + l(x_3, \hat{x}_3)$ with respect to the parameter W_x of the RNN unit.

Hint: W_x is used in multiple time steps. Also, in your results, you can use any intermediate results including $\hat{x}_2, \hat{x}_3, h_2, h_1$ or intermediate gradients like $\frac{\partial h_2}{\partial h_1}$.

Answer:

$$
\begin{aligned}
&\frac{\partial l(x_2, \hat{x}_2) + l(x_3, \hat{x}_3)}{\partial h_2} = 2W_p^T(\hat{x}_3 - x_3) \\
&\frac{\partial l(x_2, \hat{x}_2) + l(x_3, \hat{x}_3)}{\partial h_1} = 2W_p^T(\hat{x}_2 - x_2) + \left(2W_p^T(\hat{x}_3 - x_3)\right)^T \frac{\partial h_2}{\partial h_1} \\
&\frac{\partial h_2}{\partial W_x} = \left((1 - \sigma(W_x x_2 + W_h h_1 + b)) \odot \sigma(W_x x_2 + W_h h_1 + b)\right) x_2^T \\
&+ \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W_x} \\
&\frac{\partial h_1}{\partial W_x} = \left((1 - \sigma(W_x x_1 + W_h h_0 + b)) \odot \sigma(W_x x_1 + W_h h_0 + b)\right) x_1^T \\
&\frac{\partial l(x_2, \hat{x}_2) + l(x_3, \hat{x}_3)}{\partial W_x} = \left(\frac{\partial l(x_2, \hat{x}_2) + l(x_3, \hat{x}_3)}{\partial h_2}\right)^T \frac{\partial h_2}{\partial W_x} \\
&+ \left(\frac{\partial l(x_2, \hat{x}_2) + l(x_3, \hat{x}_3)}{\partial h_1}\right)^T \frac{\partial h_1}{\partial W_x}
\end{aligned}
\tag{3}
$$

Continue answer 4(b) here.

(c) [6 points] Now, assuming we have trained all parameters of the RNN and $T = 3$, derive the predictions for the rest of the sequence when given test data x_1.

Answer:

$$\begin{aligned} h_1 &= \sigma\left(W_x x_1 + W_h h_0 + b\right) \\ \hat{x}_2 &= W_p h_1 + b_p \\ h_2 &= \sigma\left(W_x x_2 + W_h h_1 + b\right) \\ \hat{x}_3 &= W_p h_2 + b_p \end{aligned} \tag{4}$$

The prediction is $(x_1, \hat{x}_2, \hat{x}_3)$.

5. **[25 points] Kernel Method for Transfer Learning**

Transfer learning is one of the widely used machine learning paradigms. Here, we consider a problem setting with 2 domains. We aim to transfer the learned model from one domain to another domain, where the two domains have different distributions of data. In this paradigm, a data point could be typically represented by (x, y, d), where $x \in \mathbb{R}^d$ is the input data and y is the label (note that y is not used in this question) and d is the domain label, which is $d = \begin{bmatrix}1\\0\end{bmatrix}$ if the data belongs to the first domain and $d = \begin{bmatrix}0\\1\end{bmatrix}$ if the data belongs to the second domain.

Now, an important step is to design a kernel to build the relationships between different data points. In this question, we will gradually build such a kernel.

All the three sub-questions are independent, you can solve the questions in any order.

(a) [10 points] First, let's build a kernel for d, which represents the relationship of data within a domain and between domains. For a dataset $\{(x_i, d_i)\}$, with $i \in 1, ..., n$ and $d_i = \begin{bmatrix}1\\0\end{bmatrix}$ or $\begin{bmatrix}0\\1\end{bmatrix}$, consider the following function:

$$k_d(d_i, d_j) = \begin{cases} \lambda & \text{if } d_i \neq d_j \\ 1 & \text{if } d_i = d_j \end{cases} \tag{5}$$

where $\lambda \in \mathbb{R}$. This indicates that the relationship coefficient of data within the same domain is 1 and between different domains is λ. Now, let's prove that only when $-1 \leq \lambda \leq 1$ will K_d, the corresponding kernel matrix, be **positive semi-definite**. i.e., for any $z \in \mathbb{R}^n$, $h(z) = z^T(K_d)z \geq 0$ only when $-1 \leq \lambda \leq 1$.

(i) [6 points] We define a function δ on the dataset, and functions, $f(z)$ and $g(z)$ on any $z \in \mathbb{R}^n$ as:

$$\delta(d_i, d_j) = \begin{cases} 1 & \text{if } d_i = d_j \\ -1 & \text{if } d_i \neq d_j \end{cases}$$

$$f(z) = \left(z_1 + \sum_{i=2}^{n} \delta(d_1, d_i) z_i \right)^2$$

$$g(z) = \left(\sum_{i=1}^{n} z_i \right)^2$$

where z_i is the i-th entry of the z vector. Prove that $\forall\ z \in \mathbb{R}^n$, either $f(z) \leq h(z) \leq g(z)$ or $f(z) \geq h(z) \geq g(z)$ when $-1 \leq \lambda \leq 1$. Hence, prove that k_d is a valid kernel when $-1 \leq \lambda \leq 1$.

(ii) [4 points] Prove that when $\lambda > 1$ or $\lambda < -1$, the kernel matrix K_d is not positive semi-definite. **Hint**: Think counter-examples!

Answer: (i) Let z_i be the i-th entry of the z vector, we define

$$\begin{aligned} f(z) &= \left(z_1 + \sum_{i=2}^{n} \delta(d_1, d_i) z_i \right)^2 \\ &= \sum_{i=1}^{n} z_i^2 + \sum_{1 \le i < j \le n} 2\delta(d_1, d_i)\delta(d_1, d_j) z_i z_j \\ g(z) &= \left(\sum_{i=1}^{n} z_i \right)^2 \\ &= \sum_{i=1}^{n} z_i^2 + \sum_{1 \le i < j \le n} 2 z_i z_j \end{aligned} \tag{6}$$

Since $k_d(d_i, d_j) = k_d(d_j, d_i)$, we have

$$h(z) = z^T K z = \sum_{i=1}^{n} z_i^2 + \sum_{1 \le i < j \le n} 2k_d(d_i, d_j) z_i z_j \tag{7}$$

First, all the functions has the term $\sum_{i=1}^{n} z_i^2$.

For all the $d_i = d_j$, we have $\delta(d_1, d_i) = 1, \delta(d_1, d_j) = 1$ or $\delta(d_1, d_i) = -1, \delta(d_1, d_j) = -1$. So we have $\delta(d_1, d_i)\delta(d_1, d_j) = 1$. Also $k_d(d_i, d_j) = 1$. Thus, $2z_i z_j$ exists in $f(z), h(z), g(z)$.

The rest of terms are $z_i z_j$, $d_i \neq d_j$, we have $\delta(d_1, d_i) = 1, \delta(d_1, d_j) = -1$ or $\delta(d_1, d_i) = -1, \delta(d_1, d_j) = 1$. So we have $\delta(d_1, d_i)\delta(d_1, d_j) = -1$. Also $k_d(d_i, d_j) = \lambda \in [-1, 1]$. Thus, either $-2z_i z_j \le 2\lambda z_i z_j \le 2z_i z_j$ or $-2z_i z_j \ge 2\lambda z_i z_j \ge 2z_i z_j$ is satisfied. So $h(z)$ falls between $f(z)$ and $g(z)$ and $f(z) \ge 0, g(z) \ge 0$. So we have $h(z) \ge 0$.

(ii) $\lambda > 1$ **or** $\lambda < -1$ **makes the kernel not positive semi-definite.**

We can have n data points with $d_1 \neq d_i (i \neq 1)$ and $d_i = d_j (i \neq 1, j \neq 1)$. Then $k_d(d_1, d_i) = k_d(d_i, d_1) = \lambda (i \neq 1)$, $k_d(d_i, d_j) = 1 (i \neq 1, j \neq 1)$ and $k(d_1, d_1) = 1$.

In the first case, we let z is a n-dimensional vector with the first and the second element to be 1 while other elements to be 0, then

$$z^T K z = 2 + 2\lambda \tag{8}$$

In the second case, we let z is a n-dimensional vector with the first element to be 1 and the second element to be -1 while other elements to be 0, then

$$z^T K z = 2 - 2\lambda \tag{9}$$

If $\lambda > 1$, in the second case $z^T K z < 0$. If $\lambda < -1$, in the first case $z^T K z < 0$. So $\lambda > 1$ or $\lambda < -1$ makes the kernel not positive semi-definite.

Continue answer 5(a) here.

Continue answer 5(a) here.

(b) [10 points] Then, let's switch to the kernel $k_x(x_i, x_j)$. Assume that a potential candidate for $k_x(x_i, x_j)$ is given by

$$k_x(x_i, x_j) = \sum_{p=1}^{m} x_i^T M_p x_j, \tag{10}$$

where all M_p are symmetric positive semi-definite matrices. Show that the corresponding matrix K_x, with $(K_x)_{ij} = k_x(x_i, x_j)$ is positive semi-definite and hence is a valid kernel.

Hint: You may find the following useful. (i) the sum of symmetric positive semi-definite matrices is positive semi-definite, (ii) if a matrix M is symmetric positive semi-definite, it can be decomposed into $M = Q^T \Lambda Q$, where Λ is a diagonal matrix and the diagonal filled with eigenvalues larger or equal to 0.

Answer: Solution 1:

A positive semi-definite matrix could be decomposed by the eigen-decomposition into $M_p = Q_p^T \Lambda_p Q_p$, where Λ is a diagonal matrix with the diagonal elements are the eigenvalues of M_p. Since M_p is positive semi-definite, all the eigenvalues are larger or equal to 0. So we use $\Lambda^{\frac{1}{2}}$ to represent the diagonal matrix with the elements as the square root of Λ. So $M_p = \left(\Lambda_p^{\frac{1}{2}} Q_p\right)^T \left(\Lambda_p^{\frac{1}{2}} Q_p\right)$. Thus,

$$k_x(x_i, x_j) = \sum_{p=1}^{m} (\Lambda_p^{\frac{1}{2}} Q_p x_i)^T (\Lambda_p^{\frac{1}{2}} Q_p x_j). \tag{11}$$

Let $\phi_i(x_i) = \Lambda_p^{\frac{1}{2}} Q_p x_i$ and $\phi = \text{concatenate}(\phi_1, ..., \phi_m)$, then

$$k_x(x_i, x_j) = \phi(x_i)^T \phi(x_j). \tag{12}$$

Based on the Mercer's rule, k_x is positive semi-definite.

Solution 2:

For a kernel matrix K with n data points, we demonstrate that $\forall z \in \mathbb{R}^n$, $z^T K z \geq 0$.

$$z^T K z = \sum_{1 \leq i,j \leq n} \sum_{p=1}^{m} z_i x_i^T M_p x_j z_j \tag{13}$$

Here z_i is the i-th entry of z, which is a scalar. Let $a_i = \sum_{i=1}^{n} z_i x_i$, then

$$\begin{aligned} z^T K z &= \sum_{p=1}^{m} a^T M_p a \\ &= a^T (\sum_{p=1}^{m} M_p) a \end{aligned} \tag{14}$$

Since the sum of symmetric positive semi-definite matrices is symmetric positive semi-definite. So $a^T (\sum_{p=1}^{m} M_p) a \geq 0$ and also $z^T K z \geq 0$.

Continue answer 5(b) here.

Continue answer 5(b) here.

(c) [5 points] With k_d and k_x, could you design a new kernel k_{xd} defined for (x_i, d_i) and (x_j, d_j)? You may use any composition rules of kernels.

Answer: $k_d(d_i, d_j) + k_x(x_i, x_j), k_d(d_i, d_j)k_x(x_i, x_j)$ are all valid kernels.

6. **[20 points] + [10 Bonus points] Generalized Least-Squares Regression**

In the lectures, we have studied the ordinary least squares regression model, as well as various methods to optimize parameters including batch gradient descent, stochastic gradient descent and the normal equations. We also know that sometimes, by leveraging the information from the second-order gradient, Newton's method can converge much faster than gradient descent.

In the following sub-problems, we will first investigate why Newton's method can be particularly attractive when optimizing the ordinary least squares cost function, and then explore some generalizations of the standard least-squares regression problem.

(a) [10 points] Recall that given a twice differentiable loss function $\ell(\boldsymbol{\theta})$ for parameter $\boldsymbol{\theta} \in \mathbb{R}^d$, the update rule of the Newton's method is:

$$\boldsymbol{\theta} := \boldsymbol{\theta} - H(\boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})$$

where $H(\boldsymbol{\theta})$ is the $d \times d$ Hessian matrix, whose entries are given by $[H(\boldsymbol{\theta})]_{ij} = \frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_j}$. Suppose we use the Newton's method to optimize the following least squares regression loss function:

$$\ell(\boldsymbol{\theta}) = \|X\boldsymbol{\theta} - \boldsymbol{y}\|_2^2$$

where $X \in \mathbb{R}^{n \times d}$ is the feature matrix, $\boldsymbol{y} \in \mathbb{R}^d$ is the target vector.

Show that starting from a random initialization $\boldsymbol{\theta}^{(0)}$, after performing one step of Newton's method update, the obtained parameter $\boldsymbol{\theta}^{(1)}$ is the optimal solution (i.e., Newton's method converges in one step for linear regression).

Answer:

We can first calculate the gradient and Hessian as:

$$\begin{aligned}
\ell(\theta) &= \|X\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 \\
&= \boldsymbol{\theta}^\top (X^\top X)\boldsymbol{\theta} + \boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{\theta}^\top X^\top \boldsymbol{y} \\
\nabla \ell(\boldsymbol{\theta}^{(0)}) &= 2X^\top X \boldsymbol{\theta}^{(0)} - 2X^\top \boldsymbol{y} \\
H(\boldsymbol{\theta}^{(0)}) &= 2X^\top X
\end{aligned}$$

According to the Newton's method update rule, we have:

$$\begin{aligned}
\boldsymbol{\theta}^{(1)} &:= \boldsymbol{\theta}^{(0)} - [H(\boldsymbol{\theta}^{(0)})]^{-1} \nabla \ell(\boldsymbol{\theta}^{(0)}) \\
&= \boldsymbol{\theta}^{(0)} - (2X^\top X)^{-1}(2X^\top X \boldsymbol{\theta}^{(0)} - 2X^\top \boldsymbol{y}) \\
&= (X^\top X)^{-1} X^\top \boldsymbol{y}
\end{aligned}$$

which is the optimal solution according to the normal equation.

Continue answer 6(a) here.

(b) [10 points]

In this subproblem, we consider a generalized scenario, where there are two linear regression problems, with $A \in \mathbb{R}^{n \times d}$ and $B \in \mathbb{R}^{m \times d}$ being the feature matrices, and $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^m$ being the feature vectors respectively. We would like to learn a versatile linear regression parameter $\boldsymbol{\theta}$ such that it can simultaneously minimize the following two objectives:

$$\ell_1(\boldsymbol{\theta}) = \|A\boldsymbol{\theta} - \boldsymbol{a}\|_2^2, \quad \ell_2(\boldsymbol{\theta}) = \|B\boldsymbol{\theta} - \boldsymbol{b}\|_2^2.$$

Suppose we are given a constant $\mu > 0$ that specifies the relative important between ℓ_1 and ℓ_2, and the overall objective is:

$$\ell(\boldsymbol{\theta}) = \ell_1(\boldsymbol{\theta}) + \mu\ell_2(\boldsymbol{\theta}).$$

Show that the new problem can be reduced to a standard linear regression (give the definitions of matrix C and vector $\boldsymbol{c}$ in the following equation):

$$\ell(\boldsymbol{\theta}) = \|C\boldsymbol{\theta} - \boldsymbol{c}\|_2^2$$

and state the corresponding new normal equation (that characterizes the optimal solution) in terms of $A, B, \boldsymbol{a}, \boldsymbol{b}, \mu$.

Answer:

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &= \ell_1(\boldsymbol{\theta}) + \mu\ell_2(\boldsymbol{\theta}) \\
&= \|A\boldsymbol{\theta} - \boldsymbol{a}\|_2^2 + \mu\|B\boldsymbol{\theta} - \boldsymbol{b}\|_2^2 \\
&= (A\boldsymbol{\theta} - \boldsymbol{a})^\top(A\boldsymbol{\theta} - \boldsymbol{a}) + \mu(B\boldsymbol{\theta} - \boldsymbol{b})^\top(B\boldsymbol{\theta} - \boldsymbol{b}) \\
&= \boldsymbol{\theta}^\top A^\top A\boldsymbol{\theta} - 2\boldsymbol{a}^\top A\boldsymbol{\theta} + \boldsymbol{a}^\top\boldsymbol{a} + \mu(\boldsymbol{\theta}^\top B^\top B\boldsymbol{\theta} - 2\boldsymbol{b}^\top B\boldsymbol{\theta} + \boldsymbol{b}^\top\boldsymbol{b}) \\
&= \boldsymbol{\theta}^\top(A^\top A + \mu B^\top B)\boldsymbol{\theta} - 2(\boldsymbol{a}^\top A + \mu\boldsymbol{b}^\top B)\boldsymbol{\theta} + (\boldsymbol{a}^\top\boldsymbol{a} + \mu\boldsymbol{b}^\top\boldsymbol{b}) \\
&= \|C\boldsymbol{\theta} - \boldsymbol{c}\|_2^2 \\
&= \boldsymbol{\theta}^\top C^\top C\boldsymbol{\theta} - 2\boldsymbol{c}^\top C\boldsymbol{\theta} + \boldsymbol{c}^\top\boldsymbol{c}
\end{aligned}$$

where C and $\boldsymbol{c}$ are defined as:

$$C = \begin{bmatrix} A \\ \sqrt{\mu}B \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} \boldsymbol{a} \\ \sqrt{\mu}\boldsymbol{b} \end{bmatrix},$$

When C is full rank, the optimal solution is given by:

$$\begin{aligned}
\boldsymbol{\theta}^* &= (C^\top C)^{-1}C^\top\boldsymbol{c} \\
&= (A^\top A + \mu B^\top B)^{-1}(A^\top\boldsymbol{a} + \mu B^\top\boldsymbol{b})
\end{aligned}$$

Continue answer 6(b) here.

(c) [10 points] **BONUS Question: Optional!**

Recall that in our lectures, we mention that linear regression has a probabilistic interpretation, where we assume that the right model is:

$$\begin{aligned} \boldsymbol{y} &= X\boldsymbol{\theta} + \boldsymbol{\epsilon} \\ \mathbb{E}[\boldsymbol{\epsilon}|X] &= \mathbf{0} \\ \mathrm{Var}[\boldsymbol{\epsilon}|X] &= \sigma^2 \boldsymbol{I} \end{aligned} \tag{15}$$

Note that the last two equations assume that the noise $\epsilon^{(i)}$ for each data instance are independent and identically distributed as standard normal distribution $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$.

In this problem, we are going to relax this assumption and allow correlated noise. i.e., we assume that $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ with

$$\mathbb{E}[\boldsymbol{\epsilon}|X] = \mathbf{0}, \qquad \mathrm{Var}[\boldsymbol{\epsilon}|X] = \Sigma.$$

where Σ may not be a diagonal matrix.

Assuming that we know the covariance matrix Σ, can we reduce this problem to an ordinary least squares regression problem (with independent noise, i.e. the covariance matrix of the noise is diagonal), and what would be the corresponding normal equation in terms of $X, \boldsymbol{y}, \Sigma$?

Hint 1: Recall that Σ is a positive definite matrix and can be factorized as $\Sigma = LL^\top$, where L is a real, invertible, lower triangular matrix with positive diagonal entries.

Hint 2: Try starting from Equation (15) and modify it.

Some useful facts:

- $(A^{-1})^\top = (A^\top)^{-1}$
- $(AB)^{-1} = B^{-1}A^{-1}$
- For a random variable vector $\boldsymbol{a}$ with expectation $\boldsymbol{\mu}$, the covariance matrix of $\boldsymbol{a}$ is defined as: $\mathrm{Cov}[\boldsymbol{a}] = \mathbb{E}[(\boldsymbol{a} - \boldsymbol{\mu})(\boldsymbol{a} - \boldsymbol{\mu})^\top]$.

Continue answer 6(c) here.

Answer:

Yes, we can reduce it to an ordinary least squares regression problem. Since the covariance matrix Σ is symmetric and positive definite, we know that it can be written as $\Sigma = LL^\top$, where L is also invertible.

We start from the following equation:

$$\boldsymbol{y} = X\boldsymbol{\theta} + \boldsymbol{\epsilon}$$

and multiply both sides by L^{-1}:

$$L^{-1}\boldsymbol{y} = L^{-1}X\boldsymbol{\theta} + L^{-1}\boldsymbol{\epsilon}$$

Because $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \Sigma)$, $\mathbb{E}[L^{-1}\boldsymbol{\epsilon}] = \mathbf{0}$ and the covariance matrix can be derived as:

$$\begin{aligned}
\mathrm{Cov}[L^{-1}\boldsymbol{\epsilon}] &= \mathbb{E}[(L^{-1}\boldsymbol{\epsilon})(L^{-1}\boldsymbol{\epsilon})^\top] \\
&= L^{-1}\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top]L^{-\top} \\
&= L^{-1}\Sigma L^{-\top} \\
&= L^{-1}LL^\top L^{-\top} = \boldsymbol{I}
\end{aligned}$$

Therefore, if we know Σ, we can first get the matrix decomposition $\Sigma = LL^\top$, we can estimate the parameter $\boldsymbol{\theta}$ by doing an ordinary least squares regression of $L^{-1}\boldsymbol{y}$ on $L^{-1}X$ and the corresponding normal equation would be:

$$\begin{aligned}
\boldsymbol{\theta}^* &= ((L^{-1}X)^\top(L^{-1}X))^{-1}(L^{-1}X)^\top L^{-1}\boldsymbol{y} \\
&= (X^\top\Sigma^{-1}X)^{-1}X^\top\Sigma^{-1}\boldsymbol{y}.
\end{aligned}$$

That's all! Congratulations on completing the midterm exam!