IITM-CS5691 : Pattern Recognition and Machine Learning — Release Date: May 18, 2021
Assignment 3 — Due Date: Jun 8, 2021 (11.59pm)

Roll No: EE18B067 — Name:Abhishek Sekar
Collaborators (if any):
References (if any): Bishop PRML book, Class slides and personal notes

- Use LaTeX to write-up your solutions (in the solution blocks of the source LaTeX file of this assignment), and submit the resulting single pdf file at GradeScope by the due date. (Note: **No late submissions** will be allowed, other than one-day late submission with 10% penalty or four-day late submission with 30% penalty! Within GradeScope, indicate the page number where your solution to each question starts, else we won't be able to grade it! You can join GradeScope using course entry code **5VDNKV**).
- For the programming question, please submit your code (rollno.ipynb file and rollno.py file in rollno.zip) directly in moodle, but provide your results/answers in the pdf file you upload to GradeScope.
- Collaboration is encouraged, but all write-ups must be done individually and independently, and mention your collaborator(s) if any. Same rules apply for codes written for any programming assignments (i.e., write your own code; we will run plagiarism checks on codes).
- If you have referred a book or any other online material for obtaining a solution, please cite the source. Again don't copy the source *as is* - you may use the source to understand the solution, but write-up the solution in your own words.
- Points will be awarded based on how clear, concise and rigorous your solutions are, and how correct your code is. Overall points for this assignment would be **min**(your score including bonus points scored, 50).

1. (10 points) [HYPER-TUNE, SET, GO...] Hyperparameters have a crucial impact on the performance of machine learning models, and there are several techniques to perform hyperparameter tuning.
   (a) (4 points) Cross-validation (CV) with grid search is a predominant technique for hyperparameter tuning that involves building models for each of the possible combinations of hyperparameter values, and the combination that gives the best performance in a CV setting is chosen.
      i. (3 points) Specify the hyperparameter(s) of any three regression/classification models seen in our class.

Solution:
Hyperparameters:

- The regularization parameter λ in any regularized regression problem is a hyperparameter.
- The initial mean and variance (m_o, s_o) of the prior over the weights in Bayesian Linear regression are hyperparameters.

- The hyperparameters pertaining to a specific kernel in SVM and kernel regression methods, i.e, the parameter γ in a radial kernel(K(u,v) = $e^{\gamma u^T v}$) and the regularization parameter C.

ii. (1 point) For any one of the hyperparameters from previous question, specify what criteria could be used to choose its range and values (over which to do grid-search on).

Solution:
Choosing appropriate λ for regularized regression:
The criteria to choose the values of λ could be the model complexity we desire, i.e, for a more complex model, we'd like very small λ values and vice versa. We can compare different models through Bayesian Methods if necessary.Then, we can have the values of λ arranged in an exponential manner within the chosen range and perform a grid search to find the best order of λ. This process can then be repeated till we're satisfied.

(b) (6 points) An alternate approach called Empirical Bayes (EB) can help estimate hyperparameters from the training data (without having to do CV). Let θ be the parameters of a model with prior distribution $p(\theta|\alpha)$ and X denote the training data. EB estimate of the hyperparameter α is simply the MLE that maximizes the marginal likelihood $p(X|\alpha) = \int_\theta p(X|\theta)p(\theta|\alpha)d\theta$. Let's look at toy problems that illustrates EB, which you can then take forward to understand EB of Bayesian logistic/linear regression beyond the course.

Consider a generative classification model with m classes, with our training data $X = \{X_i\}_{i=1,\dots,m}$ comprising one observation per class. That is,

$$X_i \mid \mu_i \overset{\text{indep}}{\sim} N(\mu_i, \sigma^2), \quad i = 1, \dots, m$$
$$\mu_i \overset{\text{iid}}{\sim} N(\phi, \tau^2), \quad \sigma^2, \tau^2 \text{ known.}$$

Now, answer the questions below.

[Hint: While you can derive the results from scratch, try using known results about marginal/-conditional (multivariate) Gaussians to avoid derivations and cite which result you are using.]

i. (2 points) Write down the marginal likelihood $p(X|\phi)$.

Solution:
Expressing the marginal likelihood of $P\left(\frac{X_i}{\phi}\right)$:
From the expression for marginal likelihood given in the question,

$$P\left(\frac{X_i}{\phi}\right) = \int_{\mu_i} P\left(\frac{X_i}{\mu_i}\right) P\left(\frac{\mu_i}{\phi}\right) d\mu_i \tag{1}$$
$$\Rightarrow \int_{\mu_i} N(\mu_i, \sigma^2) N(\phi, \tau^2) d\mu_i \qquad \text{(From the distributions given)} \tag{2}$$

We'd expect $N(\mu_i, \sigma^2)N(\phi, \tau^2)$ to form another Gaussian distribution. Probing this idea further, we have,

$$N(\mu_i, \sigma^2)N(\phi, \tau^2) = \frac{1}{2\pi\sigma\tau} e^{\left(\frac{-(X_i-\mu_i)^2}{2\sigma^2}\right)} \cdot e^{\left(\frac{-(\mu_i-\phi)^2}{2\tau^2}\right)} \tag{3}$$

Concentrating on the terms inside the exponential,

$$\left(\frac{-(X_i-\mu_i)^2}{2\sigma^2}\right) + \left(\frac{-(\mu_i-\phi)^2}{2\tau^2}\right) = \left\{\left(\frac{-X_i^2}{2\sigma^2}\right) + \left(\frac{-\phi^2}{2\tau^2}\right)\right\}_{\text{constant in}\mu_i} + \left\{\left(\frac{-\mu_i^2+2\mu_i X_i}{2\sigma^2}\right) + \left(\frac{-\mu_i^2+2\mu_i\phi}{2\tau^2}\right)\right\}_{\mu_i}$$

Now further looking at the set of terms depending on μ_i and completing the squares, we get,

$$\left\{\left(\frac{-\mu_i^2+2\mu_i X_i}{2\sigma^2}\right) + \left(\frac{-\mu_i^2+2\mu_i\phi}{2\tau^2}\right)\right\}_{\mu_i} = \left\{\frac{-\left(\sqrt{\sigma^2+\tau^2}\mu_i - \frac{\sigma^2\phi+\tau^2 X_i}{\sqrt{\sigma^2+\tau^2}}\right)^2}{2\sigma^2\tau^2}\right\}_{\mu_i} + \left\{\left(\frac{\left(\frac{\sigma^2\phi+\tau^2 X_i}{\sqrt{\sigma^2+\tau^2}}\right)^2}{2\sigma^2\tau^2}\right)\right\}_{\text{constant in}\mu_i}$$

For simplicity, let us denote $\left\{\frac{-\left(\sqrt{\sigma^2+\tau^2}\mu_i - \frac{\sigma^2\phi+\tau^2 X_i}{\sqrt{\sigma^2+\tau^2}}\right)^2}{2\sigma^2\tau^2}\right\}_{\mu_i}$ by A.

Now, incorporating the extra term we obtained which is constant in μ_i to the other terms constant in μ_i and simplifying it, we obtain,

$$\left\{\left(\frac{\left(\frac{\sigma^2\phi+\tau^2 X_i}{\sqrt{\sigma^2+\tau^2}}\right)^2}{2\sigma^2\tau^2}\right)\right\} + \left\{\left(\frac{-X_i^2}{2\sigma^2}\right) + \left(\frac{-\phi^2}{2\tau^2}\right)\right\} = \left\{\frac{-(X_i-\phi)^2}{2(\sigma^2+\tau^2)}\right\}$$

For simplicity, let us denote $\left\{\frac{-(X_i-\phi)^2}{2(\sigma^2+\tau^2)}\right\}$ by B.

Now, with the set up ready, we can use this to simplify 3 and plug it into the integral in 1.

Therefore with all of these,1 becomes,

$$\int_{\mu_i} P\left(\frac{X_i}{\mu_i}\right) P\left(\frac{\mu_i}{\phi}\right) d\mu_i = \frac{1}{\sqrt{2\pi}} e^B \int_{\mu_i} \frac{1}{\sqrt{2\pi}\sigma\tau} e^A d\mu_i$$

Observing that $\frac{1}{\sqrt{2\pi}\sigma\tau}e^{A}$ is nothing but $\sqrt{\sigma^2+\tau^2}\mu_i \sim N\left(\frac{\sigma^2\phi+\tau^2 X_i}{\sqrt{\sigma^2+\tau^2}}, \sigma^2\tau^2\right)$, we can change the parameters of the integral from μ_i to $\sqrt{\sigma^2+\tau^2}\mu_i$.
Then, the integral evaluates to,

$$\frac{1}{\sqrt{2\pi}}e^{B}\int_{\mu_i}\frac{1}{\sqrt{2\pi}\sigma\tau}e^{A}d\mu_i = \frac{1}{\sqrt{2\pi\left(\sigma^2+\tau^2\right)}}e^{B}\int_{\sqrt{\sigma^2+\tau^2}\mu_i}\frac{1}{\sqrt{2\pi}\sigma\tau}e^{A}d\sqrt{\sigma^2+\tau^2}\mu_i$$
$$\Rightarrow \frac{1}{\sqrt{2\pi\left(\sigma^2+\tau^2\right)}}e^{B} \text{ (As } \int_x N\left(\mu,\sigma^2\right)dx = 1)$$

Therefore, with this, we can say that,

$$P\left(\frac{X_i}{\phi}\right) = \frac{1}{\sqrt{2\pi\left(\sigma^2+\tau^2\right)}}e^{\left\{\frac{-(X_i-\phi)^2}{2(\sigma^2+\tau^2)}\right\}} \tag{4}$$

From $P\left(\frac{X_i}{\phi}\right)$ to $P\left(\frac{X}{\phi}\right)$:
Now as each $X_i \mid \mu_i$ and $\mu_i \mid \phi$ are i.i.d, we can assert that $X_i \mid \phi$ is independent. This is doubly reinforced by the form of the expression in 4.
Therefore,

$$P\left(\frac{\mathbf{X}}{\phi}\right) = \pi_{i=1}^{m}P\left(\frac{X_i}{\phi}\right) \tag{5}$$
$$\Rightarrow \left(\frac{1}{\sqrt{2\pi\left(\sigma^2+\tau^2\right)}}\right)^{m}\left(e^{\left\{\frac{-\sum_{i=1}^{m}(X_i-\phi)^2}{2(\sigma^2+\tau^2)}\right\}}\right) \tag{6}$$

ii. (2 points) Use the computed marginal likelihood to derive the hyperparameter's EB estimate, denoted $\widehat{\phi}$.

Solution:
EB estimate:
From the information given in the problem, the EB estimate of $\phi = \widehat{\phi}$ is nothing but the MLE of the marginal likelihood expression derived in 5.
Closely observing the expression in 5, we see that $\left(\frac{1}{\sqrt{2\pi(\sigma^2+\tau^2)}}\right)^{m}\left(e^{\left\{\frac{-\sum_{i=1}^{m}(X_i-\phi)^2}{2\sqrt{\sigma^2+\tau^2}}\right\}}\right)$ is what we'll get in an alternative scenario should our training dataset be distributed as $X_i \overset{\text{indep}}{\sim} N\left(\phi,\left(\sigma^2+\tau^2\right)\right), \quad i = 1,\ldots,m$ for an unknown parameter ϕ.

Employing the standard result for the MLE of the mean for m datapoints, shown below,

$$\mu_{ML} = \frac{1}{m}\sum_{i=1}^{m} X_i = \bar{X} \tag{7}$$

Therefore, using the formula in 7 and solving the aforementioned equivalent problem that results in the same expression, $\widehat{\phi} = \bar{X}$

iii. (2 points) What is the posterior mean of μ_i at this EB estimate (i.e., $E[\mu_i|X, \widehat{\phi}]$)? Show that it has an intuitive interpretation as the weighted average of X_i and $\bar{X}$.

Solution:
Posterior mean of μ_i:
The posterior mean $E[\mu_i|X, \widehat{\phi}]$ is nothing but the mean or expectation of μ_i given we know X and $\widehat{\phi}$.
From our derivation in the first subdivision,
we know that, $\sqrt{\sigma^2+\tau^2}\mu_i \sim N\left(\frac{\sigma^2\phi+\tau^2 X_i}{\sqrt{\sigma^2+\tau^2}}, \sigma^2\tau^2\right)$. Using the below property of a Normal distribution,

$$X \sim N(\mu, \sigma^2) \tag{8}$$
$$cX \sim N(c\mu, c^2\sigma^2) \text{ for some } c \in \mathbb{R} \tag{9}$$

Substituting for $c = \frac{1}{\sqrt{\sigma^2+\tau^2}}$ and using 8 we can find that $\mu_i \sim N\left(\frac{\sigma^2\phi+\tau^2 X_i}{\sigma^2+\tau^2}, \frac{\sigma^2\tau^2}{\sigma^2+\tau^2}\right)$.
Now, given that we're taking the EB estimate of $\phi = \widehat{\phi}$, plugging that, we have, $\mu_i \sim N\left(\frac{\sigma^2\widehat{\phi}+\tau^2 X_i}{\sigma^2+\tau^2}, \frac{\sigma^2\tau^2}{\sigma^2+\tau^2}\right)$.
Therefore, from this and the results from the previous subdivision, we have,

$$E[\mu_i|X, \widehat{\phi}] = \frac{\sigma^2\widehat{\phi} + \tau^2 X_i}{\sigma^2+\tau^2} \qquad \text{(Comparing parameters of the distribution of } u_i \mid (X_i, \widehat{\phi}))$$
$$\Rightarrow \frac{\sigma^2\bar{X} + \tau^2 X_i}{\sigma^2+\tau^2} \qquad \text{(Substituting for } \widehat{\phi})$$
$$\Rightarrow \frac{\tau^2 X_i}{\sigma^2+\tau^2} + \frac{\sigma^2\bar{X}}{\sigma^2+\tau^2}$$

Therefore, we can see that the posterior mean is the weighted average of X and $\bar{X}$.This expression is intuitive as it can be interpreted as the sum of the respective contributions of X_i and $\bar{X}$ to μ_i.

This becomes clear if we rewrite the expression as shown below,

$$\frac{\tau^2 X_i}{\sigma^2+\tau^2} + \frac{\sigma^2 \bar{X}}{\sigma^2+\tau^2} = \frac{X_i}{\frac{\sigma^2}{\sigma^2_{u_i}}} + \frac{\bar{X}}{\frac{\tau^2}{\sigma^2_{u_i}}}$$

$$\sigma^2_{u_i} = \frac{\sigma^2\tau^2}{\sigma^2+\tau^2}$$

Where, in crude terms, the contribution of $\mu_i \mid X_i = X_i$ and the contribution of $\mu_i \mid \widehat{\phi} = \bar{X}$ gets divided by their respective contribution to the variance of μ_i once both these quantities are known.

(c) (3 points) [OPTIONAL BONUS] Answer the same three questions above when you've n observations (instead of one) per class for a total of nm observations.

Solution:
How the answers change should we have N observations per class:
Sample mean and Sample Variance:
As we have n observations per class instead of just 1 like we had before, we can interpret this as getting n-1 additional observations to model the density of X_i belonging to the i^{th} class.Let the datapoints be distributed as $X_{ij} \mid \mu_i \overset{iid}{\sim} N(\mu_i, \sigma^2), i = 1, \ldots, m, j = 1, ..., n$.
As we have n datapoints for class i, without loss of generality, we can choose a particular X_i in terms of the respective $\{X_{ij}\}_{\{j=1,\ldots,n\}}$ and then attempt at finding the distribution for this particular X_i.
Let us choose $X_i = \frac{1}{n}\sum_{j=1}^{n} X_{ij} = \mathbb{E}_j[X_{ij}]$, the mean of the n observations from that class.
Shown below are the mean and variance for the distribution involving our chosen X_i given the means and variances of the i.i.d X_{ij}.

$$\begin{aligned}
\mathbb{E}[X_i] &= \mathbb{E}\left[\frac{1}{n}\sum_{j=1}^{n} X_{ij}\right] \\
&\Rightarrow \frac{1}{n}\sum_{j=1}^{n} \mathbb{E}[X_{ij}] && \text{(Linearity of expectation)} \\
&\Rightarrow \mu_i && \text{(As } \mathbb{E}[X_{ij}] = \mu_i \text{ for each j)}
\end{aligned}$$

$$\begin{aligned} var(X_i) &= var\left(\frac{1}{n}\sum_{j=1}^{n} X_{ij}\right) \\ &\Rightarrow \frac{1}{n^2}\sum_{j=1}^{n} var(X_{ij}) && \text{(By properties of variance)} \\ &\Rightarrow \frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n} && \text{(As } var(X_{ij}) = \sigma^2 \text{ for each j)} \end{aligned}$$

From this, we can see that $X_i \mid \mu_i \overset{\text{indep}}{\sim} N(\mu_i, \frac{\sigma^2}{n}), \quad i = 1, \ldots, m$ for our choice of X_i. Now that we know this, we can use this expression to compute the three subdivisions.

The solutions to the three questions:

- $P\left(\frac{X}{\phi}\right) = \left(\frac{1}{\sqrt{2\pi\left(\frac{\sigma^2}{n}+\tau^2\right)}}\right)^m \left(e^{\left\{\frac{-\sum_{i=1}^{m}(X_i-\phi)^2}{2\left(\frac{\sigma^2}{n}+\tau^2\right)}\right\}}\right)$
- $\widehat{\phi} = \bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}$
- $E[\mu_i | X, \widehat{\phi}] = \frac{n\tau^2 X_i}{\sigma^2+n\tau^2} + \frac{\sigma^2 \bar{X}}{\sigma^2+n\tau^2}$

(d) (2 points) [OPTIONAL BONUS] How will your answer to the above question simplify if you've a density estimation problem, where there is only one class with n observations? That is, $X_j \mid \mu \overset{\text{iid}}{\sim} N(\mu, \sigma^2)$, $j = 1, ..., n$ with σ^2 known, and a prior distribution $\mu \sim N(\phi, \tau^2)$ with hyperparameter ϕ unknown and τ^2 known. Do you see any issues with EB in this question here, and can it be overcome with historical data (i.e., data collected prior to current data X)?

Solution:

The solutions to the three questions:

The given question is a specific simplified version of the previous problem with m = 1.

Substituting this in the answers derived in the previous problem and using the substitution $X = \frac{1}{n}\sum_{j=1}^{n} X_j = \mathbb{E}_j[X_j]$, we get,

- $P\left(\frac{x}{\phi}\right) = \left(\frac{1}{\sqrt{2\pi\left(\frac{\sigma^2}{n}+\tau^2\right)}}\right)\left(e^{\left\{\frac{-(X-\phi)^2}{2\left(\frac{\sigma^2}{n}+\tau^2\right)}\right\}}\right)$
- $\widehat{\phi} = \bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i = \frac{1}{n}\sum_{j=1}^{n} X_j = X$
- $E[\mu_i|X, \widehat{\phi}] = \frac{n\tau^2 X_i}{\sigma^2+n\tau^2} + \frac{\sigma^2 \bar{X}}{\sigma^2+n\tau^2} = \frac{n\tau^2 X}{\sigma^2+n\tau^2} + \frac{\sigma^2 X}{\sigma^2+n\tau^2} = X$, as X_i with one class is just X.

Issue with the EB approach:
As we can see in 7, the ML approach for this density estimation problem gives us $\mu_{ML} = X$ which is the same as the EB estimate for the hyperparameter ϕ rendering it useless. This wrongly conveys that the hyperparameter doesn't actually provide any new information about the data when in reality it does.
Overcoming the issue:
Yes it can be overcome using historical data.
This can be overcome by modeling the mean through historical data and updating it after looking at the dataset. This can be shown using Bayes Theorem.

- **Prior:** $P\left(\frac{\mu}{\phi}\right)$
- **Likelihood:** $P\left(\frac{\{X_j\}_{\{j=1,2,\ldots,n\}}}{\mu}\right)$
- **Posterior:** $P\left(\frac{\mu}{\phi,\{X_j\}_{\{j=1,2,\ldots,n\}}}\right) \propto P\left(\frac{\mu}{\phi}\right) \cdot P\left(\frac{\{X_j\}_{\{j=1,2,\ldots,n\}}}{\mu}\right)$

The posterior (conjugate prior) distribution influenced by the historical data and the current dataset tells us the distribution of μ we can consider.This overcomes the limitation of EB. As the size of the dataset $\to \infty$ the $\mathbb{E}[\mu]$ value approaches that of the MLE which justifies the presence of hyperparameter ϕ used while modeling μ through the historical data for finite n.

2. (10 points) [(LOGISTICALLY) CLASSIFIED INFORMATION]
   (a) (3 points) Consider the following 2-dimensional (2D) binary classification dataset with 8 points given by:

$$X^T = \begin{bmatrix} -2 & -2 & -1 & -1 & 1 & 1 & 2 & 3 \\ -1 & 2 & 1 & 2 & 1 & 3 & 3 & 2 \end{bmatrix}$$

$$y^T = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 \end{bmatrix}$$

Run one iteration of gradient descent with the logistic regression objective by hand (pen-and-paper). No bias required, only the 2D weight vector is to be optimized. Choose the step size $\eta = 1$. Initialise at $w = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$.

Solution:
One iteration of Gradient Descent:

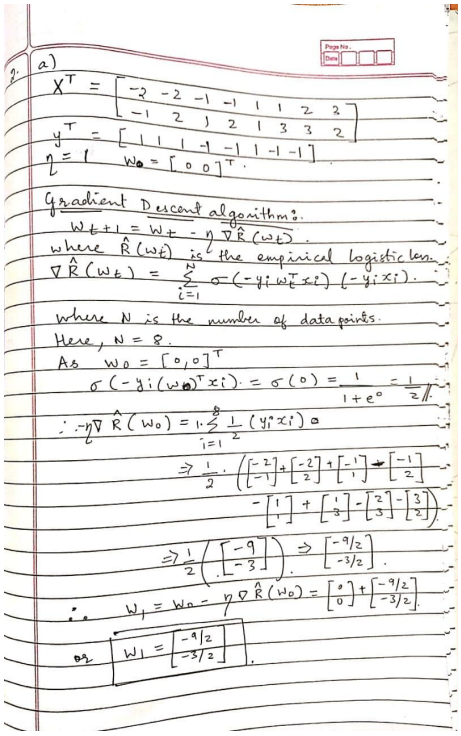

Figure 1: One iteration of Gradient Descent for problem 2 (a)

(b) (2 points) Plot the sigmoid function $\dfrac{1}{1+e^{-wX}}$ vs $X \in \mathbb{R}$ for $w \in [1, 5, 100]$. A qualitative sketch would also work. From these plots infer why logistic regression can lead to overfitting if the the weights are high.

Solution:
Plots of the sigmoid function for different weights:

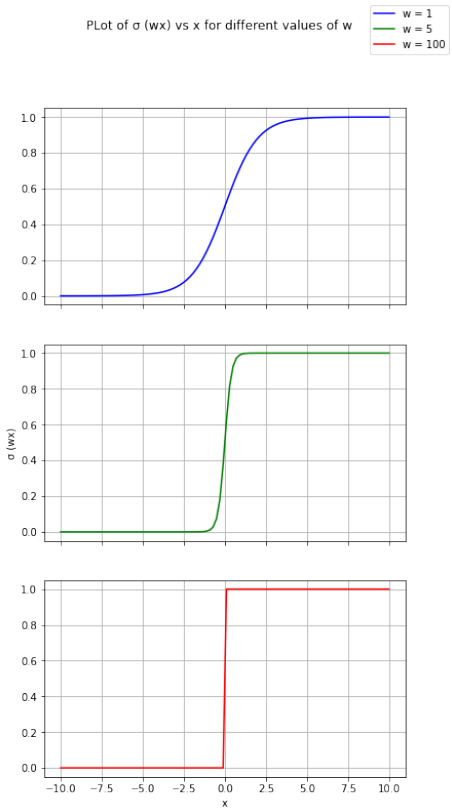

Figure 2: Plots of $\sigma(wx)$ vs x for $w \in [1, 5, 100]$

Inferences and Explanation:
From the plots shown above, we can see that as the magnitude of the weights increase, the sigmoid function becomes almost heaviside like, i.e asymptotically approaches it, where the heaviside step function is given below.

$$u(t) = \begin{cases} 1 & \text{When } t > 0 \\ 0 & \text{When } t < 0 \end{cases}$$

For the sake of illustration, if we consider the toy example taken in class, where the class label $y^{(i)} \in \{+1, -1\}$.
Then, the posterior probability and the decision boundaries are given as follows:

$$P\left(\frac{y^{(i)}}{x^{(i)}}\right) = \sigma\left(y^{(i)}\left(w^T x^{(i)}\right)\right)$$

Decision boundary = $\sigma(0.5)$

$$\Rightarrow \sigma(0.5) : y^{(i)}\left(w^T x^{(i)}\right) = 0$$

$$\Rightarrow w^T x^{(i)} = 0$$

Therefore, the sigmoid assigns a probability pertaining to either class for each datapoint $x^{(i)}$.
As seen in the plots, as the weights get larger and larger, the sigmoid starts looking like a heaviside function. This means that the posterior probabilities tend to be *binary* in nature with a value of 1 implying $x^{(i)}$ belongs to class $y^{(i)}$ and 0 implying otherwise, indicating surety. Additionally, observing the decision boundary,as the weights are large, even $x^{(i)}$s for which the value $\sigma\left(y^{(i)}\left(w^T x^{(i)}\right)\right)$ ends up being very close to the decision boundary are assigned these *binary* probabilities despite the presence of some ambiguity. This is a classic case of overfitting, as it performs a "hard" classification even for ambiguous datapoints and the model fits to the noise present in the training data.
Therefore, logistic regression tends to overfit with the presence of large weights.

(c) (5 points) Let us look at multi-class logistic regression. Say we have K classes and each input x is a d-dimensional vector. The posterior probability (after ignoring the bias term) is given by:

$$P(Y = k \mid X = x) = \frac{e^{w_k^T x}}{\sum_{l=1}^{K} e^{w_l^T x}} \quad k = 1, 2 \cdots, K$$

i. (1 point) What are the parameters to estimate and how many are there?

Solution:

- **Parameters to estimate:** The weights $w_k \forall k \in \{1, 2, \ldots, K\}$.
- **Number of parameters:** There are K weights w_k and each w_k is d-dimensional.As there is no bias term, the total number of parameters to estimate = Kd.Once we know the w_ks we can easily find the posterior probabilities.

ii. (2 points) Given n training samples $\{(x_1, y_1), (x_2, y_2), \cdots (x_n, y_n)\}$, write down the log likelihood function, and simplify.

Solution:
Expression for the likelihood:

The likelihood $P\left(\frac{t^{(i)}}{\{w_k\}_{\{k=1,2,\ldots,K\}}}\right)$ where $t^{(i)}$ represents the target class of the i^{th} datapoint is given as follows if x_i belongs to class K.

$$P\left(\frac{t^{(i)}}{\{w_k\}_{\{k=1,2,\ldots,K\}}}\right) \propto P(\frac{Y=k}{X=x_i}) = \frac{e^{w_k^T x_i}}{\sum_{l=1}^{K} e^{w_l^T x_i}} \tag{10}$$

For simplicity,lets denote $P(\frac{Y=k}{X=x_i})$ by $y_i^{(k)}$ and $\{w_k\}_{\{k=1,2,\ldots,K\}}$ by θ
Expression for the Log Likelihood:
Now, since there are n i.i.d datapoints, the likelihood $\mathcal{L}(\theta;\mathcal{D})$ is given by:

$$\mathcal{L}(\theta;\mathcal{D}) = \prod_{i=1}^{n} P\left(\frac{t^{(i)}}{\theta}\right)$$

Let us define a target matrix **T** with dimensions NxK where the i^{th} row represents the one-hot representation of the class the i^{th} datapoint is allocated to.

$$T_{ij} = \begin{cases} 1 & \text{if } x_i \text{ belongs to class j} \\ 0 & \text{if } x_i \text{ does not belong to class j} \end{cases}$$

Then, the likelihood can be written as,

$$\mathcal{L}(\theta;\mathcal{D}) = \prod_{i=1}^{n}\prod_{j=1}^{K}\left(y_i^{(j)}\right)^{T_{ij}} \qquad \text{(From 10)}$$

This takes care that only the $y_i^{(j)}$ corresponding to the class x_i belongs to is taken into account as T_{ij} will be 0 for the other cases.
Therefore the simplified log likelihood is,

$$\mathcal{LL}(\theta;\mathcal{D}) = \sum_{i=1}^{n}\sum_{j=1}^{K} T_{ij}\log\left(y_i^{(j)}\right) \tag{11}$$

The negative of the expression in 11 is known as the *Cross Entropy* error function for the given problem.

iii. (2 points) Compute the gradient of the log likelihood with respect to w_k, and simplify.

Solution:
The gradient of $\mathcal{LL}(\theta;\mathcal{D})$ with respect to w_k:
The expression of the Log Likelihood is shown in 11.

The gradient of the log likelihood with respect to w_k is given below.

$$\nabla_{w_k}\mathcal{LL}\left(\theta;\mathcal{D}\right)=\sum_{i=1}^{n}\sum_{j=1}^{K}\frac{\partial\mathcal{LL}\left(\theta;\mathcal{D}\right)}{\partial y_i^{(j)}}\cdot\frac{\partial y_i^{(j)}}{\partial w_k}\text{(Differentiating by chain rule)}\tag{12}$$

Computing the two required partial derivatives in 12,

$$\frac{\partial\mathcal{LL}\left(\theta;\mathcal{D}\right)}{\partial y_i^{(j)}}=\frac{T_{ij}}{y_i^{(j)}}\tag{13}$$

$$\frac{\partial y_i^{(j)}}{\partial w_k}=\frac{\partial\left(\frac{e^{w_j^T x_i}}{\sum_{l=1}^{K}e^{w_l^T x_i}}\right)}{\partial w_k}$$

- If j=k, differentiating by parts, we get,

$$\begin{aligned}\frac{\partial\left(\frac{e^{w_j^T x_i}}{\sum_{l=1}^{K}e^{w_l^T x_i}}\right)}{\partial w_k}&=x_i^T\left(\frac{e^{w_j^T x_i}}{\sum_{l=1}^{K}e^{w_l^T x_i}}\right)-x_i^T\left(\frac{e^{w_j^T x_i}}{\sum_{l=1}^{K}e^{w_l^T x_i}}\right)^2 &\text{(As } w^T x=x^T w\text{)}\\&\Rightarrow x_i^T y_i^{(k)}\left(1-y_i^{(k)}\right) &\text{(As k=j)}\end{aligned}$$

- If $j\neq k$, differentiating by parts, we get,

$$\begin{aligned}\frac{\partial\left(\frac{e^{w_j^T x_i}}{\sum_{l=1}^{K}e^{w_l^T x_i}}\right)}{\partial w_k}&=-x_i^T\left(\frac{e^{w_j^T x_i}e^{w_k^T x_i}}{\left(\sum_{l=1}^{K}e^{w_l^T x_i}\right)^2}\right) &\text{(As } w^T x=x^T w\text{)}\\&\Rightarrow x_i^T\left(-y_i^{(k)}y_i^{(j)}\right)\end{aligned}$$

Expressing both of these under a single expression, we have,

$$\frac{\partial y_i^{(j)}}{\partial w_k}=x_i^T y_i^{(j)}\left(\mathbb{I}_{kj}-y_i^{(k)}\right)\tag{14}$$

Where $\mathbb{I}$ is the identity matrix.

Substituting the results obtained in 13 and 14 in 12 we have,

$$\nabla_{w_k}\mathcal{LL}(\theta;\mathcal{D}) = \sum_{i=1}^{n}\sum_{j=1}^{K}\frac{T_{ij}}{y_i^{(j)}}\cdot x_i^T y_i^{(j)}\left(\mathbb{I}_{kj} - y_i^{(k)}\right)$$

$$\Rightarrow \sum_{i=1}^{n}\sum_{j=1}^{K} T_{ij}\cdot x_i^T\left(\mathbb{I}_{kj} - y_i^{(k)}\right) \qquad (\text{As } y_i^{(j)} \neq 0)$$

$$\Rightarrow \sum_{i=1}^{n} x_i^T\left(T_{ik} - y_i^{(k)}\right)$$

As $\sum_{j=1}^{K} T_{ij}\mathbb{I}_{kj} = T_{ik}$ and $\sum_{j=1}^{K} T_{ij} = 1$, i.e, only one class for a datapoint.

Therefore, the simplified expression for the gradient of the log likelihood with respect to w_k is,

$$\nabla_{w_k}\mathcal{LL}(\theta;\mathcal{D}) = \sum_{i=1}^{n} x_i^T\left(T_{ik} - y_i^{(k)}\right) \tag{15}$$

3. (10 points) [KERNELIZE...] Let K_1 and K_2 be a valid kernel functions, with feature mapping $\varphi_1 : \mathbb{R}^d \to \mathbb{R}^{d_1}$ and $\varphi_2 : \mathbb{R}^d \to \mathbb{R}^{d_2}$.

   (a) (2 points) Show that $K_3 = K_1 + K_2$ is also a valid kernel. Give the feature mapping φ_3 corresponding to K_3 in terms of φ_1 and φ_2.

Solution:
Properties of a Kernel Function:

- K: $\mathbb{R}^d x \mathbb{R}^d \to \mathbb{R}$ is a kernel function if, $K(u,v)$ can be expressed as an inner product of feature mappings of u and v, i.e, $K(u,v) = \varphi(u)^T\varphi(v)$ for some feature mapping $\varphi : \mathbb{R}^d \to \mathcal{H}$ where $\mathcal{H}$ is a Hilbert Space.

Proof:
Now given we know this, to show that K_3 is a kernel function, we just have to prove that it

can be expressed in the above form.
Using the above property on some vectors u and $v \in \mathbb{R}^d$

$$\begin{aligned} K_1(u,v) &= \varphi_1(u)^T \varphi_1(v) \\ K_2(u,v) &= \varphi_2(u)^T \varphi_2(v) \\ \Rightarrow K_1(u,v) + K_2(u,v) &= \varphi_1(u)^T \varphi_1(v) + \varphi_2(u)^T \varphi_2(v) \end{aligned}$$

As both feature mappings are finite dimensional,
Expressing them in matrix form with $\varphi^{(i)}(u) : \mathbb{R}^d \to \mathbb{R}$ representing the i^{th} basis function of the mapping φ, i.e, $\varphi = \begin{bmatrix} \varphi^{(1)} & \dots & \varphi^{(d_\varphi)} \end{bmatrix}$ we have,

$$K_1(u,v) + K_2(u,v) = \begin{bmatrix} \varphi_1^{(1)}(u) & \dots & \varphi_1^{(d_1)}(u) \end{bmatrix} \begin{bmatrix} \varphi_1^{(1)}(v) \\ \vdots \\ \varphi_1^{(d_1)}(v) \end{bmatrix} + \begin{bmatrix} \varphi_2^{(1)}(u) & \dots & \varphi_2^{(d_2)}(u) \end{bmatrix} \begin{bmatrix} \varphi_2^{(1)}(v) \\ \vdots \\ \varphi_2^{(d_2)}(v) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \varphi_1^{(1)}(u) & \dots & \varphi_1^{(d_1)}(u) & \varphi_2^{(1)}(u) & \dots & \varphi_2^{(d_2)}(u) \end{bmatrix} \begin{bmatrix} \varphi_1^{(1)}(v) \\ \vdots \\ \varphi_1^{(d_1)}(v) \\ \varphi_2^{(1)}(v) \\ \vdots \\ \varphi_2^{(d_2)}(v) \end{bmatrix}$$

$$\Rightarrow \varphi_3(u)^T \varphi_3(v) = K_3(u,v)$$

Therefore, K_3 is indeed a valid kernel with a feature mapping $\varphi_3 : \mathbb{R}^d \to \mathbb{R}^{d_1} \bigoplus \mathbb{R}^{d_2}$ where $\varphi_3 = \varphi_1 \bigoplus \varphi_2 = \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix}$, the concatenation of the two feature mappings.

(b) (2 points) Show that $K_4 = K_1 \cdot K_2$ is also a valid kernel. Give the feature mapping φ_4 corresponding to K_4 in terms of φ_1 and φ_2.

Solution:
Proof:
Proceeding like in the previous subdivision,
We have,

$$\begin{aligned} K_1(u,v) &= \varphi_1(u)^T \varphi_1(v) \\ K_2(u,v) &= \varphi_2(u)^T \varphi_2(v) \\ K_1(u,v) \cdot K_2(u,v) &= \left(\varphi_1(u)^T \varphi_1(v)\right) \cdot \left(\varphi_2(u)^T \varphi_2(v)\right) \end{aligned}$$

Rewriting the above expression as a product of sums,
We have,

$$
\begin{aligned}
K_1(u,v) \cdot K_2(u,v) &= \left(\varphi_1(u)^T \varphi_1(v)\right) \cdot \left(\varphi_2(u)^T \varphi_2(v)\right) \\
&\Rightarrow \sum_{i=1}^{d_1} \left(\varphi_1^{(i)}(u)\varphi_1^{(i)}(v)\right) \cdot \sum_{j=1}^{d_2} \left(\varphi_2^{(j)}(u)\varphi_2^{(j)}(v)\right) \\
&\Rightarrow \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \left(\varphi_1^{(i)}(u)\varphi_1^{(i)}(v) \cdot \varphi_2^{(j)}(u)\varphi_2^{(j)}(v)\right) \\
&\Rightarrow \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \left(\varphi_1^{(i)}(u)\varphi_2^{(j)}(u) \cdot \varphi_1^{(i)}(v)\varphi_2^{(j)}(v)\right) \qquad \text{(As } \varphi^{(i)}(x) \text{ is a scalar)} \\
&\Rightarrow \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \left(\varphi_4^{(ij)}(u) \cdot \varphi_4^{(ij)}(v)\right) \\
&\Rightarrow \varphi_4(u)^T \varphi_4(v) = K_4(u,v)
\end{aligned}
$$

Therefore, K_4 is indeed a valid kernel with a feature mapping $\varphi_4 : \mathbb{R}^d \to \mathbb{R}^{d_1 d_2}$ where $\varphi_4 = \varphi_1 \bigotimes \varphi_2$,the outer product of the two feature mappings, i.e, $\varphi_4^{(ij)} = \varphi_1^{(i)}\varphi_2^{(j)} \forall i \in \{1, 2, \ldots, d_1\}, j \in \{1, 2, \ldots, d_2\}$.

(c) (2 points) Show that $K_5 = f(u)K_1(u,v)f(v)$ is also a valid kernel for any function $f : \mathbb{R}^d \to \mathbb{R}$. Give the feature mapping φ_5 corresponding to K_5 in terms of φ_1 and f.

Solution:
Proof:
Proceeding like in the previous subdivisions,
We have,

$$
\begin{aligned}
K_1(u,v) &= \varphi_1(u)^T \varphi_1(v) \\
f(u)K_1(u,v)f(v) &= f(u)\varphi_1(u)^T \varphi_1(v) f(v)
\end{aligned}
$$

Now, as $f : \mathbb{R}^d \to \mathbb{R}$ returns a scalar,

$$
\begin{aligned}
f(u)K_1(u,v)f(v) &= f(u)\varphi_1(u)^T \varphi_1(v) f(v) \\
&\Rightarrow \left(f(u)\varphi_1(u)\right)^T \varphi_1(v) f(v) \\
&\Rightarrow \varphi_5(u)^T \varphi_5(v) = K_5(u,v)
\end{aligned}
$$

Therefore, K_5 is indeed a valid kernel with a feature mapping $\varphi_5 : \mathbb{R}^d \to \mathbb{R}^{d_1}$ where $\varphi_5(u) = f(u) \cdot \varphi_1(u)$.

(d) (2 points) Show that a Kernel given by $K(u, v) = \exp(2u^T v)$ is a valid kernel. [Hint: Use the results above on a polynomial expansion of $\exp(t)$.]

Solution:
Proof:
Using the hint, we can employ the Maclaurin series of e^t shown below.

$$e^t = \sum_{i=0}^{\infty} \frac{t^i}{i!} \tag{16}$$

Using 16 and substituting $t = 2u^T v$, we have,

$$e^{(2u^T v)} = \sum_{i=0}^{\infty} \frac{(2u^T v)^i}{i!}$$

Using the results derived above, we can show that the above function is a kernel.

- **Demonstrating that $u^T v$ is a valid kernel:**
  $u^T v$ is a valid kernel as it can be expressed as $\varphi(u)^T \varphi(v)$ for $\varphi(u) = u$.

- **Demonstrating that $\frac{(2u^T v)^i}{i!}$ is a valid kernel :**
  As derived in the previous subdivisions, if $K_1(u, v)$ and $K_2(u, v)$ are two valid kernels, their product $K_3(u, v) = K_1(u, v) \cdot K_2(u, v)$ is also a valid kernel.As shown above, $u^T v$ is a valid kernel. Therefore, employing this result repeatedly in a sequential fashion, using the principles of mathematical induction, we can show that $(u^T v)^i$ is a valid kernel.

  Let $K_i(u, v) = (u^T v)^i$
  Then $K_2(u, v) = (u^T v)^2 = K_1(u, v) \cdot K_1(u, v)$ is a valid kernel
  Consider for some $i \in \mathbb{N}$, $K_i(u, v) = K_{i-1}(u, v) \cdot K_1(u, v)$ as a valid kernel
  Proving that $K_{i+1}(u, v)$ is a valid kernel using the above step,
  $K_{i+1}(u, v) = K_i(u, v)K_1(u, v)$ is a valid kernel being the product of two kernels
  Therefore, using the principle of mathematical induction, $K_i(u, v)$ is a valid kernel $\forall i \in \mathbb{N}$

  Now, we know that if $K_1(u, v)$ is a valid kernel any $cK_1(u, v), c \in \mathbb{R}$ is also a kernel, by employing the result derived in the previous subdivision, by substituting $f(u) = \sqrt{c} \forall u \in \mathbb{R}^d$.Choosing $c = \frac{2^i}{i!}$, $\frac{(2u^T v)^i}{i!}$ is a valid kernel.

- **Demonstrating that $\sum_{i=0}^{\infty} \frac{(2u^T v)^i}{i!}$ is a valid kernel :**
  We know that, the sum of two kernels is a valid kernel.As in the previous step, we

can again employ the principles of mathematical induction to prove that the given function is a valid kernel.

Let $K_j(u,v) = \sum_{i=1}^{j} \frac{(2u^T v)^i}{i!}$

Then $K_2(u,v) = \sum_{i=1}^{2} \frac{(2u^T v)^i}{i!} = K_1(u,v) + K_2(u,v)$ is a valid kernel

Consider for some $j \in \mathbb{N}$, $K_j(u,v) = \sum_{i=1}^{j} \frac{(2u^T v)^i}{i!}$ as a valid kernel

Proving that $K_{j+1}(u,v)$ is a valid kernel using the above step,

$K_{j+1}(u,v) = \sum_{i=1}^{j+1} \frac{(2u^T v)^i}{i!} = K_j(u,v) + \frac{(2u^T v)^{j+1}}{(j+1)!}$ is a valid kernel (sum of two kernels)

Therefore, using the principle of mathematical induction, $K_j(u,v)$ is a valid kernel $\forall j \in \mathbb{N}$

Therefore, using the result for $j = \infty$, we can show that $\sum_{i=0}^{\infty} \frac{\left(2u^T v\right)^i}{i!}$ is a valid kernel.

Therefore, $e^{(2u^T v)} = \sum_{i=0}^{\infty} \frac{\left(2u^T v\right)^i}{i!}$ is a valid kernel.

(e) (2 points) Show that a Kernel given by $K(u,v) = \exp(-\|u-v\|^2)$ is a valid kernel. [Hint: Use last two parts' results.]

Solution:
Proof:
Firstly, simplifying the parameters of the exponential,

$\|u-v\|^2 =< (u-v), (u-v) >$ (Where <a,b> is the inner product between a and b)

$\Rightarrow < u,u > + < v,v > -2 < u,v >$

$\Rightarrow \|u\|^2 + \|v\|^2 - 2u^T v$ ($< a,b >= a^T b$)

Employing this result,the function can be expressed as a product of three exponentials.

$$e^{-||u-v||^2} = e^{-||u||^2} \cdot e^{2u^T v} \cdot e^{-||v||^2}$$

In the previous subdivision, we saw that $K(u,v) = e^{2u^T v}$ is a valid kernel function. Therefore, by invoking the result derived in part c and choosing $f(u) = e^{-||u||^2}$, $f(u) \cdot K(u,v) \cdot f(v)$ is a valid kernel.
Therefore,$K(u,v) = \exp(-||u-v||^2)$ is a valid kernel.

4. (10 points) [YESS, VEE, EMM...] Consider a soft margin SVM with the regularization parameter C. For the dataset $\{x_i, y_i\}_{i=1}^n$ such that $x_i \in \mathbb{R}^d$ and $y_i \in \{-1,+1\}$, let $\alpha^* \in \mathbb{R}^n$ be the optimal dual solution and $w^* \in \mathbb{R}^d, b^* \in \mathbb{R}, \epsilon^* \in \mathbb{R}^n$ be the optimal primal solution.

   (a) (3 points) Compute the range of α_i. Say each $\alpha_i \in [l_i, u_i]$. Compute the range of $w^{*T} x_i + b^*$ when 1) $\alpha_i = l_i$, 2) $\alpha_i = u_i$, or 3) $l_i < \alpha_i < u_i$.

Solution:
Computing the range of α_i:
The optimal primal solution of the SVM satisfies the below equation.

$$\min_{w,b,\epsilon} \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{n} \epsilon_i \text{ subject to } \epsilon_i \geqslant 0, y_i \left(w^T x_i + b\right) \geqslant (1 - \epsilon_i) \tag{17}$$

Where $\frac{1}{2} \parallel w \parallel^2$ is the term that represents maximizing the margin and the term $C \sum_{i=1}^{n} \epsilon_i$ incorporates misclassifications.
Given this primal problem, computing the equivalent dual problem through Lagrangian multipliers, we have,

$$\phi(\alpha, \beta) = \min_{w,b,\epsilon} \mathcal{L}(w, b, \epsilon, \alpha, \beta)$$
$$\Rightarrow \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{n} \epsilon_i + \sum_{i=1}^{n} \alpha_i \left(1 - \epsilon_i - y_i \left(w^T x_i + b\right)\right) + - \sum_{i=1}^{n} \beta_i \epsilon_i$$

As we don't want $\phi(\alpha, \beta)$ to go all the way upto $-\infty$ we can impose the following constraints on the values of α and β.

$$\sum_{i=1}^{n} - \alpha_i y_i = 0$$
$$C = \alpha_i + \beta_i$$

Now, as the lagrangian multiplier $\beta_i \geqslant 0$, we have $C - \alpha_i \geqslant 0$. Using this with the fact that $\alpha_i \geqslant 0$, we obtain the range of α_i to be, $0 \leqslant \alpha_i \leqslant C$. Therefore, $l_i = 0$ and $u_i = C$.
Computing the range of $w^{*T}x_i + b^*$:

- **When $\alpha_i = 0$:**
  When $\alpha_i = 0$, $\beta_i = C$ and from the KKT conditions, we have, $y_i\left(w^T x_i + b\right) > (1 - \epsilon_i)$ and $\epsilon_i = 0$. Therefore we have the optimal primal solution satisfying $y_i\left(w^{*T}x_i + b^*\right) > 1$. This can be visualized as $\left(w^{*T}x_i + b^*\right)$ lying outside the margin hyperplanes.
  Therefore the range of $\left(w^{*T}x_i + b^*\right)$ satisfying the above condition is $\left(w^{*T}x_i + b^*\right) \in (-\infty, -1) \cup (1, \infty)$.

- **When $\alpha_i = C$:**
  When $\alpha_i = C$, $\beta_i = 0$ and from the KKT conditions, we have, $y_i\left(w^T x_i + b\right) = (1 - \epsilon_i)$ and $\epsilon_i > 0$. Therefore we have the optimal primal solution satisfying $y_i\left(w^{*T}x_i + b^*\right) = 1 - \epsilon_i^*$. This can be visualized as $\left(w^{*T}x_i + b^*\right)$ lying on the wrong side of the margin hyperplanes with a *slack* of ϵ_i^*.
  Therefore the value of $\left(w^{*T}x_i + b^*\right)$ satisfying the above condition is

$$\left(w^{*T}x_i + b^*\right) = \begin{cases} 1 - \epsilon_i^* & \text{if } y_i = 1 \\ -1 + \epsilon_i^* & \text{if } y_i = -1 \end{cases}$$

- **When $0 < \alpha_i < C$:**
  Here $\alpha_i \neq 0$, $\beta_i \neq 0$ and from the KKT conditions, we have, $y_i\left(w^T x_i + b\right) = (1 - \epsilon_i)$ and $\epsilon_i = 0$. Therefore we have the optimal primal solution satisfying $y_i\left(w^{*T}x_i + b^*\right) = 1$. This can be visualized as $\left(w^{*T}x_i + b^*\right)$ lying on the margin hyperplanes.
  Therefore the value of $\left(w^{*T}x_i + b^*\right)$ satisfying the above condition is

$$\left(w^{*T}x_i + b^*\right) = \begin{cases} 1 & \text{if } y_i = 1 \\ -1 & \text{if } y_i = -1 \end{cases}$$

(b) (3 points) Explain what would happen if C is 1) negative, 2) zero, or 3) infinity.

Solution:
Different cases for C:
As we saw in the previous subdivision, C is the regularization constant that penalizes the slack term ϵ.

- **$C < 0$:** If C is negative, it actually encourages slack. Therefore, this will lead to a very

poor classification where several points end up being misclassified.

- **C = 0:** From the expression in 17, a small value of C indicates a small value for $\| w \|$ thereby a large margin as margin $\propto \frac{1}{\|w\|}$.Therefore, this prioritizes a larger margin over the corresponding misclassification encountered in some sense. When C = 0, $\| w \| = 0$, meaning the margin hyperplanes are at ∞. This means, that there is no classification made as all the datapoints lie within the margin.
- **C = ∞:** From the expression in 17, a large value of C indicates a large value for $\| w \|$ thereby a small margin as margin $\propto \frac{1}{\|w\|}$. Therefore, this prioritizes a small margin in a bid to prevent any misclassification. When C = ∞, any misclassification whatsoever blows up the error. This means that $\epsilon^* = 0$ and the problem reduces to a hard margin SVM.

(c) (4 points) Construct a Kernel SVM for the XOR function with inputs x_1 and x_2 where $x_1, x_2 \in \{0, 1\}$. Visualize the decision boundary back in Euclidean space. Would it be possible to construct the function without using a kernel? If yes, how? If not, why?

Solution:
Kernel SVM and decision boundary:
First let us have a look at the XOR function.

XOR function

x_1	x_2	$x_1 \bigoplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

To simplify proceedings, we can do the variable transformation $X_i = 2(x_i - 0.5)$.Assigning class labels, the classification dataset is given as,

XOR classification

v	x_1	x_2	$x_1 \bigoplus x_2$	X_1	X_2	y
v_1	0	0	0	-1	-1	-1
v_2	0	1	1	-1	1	1
v_3	1	0	1	1	-1	1
v_4	1	1	0	1	1	-1

Let us choose the polynomial kernel, $K(u, v) = \left(1 + u^T v\right)^2$.
The corresponding feature mapping is as follows, $\phi(u) = \left[1, u_1^2, \sqrt{2}u_1u_2, u_2^2, \sqrt{2}u_1, \sqrt{2}u_2\right]$

where u_i represents the i^{th} component of u and 1 is for the bias term.
Computing the Kernel Matrix K and Y, a diagonal matrix with entry Y_{ii} denoting the class of vector v_i for the vectors v_1, v_2, v_3, v_4 in the transformed coordinate space, we have,

$$K = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

$$Y = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

We know that the optimal solution weights are given by $w^* = \phi(X)^T Y \alpha^*$ where each row of $\phi(X) = \phi(v_i)$ Substituting this back into the objective function to be maximized, we get the below Dual SVM problem.

$$J(\alpha) = \max_{\alpha} \frac{-1}{2} \alpha^T Y K Y \alpha + \sum_{i=1}^{4} \alpha_i \tag{18}$$

Plugging in the values for K and Y, we get,

$$J(\alpha) = \max_{\alpha} \sum_{i=1}^{4} \alpha_i - \frac{1}{2} \left(\sum_{i=1}^{4} 9\alpha_i^2 + \sum_{i=1}^{4} \sum_{j=i+1}^{4} 2\alpha_i \alpha_j Y_{ii} Y_{jj} \right)$$

Taking the gradient of $J_(\alpha)$, we can find out the maximizing α^*.

$$\frac{\partial J}{\alpha_1} = 9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 - 1 = 0$$
$$\frac{\partial J}{\alpha_2} = 9\alpha_2 - \alpha_1 - \alpha_4 + \alpha_3 - 1 = 0$$
$$\frac{\partial J}{\alpha_3} = 9\alpha_3 - \alpha_1 - \alpha_4 + \alpha_2 - 1 = 0$$
$$\frac{\partial J}{\alpha_1} = 9\alpha_4 - \alpha_2 - \alpha_3 + \alpha_1 - 1 = 0$$

Solving these simultaneous equations, we arrive at $\alpha^* = \begin{bmatrix} \frac{1}{8} \\ \frac{1}{8} \\ \frac{1}{8} \\ \frac{1}{8} \end{bmatrix}$.

Computing w^* using this, we have $w^* = \begin{bmatrix} 0 \\ 0 \\ \frac{-1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \end{bmatrix}$

We can see that the bias term, $w_0 = 0$, therefore the decision boundary hyperplane is given by $w^{*T}\phi v = 0$ which is $-v_1 v_2 = 0$. Here, v_i is nothing but X_i.
Transforming the axes back to our original set of coordinates, we get the decision boundary hyperplane as $-(x_1 - 0.5)(x_2 - 0.5) = 0$. This leads to the pair of straight lines $x_2 = 0.5$ and $x_1 = 0.5$. This is shown in the figure below,

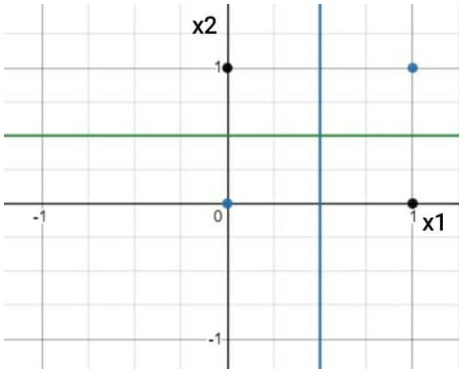

Figure 3: The decision boundary for the XOR problem

So, for values in which $-(x_1 - 0.5)(x_2 - 0.5) > 0$ such as the black datapoints, it gets classified to a class of similar sign, ie, $y = 1$ which represents the output 1. Similarly, for values in which $-(x_1 - 0.5)(x_2 - 0.5) < 0$ such as the blue datapoints, it gets classified to $y = -1$ which represents the output 0. Hence, the decision boundary separates the datapoints from different classes appropriately.

Possibility of recreating the function without a Kernel:

No, this function cannot be recreated without implementing a Kernel.In the original dimensions, the dataset appears as though it is not linearly separable, i.e, we can't have a linear hyperplane decision boundary. We implement a Kernel to take these datapoints to a higher dimensional space where it is linearly separable and then solve for the decision

boundary hyperplane in those higher dimensions.

(d) (3 points) [OPTIONAL BONUS] Consider an n dimensional input dataset X with m tuples. Derive an expression in terms of a Kernel to calculate the average distance of center of mass from each $\phi(x_i)$ in the feature space, where center of mass is just the average of all $\phi(x_i)$ where $x_i \in X$.

Solution:
Average distance to the center of mass in terms of a Kernel:
Here I am assuming that the distance is in the form of squared norm.
Let us denote the center of mass by $\bar{\phi}(X)$.

$$\bar{\phi}(X) = \frac{1}{m}\sum_{i=1}^{m} \phi(x_i) \tag{19}$$

Then the given problem can be formulated as finding a simplified kernel expression for $\frac{1}{m}\sum_{i=1}^{m} \| \phi(x_i) - \bar{\phi}(X) \|^2$ as this depicts the average distance from the center of mass.

$$\frac{1}{m}\sum_{i=1}^{m} \| \phi(x_i) - \bar{\phi}(X) \|^2 = \frac{1}{m}\sum_{i=1}^{m} \left\langle \phi(x_i) - \bar{\phi}(X), \phi(x_i) - \bar{\phi}(X) \right\rangle$$

Splitting the inner products,

$$\Rightarrow \frac{1}{m}\sum_{i=1}^{m} \left(\langle \phi(x_i), \phi(x_i) \rangle + \left\langle \bar{\phi}(X), \bar{\phi}(X) \right\rangle - 2 \left\langle \bar{\phi}(X), \phi(x_i) \right\rangle \right)$$

$$\Rightarrow \frac{1}{m}\sum_{i=1}^{m} \langle \phi(x_i), \phi(x_i) \rangle + \frac{1}{m}\sum_{i=1}^{m} \left\langle \bar{\phi}(X), \bar{\phi}(X) \right\rangle - \frac{2}{m}\sum_{i=1}^{m} \left\langle \bar{\phi}(X), \phi(x_i) \right\rangle$$

There are three distinct inner products in the above problem.We can try expressing them in terms of elements of a Kernel Matrix K, where $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$.

- $\frac{1}{m}\sum_{i=1}^{m} \langle \phi(x_i), \phi(x_i) \rangle = \frac{1}{m}\sum_{i=1}^{m} K_{ii} = \frac{1}{m}\text{trace}(K)$.

- $\frac{1}{m}\sum_{i=1}^{m}\left\langle \bar{\phi}(X), \bar{\phi}(X)\right\rangle = \left\langle \bar{\phi}(X), \bar{\phi}(X)\right\rangle$ as it is independent of i.

$$\left\langle \bar{\phi}(X), \bar{\phi}(X)\right\rangle = \left\langle \frac{1}{m}\sum_{i=1}^{m}\phi(x_i), \frac{1}{m}\sum_{i=1}^{m}\phi(x_i)\right\rangle \qquad \text{(From 19)}$$

$$\Rightarrow \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left\langle \phi(x_i), \phi(x_j)\right\rangle \qquad \text{(Linearity of inner product)}$$

$$\Rightarrow \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} K_{ij} = \bar{K}$$

Where $\bar{K}$ is the average of all the entries of the Kernel Matrix.

- $-\frac{2}{m}\sum_{i=1}^{m}\left\langle \bar{\phi}(X), \phi(x_i)\right\rangle$

$$-\frac{2}{m}\sum_{i=1}^{m}\left\langle \bar{\phi}(X), \phi(x_i)\right\rangle = -\frac{2}{m}\sum_{i=1}^{m}\left\langle \frac{1}{m}\sum_{i=1}^{m}\phi(x_i), \phi(x_i)\right\rangle \qquad \text{(From 19)}$$

$$\Rightarrow -\frac{2}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left\langle \phi(x_j), \phi(x_i)\right\rangle \qquad \text{(Linearity of inner product)}$$

$$\Rightarrow -\frac{2}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} K_{ij}$$

$$\Rightarrow -2\bar{K}$$

Therefore, the required expression is nothing but the sum of these three expressions,i.e, $\frac{1}{m}\text{trace}(K) - \bar{K}$.
This is an extremely intuitive expression since it somewhat resembles the expression of variance, i.e, $\mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2$. This is because we indeed calculated variance in the feature space which is nothing but the average distance from the center mass for each point.

Therefore, the average distance of each point from the center of mass in the feature space can be expressed in terms of a Kernel matrix K as $\frac{1}{m}\text{trace}(K) - \bar{K}$ where $\bar{K}$ is the average across all entries of K.

5. (10 points) [SVM LIFE IN HIGHER DIMENSIONS] Let's learn some SVM models in this question. You are required to choose the best hyperparameters for three kernel types, and may use `sklearn.svm` for this purpose (using a part of the training data as a validation set). You will report the best regularisation parameter for each kernel type, and the overall best among these three kernel types:

   1. Linear,
   2. RBF, and
   3. Polynomial kernels.

   Note: Linear Kernel has no kernel parameter.

   Please use the template `.ipynb` file in this folder to prepare your solution. Run your SVM algorithm on the classification datasets A,B,C in this folder; report the training and test zero-one error for the different hyperparameter settings; and illustrate the learned classifier for each kernel type for datasets A,B.

   Provide your results/answers in the pdf file you upload to GradeScope, and submit your code separately in this moodle link. The code submitted should be a `rollno_name.zip` file containing two files: `rollno_name.ipynb` file (including your code as well as the exact same results/plots uploaded to Gradescope) and the associated `rollno_name.py` file.

   Please note that you are not allowed to add any import statements in the code (given template). All the required libraries have already been added to the template.

   (a) (3 points) Plot the decision boundary of SVM (3 learned kernels on 2 datasets {A,B} = 6 plots) using a 2D plot similar to Fig 4.5 in Bishop's book (i.e., add the training data points to the plots; color the positively classified area light green and negatively classified area light red, etc.).

   Solution:
   Plots for the classified datasets:

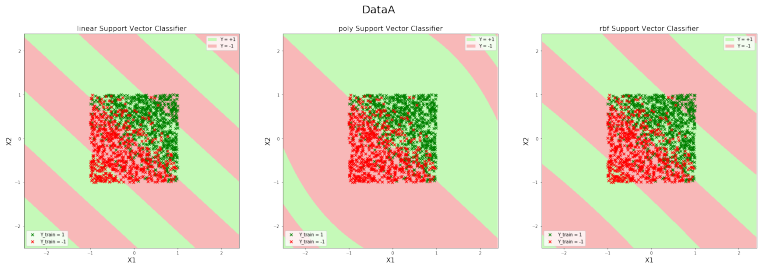

Figure 4: The plots of the decision boundary with the training data for Dataset A

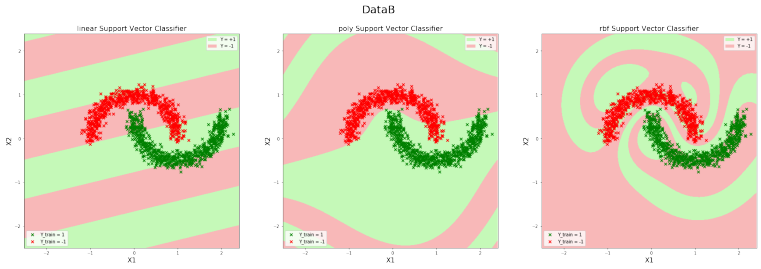

Figure 5: The plots of the decision boundary with the training data for Dataset B

(b) (4 points) For all the three datasets {A,B,C}, provide the training and test zero-one error rates for all three kernels, along with various choices of hyperparameters (in tables with appropriately named rows and columns). Also highlight your optimal choice of hyperparameters.

Solution:
Training and testing zero-one error rates:
The datasets were first subject to a grid search with the values $C = [0.1, 1, 10, 100, 1000]$ and $\gamma = [1, 0.1, 0.01, 0.001, 0.0001]$. After this, finding out the optimal hyperparameters, a second order testing was done with other hyperparameters in the vicinity. For simplicity, only the results of the second order testing is shown.

The Zero-One error rates

Dataset	kernel type	C	γ	train error	test error
A	linear	10	-	0.177	0.208
A	linear	50	-	0.177	0.208
A	poly	10	1	0.175	0.206
A	poly	10	2	0.176	0.202
A	poly	50	1	0.176	0.202
A	poly	50	2	0.176	0.202
A	rbf	1	0.01	0.174	0.192
A	rbf	1	0.02	0.175	0.192
A	rbf	5	0.01	0.177	0.202
A	rbf	5	0.02	0.174	0.204
B	linear	10	-	0.129	0.134
B	linear	50	-	0.129	0.132
B	poly	0.1	1	0.061	0.082
B	poly	0.1	2	0.062	0.084
B	poly	0.5	1	0.06	0.082
B	poly	0.5	2	0.059	0.08
B	rbf	100	1	0.0	0.002
B	rbf	100	2	0.0	0.002
B	rbf	500	1	0.0	0.002
B	rbf	500	2	0.0	0.002

Optimal Hyperparameters:
From the above table, we can see that the errors are minimized for the below combinations of the hyperparameter values. $C_{\text{optimal}} = \{A : \{'linear' : 10, 'poly' : 10, 'rbf' : 1\}, B : \{'linear' : 50, 'poly' : 0.5, 'rbf' : 100\}\}$
$\gamma_{\text{optimal}} = \{A : \{'linear' : 0, 'poly' : 2, 'rbf' : 0.01\}, B : \{'linear' : 0, 'poly' : 2, 'rbf' : 1\}\}$

(c) (3 points) Summarise and explain your observations based on your plots and the assumptions given in the problem.

Solution:
Inferences and Observations:

- From this we can see that the rbf kernel performs the best followed by the polynomial and then the linear kernel for any given dataset and it is intuitive to understand why. The rbf kernel, expands the input data to an infinite dimensional feature space which ensures that a better fit can be found.
- Comparing datasets A and B, we can see that there is not much improvement in the

rbf and the polynomial kernels when compared to the linear one in dataset A. This implies that the dataset is characteristically like that and no matter how many dimensions we expand it to, it will not get uncluttered. On the other hand, the rbf and polynomial kernels perform magnitudes better than the linear one for dataset B which is because, the dataset becomes linearly separable at a higher order dimension.

- The rbf kernel has a very low error for dataset B. This is because it makes the soft margin SVM problem into a hard margin one by taking a large regularization value.Again, since we are taking infinite dimensions, the dataset becomes linearly separable which paves the way for this to happen.