IFT-6390 Fundamentals of Machine Learning
Professor: Ioannis Mitliagkas

Homework 2 - Theoretical part

1. **Bias-Variance decomposition** [2 points]

   Consider the following data generation process: an input point x is drawn from an unknown distribution and the output y is generated using the formula
   $$y = f(x) + \epsilon,$$
   where f is an unknown deterministic function and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. This process implicitly defines a distribution over inputs and outputs; we denote this distribution by p.

   Given an i.i.d. training dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn from p, we can fit the hypothesis h_D that minimizes the empirical risk with the squared error loss function. More formally,
   $$h_D = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} (y_i - h(x_i))^2$$
   where $\mathcal{H}$ is the set of hypotheses (or function class) in which we look for the best hypothesis/function.

   The expected error[1] of h_D on a fixed data point (x', y') is given by $\mathbb{E}[(h_D(x') - y')^2]$. Two meaningful terms that can be defined are:

   - The bias, which is the difference between the expected value of hypotheses at x' and the true value $f(x')$. Formally,
   $$\mathit{bias} = \mathbb{E}[h_D(x')] - f(x')$$
   - The variance, which is how far hypotheses learned on different datasets are spread out from their mean $\mathbb{E}[h_D(x')]$. Formally,
   $$\mathit{variance} = \mathbb{E}[(h_D(x') - \mathbb{E}[h_D(x')])^2]$$

[1] Here the expectation is over random draws of the training set D of n points from the unknown distribution p. For example (and more formally): $\mathbb{E}[(h_D(x')] = \mathbb{E}_{(x_1,y_1)\sim p} \cdots \mathbb{E}_{(x_n,y_n)\sim p} \mathbb{E}[(h_{\{(x_1,y_1),\ldots,(x_n,y_n)\}}(x')]$.

Show that the expected prediction error on (x', y') can be decomposed into a sum of 3 terms: $(bias)^2$, *variance*, and a *noise* term involving ϵ. You need to justify all the steps in your derivation.

Answer. For ease of reading, we will use x instead of x' and y instead of y', and $h(x)$ instead of $h_D(x)$

$\mathbb{E}[(h(x) - y)^2]$

expanding the ^{2}

$= \mathbb{E}[h(x)^2 - 2yh(x) + y^2]$

using linearity of expectations

$= \mathbb{E}[h(x)^2] - 2\,\mathbb{E}[yh(x)] + \mathbb{E}[y^2]$

add and subtract the term $E[h(x)]^2$

$= \mathbb{E}[h(x)^2] - 2\,\mathbb{E}[yh(x)] + \mathbb{E}[y^2] + E[h(x)]^2 - E[h(x)]^2$

using $Var(h(x)) = \mathbb{E}[(h(x) - \mathbb{E}[h(x)])^2]$

$= Var(h(x)) - 2\,\mathbb{E}[yh(x)] + \mathbb{E}[y^2] + E[h(x)]^2$

expand $y = f(x) + \epsilon$

$= Var(h(x)) - 2\,\mathbb{E}[(f(x) + \epsilon)h(x)] + \mathbb{E}[(f(x) + \epsilon)^2] + E[h(x)]^2$

expand $(f(x) + \epsilon)^2$

$= Var(h(x)) - 2\,\mathbb{E}[(f(x) + \epsilon)h(x)] + \mathbb{E}[f(x)^2 + 2f(x)\epsilon + \epsilon^2] + E[h(x)]^2$

by linearity of expectations

$= Var(h(x)) - 2\,\mathbb{E}[(f(x) + \epsilon)h(x)] + \mathbb{E}[f(x)^2] + \mathbb{E}[2f(x)\epsilon] + \mathbb{E}[\epsilon^2] + E[h(x)]^2$

tricky part here $\mathbb{E}[2f(x)\epsilon] = 2 * (\mathbb{E}[f(x)]\,\mathbb{E}[\epsilon] + Covariance(f(x), \epsilon))$ and we can assume that $Covariance(f(x), \epsilon) = 0$ since the noise ϵ is random, and we know that $\mathbb{E}[\epsilon] = \mathbb{E}[\mathcal{N}(0, \sigma^2)] = 0$ so $\mathbb{E}[2f(x)\epsilon] = 0$

$= Var(h(x)) - 2\,\mathbb{E}[(f(x) + \epsilon)h(x)] + \mathbb{E}[f(x)^2] + \mathbb{E}[\epsilon^2] + E[h(x)]^2$

expand $(f(x) + \epsilon)h(x)$ and again use linearity of expectations

$$= Var(h(x)) - 2\,\mathbb{E}[f(x)h(x)] + 2\,\mathbb{E}[\epsilon h(x)] + \mathbb{E}[f(x)^2] + \mathbb{E}[\epsilon^2] + E[h(x)]^2$$

use the same trick as before to show that $2\,\mathbb{E}[\epsilon h(x)] = 0$

$$= Var(h(x)) - 2\,\mathbb{E}[f(x)h(x)] + \mathbb{E}[f(x)^2] + \mathbb{E}[\epsilon^2] + E[h(x)]^2$$

using $\mathbb{E}[f(x)^2] = f(x)^2$

$$= Var(h(x)) - 2\,\mathbb{E}[f(x)h(x)] + f(x)^2 + \mathbb{E}[\epsilon^2] + E[h(x)]^2$$

again tricky $\mathbb{E}[f(x)h(x)] = \mathbb{E}[f(x)]\,\mathbb{E}[h(x)] + Covariance(f(x), h(x))$ we can again assume $Covariance = 0$ so $\mathbb{E}[f(x)h(x)] = f(x)\,\mathbb{E}[h(x)]$

$$= Var(h(x)) - 2f(x)\,\mathbb{E}[h(x)] + f(x)^2 + \mathbb{E}[\epsilon^2] + E[h(x)]^2$$

using $Bias(h(x))^2 = \mathbb{E}[h(x)]^2 - 2f(x)\,\mathbb{E}[h(x)] + f(x)^2$

$$= Var(h(x)) - Bias(h(x))^2 + \mathbb{E}[\epsilon^2]$$

we can subtract $\mathbb{E}[\epsilon]^2$ since it is 0 because $\mathbb{E}[\epsilon] = 0$

$$= Var(h(x)) - Bias(h(x))^2 + \mathbb{E}[\epsilon^2] - \mathbb{E}[\epsilon]^2$$

using $Var(\epsilon) = \mathbb{E}[\epsilon^2] - \mathbb{E}[\epsilon]^2$

$$= Var(h(x)) - Bias(h(x))^2 + Var(\epsilon)$$

since $\epsilon \sim \mathcal{N}(0, \sigma^2)$ we know its variance is σ^2

$$= Var(h(x)) - Bias(h(x))^2 + \sigma^2$$

2. **Optimization** [10 points] Assume a 1D logistic function:

$$\sigma(wx) = \frac{1}{1 + e^{-wx}}$$

where $x, w \in \mathbb{R}$, and the associated cost function:

$$L\left(w\right) = -y \log \sigma(wx) - (1 - y) \log\left(1 - \sigma(wx)\right)$$

(a) Show that the cost function associated with logistic regression is convex. You can use one of the following two definitions of convexity:

- $L(w)$ is convex if and only if

  $$\forall w_1, w_2, t \in [0, 1] : L\left(tw_1 + (1 - t)w_2\right) \leq tL\left(w_1\right) + (1 - t)L\left(w_2\right)$$

- L is convex if and only if $\frac{d^2 L}{dw^2}(w) \geq 0$ for all w

You can also use another definition of convexity but you have to explicitly state it.

(b) Find the gradient of $\sigma(wx)$ at some point w. What are the dimensions of the gradient?

(c) Find all of the stationary points of $L(w)$ with respect to w analytically (Justify).

(d) Show one step of gradient descent from w_0 to w_1, using the gradient of the cost function mentioned above.

Answer.

(a)

$$L\left(w\right) = -y \log \sigma(wx) - (1 - y) \log\left(1 - \sigma(wx)\right)$$

let's simplify the expression starting with the first term

$$\begin{aligned} -y \log \sigma(wx) &= -y \log \frac{1}{1 + e^{-wx}} \\ &= -y[\log 1 - \log(1 + e^{-wx})] \\ &= y \log(1 + e^{-wx}) \end{aligned}$$

now the second term

$$\begin{aligned}
-(1-y)\log\left(1-\sigma(wx)\right) &= -(1-y)\log 1-\frac{1}{1+e^{-wx}}\\
&= -(1-y)\log\frac{1+e^{-wx}}{1+e^{-wx}}-\frac{1}{1+e^{-wx}}\\
&= -(1-y)\log\frac{e^{-wx}}{1+e^{-wx}}\\
&= -(1-y)[\log e^{-wx}-\log\left(1+e^{-wx}\right)]\\
&= -(1-y)[-wx-\log\left(1+e^{-wx}\right)]\\
&= (1-y)[wx+\log\left(1+e^{-wx}\right)]
\end{aligned}$$

now combining those terms

$$\begin{aligned}
L\left(w\right) &= y\log(1+e^{-wx})+(1-y)[wx+\log(1+e^{-wx})]\\
&= (1-y)wx+\log(1+e^{-wx})
\end{aligned}$$

now let’s take the second derivative to show convexity

start with the first derivative

$$\begin{aligned}
\frac{dL}{dw} &= (1-y)x-\frac{xe^{-wx}}{1+e^{-wx}}\\
&= (1-y)x-\frac{(1+e^{-wx})x-x}{1+e^{-wx}}\\
&= (1-y)x-x+\frac{x}{1+e^{-wx}}\\
&= -yx+\frac{x}{1+e^{-wx}}\\
&= -yx+x\sigma(wx)
\end{aligned}$$

now we do the second derivative using our knowledge of derivative of a sigmoid

$$\begin{aligned}
\frac{d^2L}{dw^2} &= x\sigma(wx)(1-\sigma(wx))x\\
&= x^2\sigma(wx)(1-\sigma(wx))
\end{aligned}$$

Since $\sigma() \in (0,1)$, therefore $\sigma(wx) \in (0,1)$ and $(1-\sigma(wx)) \in (0,1)$. Finally, $x^2 > 0$ so each of our three terms > 0 therefore the product $\frac{d^2L}{dw^2} > 0$ and this loss is convex.

(b)

$$\begin{aligned}
\frac{d}{dw}\sigma(wx) &= \frac{d}{dw}\frac{1}{1+e^{-wx}} \\
&= \frac{d}{dw}(1+e^{-wx})^{-1} \\
&= -(1+e^{-wx})^{-2} * \frac{d}{dw}(1+e^{-wx}) \\
&= -(1+e^{-wx})^{-2} * e^{-wx} * -x \\
&= \frac{xe^{-wx}}{(1+e^{-wx})^2} \\
&= x\frac{1}{1+e^{-wx}}\frac{e^{-wx}}{1+e^{-wx}} \\
&= x\frac{1}{1+e^{-wx}}(1-\frac{1}{1+e^{-wx}}) \\
&= x\sigma(wx)(1-\sigma(wx))
\end{aligned}$$

The dimension of this is the same as the number of points represented by x. For a single point x this is just a scalar.

(c) Stationary points are where $\frac{dL(w)}{dw} = 0$. Using our formula from (a)

$$\begin{aligned}
\frac{dL(w)}{dw} &= 0 \\
-yx + x\sigma(wx) &= 0 \\
x\sigma(wx) &= yx
\end{aligned}$$

assume $x \neq 0$

$$\begin{aligned}
\sigma(wx) &= y \\
\frac{1}{1+e^{-wx}} &= y \\
e^{-wx} &= \frac{1}{y} - 1 \\
-wx &= \log(\frac{1-y}{y}) \\
w &= -\frac{1}{x}\log(\frac{1-y}{y})
\end{aligned}$$

To figure out possible values for w, let's see our possible values for y. If the example is positive, $y = 1$ and $w = -\frac{1}{x}\log 0$. This is undefined, but we can see that *in the limit*

$$\lim_{y \to 0+} \log y = -\infty$$

Therefore, *in the limit*, there is a stationary point

$$\begin{aligned} w &= -\frac{1}{x} * -\infty \\ &= \infty \end{aligned}$$

Similarly, for a negative example $y = 0$, we would expect the stationary point in the limit to be $-\infty$

(d) Using α as our learning rate

$$\begin{aligned} w_1 &= w_0 - \alpha * \frac{d}{dw} L(w) \\ &= w_0 - \alpha * (-yx + x\sigma(wx)) \\ &= w_0 + \alpha(yx - x\sigma(wx)) \end{aligned}$$

3. **Least Squares Estimator and Ridge Regression** [10 points]

We consider the problem of learning a vector-valued function $f : \mathbb{R}^d \to \mathbb{R}^p$ from input-output training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ where each $\mathbf{x}_i$ is a d-dimensional vector and each $\mathbf{y}_i$ is a p-dimensional vector. We choose our hypothesis class to be the set of linear functions from $\mathbb{R}^d$ to $\mathbb{R}^p$, that is function satisfying $f(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ for some $d \times p$ regression matrix $\mathbf{W}$, and we want to minimize the squared error loss function

$$J(\mathbf{W}) = \sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{W}^\top \mathbf{x}_i\|_2^2 \tag{1}$$

over the training data.

Let $\mathbf{W}^*$ be the minimizer of the empirical risk:

$$\mathbf{W}^* = \arg\min_{\mathbf{W} \in \mathbb{R}^{d \times p}} J(\mathbf{W}).$$

(a) Derive a closed-form solution for $\mathbf{W}^*$ as a function of the data matrices $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{Y} \in \mathbb{R}^{n \times p}$.
(hint: once you have expressed $J(\mathbf{W})$ as a function of $\mathbf{X}$ and $\mathbf{Y}$, you may find the matrix cookbook useful to compute gradients w.r.t. to the matrix $\mathbf{W}$)

Rigde regression

A variation of the least squares estimation problem known as ridge regression considers the following optimization problem:

$$\arg\min_{\mathbf{W}} J(\mathbf{W}) + \lambda \|\mathbf{W}\|_{\mathbf{F}}^{\mathbf{2}} \tag{2}$$

where $\lambda > 0$ is a regularization parameter. The regularizing term penalizes large components in $\mathbf{W}$ which causes the optimal $\mathbf{W}$ to have a smaller norm.

(b) Derive the solution of the ridge regression problem. Do we still have to worry about the invertibility of $\mathbf{X}^\top \mathbf{X}$?

(c) Explain why the ridge regression estimator is likely to be more robust to issues of high variance compared with the least squares estimator.

(d) How does the value of λ affect the bias and the variance of the estimator?

Answer.

(a)

$$\begin{aligned} 0 &= \frac{d}{dW} J(W) \\ &= \frac{d}{dW} ||Y - XW||_2^2 \end{aligned}$$

using the matrix cookbook

$$\begin{aligned} &= 2X^T(XW - Y) \\ &= X^T XW - X^T Y \\ X^T Y &= X^T XW \end{aligned}$$

if $X^T X$ is invertible, we can isolate

$$W = (X^T X)^{-1} X^T Y$$

(b)

$$\arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2$$

$$\begin{aligned}
\frac{d\left[\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2\right]}{d\boldsymbol{\theta}} &= \frac{d\left[(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) + \lambda\boldsymbol{\theta}^\top\boldsymbol{\theta}\right]}{d\boldsymbol{\theta}} \\
&= 2\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta} - 2\mathbf{X}^\top\mathbf{y} + 2\lambda\boldsymbol{\theta} \\
\left.\frac{d\left[\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2\right]}{d\boldsymbol{\theta}}\right|_{\hat{\theta}} &= 2\mathbf{X}^\top\mathbf{X}\hat{\boldsymbol{\theta}} - 2\mathbf{X}^\top\mathbf{y} + 2\lambda\hat{\boldsymbol{\theta}} \\
&= 0 \\
\text{So, } (\mathbf{X}^\top\mathbf{X} + \lambda I)\hat{\boldsymbol{\theta}} &= \mathbf{X}^\top\mathbf{y} \\
\text{And hence, } \hat{\boldsymbol{\theta}} &= (\mathbf{X}^\top\mathbf{X} + \lambda I)^{-1}\mathbf{X}^\top\mathbf{y}
\end{aligned}$$

We no longer require $\mathbf{X}^\top\mathbf{X}$ be invertible. Now, it is $\mathbf{X}^\top\mathbf{X} + \lambda I$ that must be invertible. Since $\mathbf{X}^\top\mathbf{X}$ is positive semidefinite, $\mathbf{X}^\top\mathbf{X} + \lambda I$ is positive definite (since $\lambda > 0$), and so is always invertible.

(c) The extra λ term serves to regularize the parameters found by the estimator. Since they are pushed toward zero by the regularization term, they are less susceptible to change due to changes in the sampled data.

Individual outliers due to data variance, for example, will have greater influence on the parameters estimated by the least squares estimator than on those estimated by the ridge regression estimator.

(d) As λ increases, the bias increases and the variance decreases. At $\lambda = 0$, the estimator has minimal bias and maximum variance, and as $\lambda \to \infty$ the bias increases and the variance tends toward 0.

4. **k-fold cross-validation** [10 points]

Let $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a training sample set drawn i.i.d. from an unknown distribution p. To estimate the risk (a.k.a. the test error) of a learning algorithm using D, k-fold cross validation (CV) involves using the ith fold of the data $D_i = \{(x_j, y_j) \mid j \in \text{ind}[i]\}$ (where ind$[i]$ are the indices of the data points in the ith fold) to evaluate the risk of the hypothesis returned by a learning algorithm trained on all the data except those in the ith fold, $D_{\setminus i} = \{(x_j, y_j) \mid j \notin \text{ind}[i]\}$.

Formally, if we denote the hypothesis returned by the learning algorithm trained on $D_{\setminus i}$ as $h_{D\setminus i}$, the k-fold CV error is given by

$$\text{error}_{k-fold} = \frac{1}{k}\sum_{i=1}^{k} \frac{1}{n/k} \sum_{j\in\text{ind}[i]} l(h_{D\setminus i}(x_j), y_j)$$

where l is the loss function.

In this exercise, we will investigate some interesting properties of this estimator.

k-fold is unbiased

(a) State the definition of the risk of a hypothesis h for a regression problem with the mean squared error loss function.

(b) Let D' denote a dataset of size $n - \frac{n}{k}$. Show that

$$\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{k-fold}] = \mathop{\mathbb{E}}_{\substack{D'\sim p,\\ (x,y)\sim p}}[(y - h_{D'}(x))^2]$$

where the notation $D \sim p$ means that D is drawn i.i.d. from the distribution p, h_D denotes the hypothesis returned by the learning algorithm trained on D. Explain how this shows that error_{k-fold} is an (almost) unbiased estimator of the risk of h_D.

Complexity of k-fold We will now consider k-fold in the context of linear regression where inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are d-dimensional vectors. Similarly to exercise 3, we use $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{y} \in \mathbb{R}^n$ to denote the input matrix and the vector of outputs.

(c) Assuming that the time complexity of inverting a matrix of size $m \times m$ is in $\mathcal{O}(m^3)$, what is the complexity of computing the solution of linear regression on the dataset D? (i.e. similar to the solution of 3 (a))

(d) Let $\mathbf{X}_{-i} \in \mathbb{R}^{(n-\frac{n}{k})\times d}$ and $\mathbf{y}_{-i} \in \mathbb{R}^{(n-\frac{n}{k})}$ be the data matrix and output vector obtained by removing the rows corresponding to the ith fold of the data. Using the formula for $error_{k-fold}$ mentioned at the start of this question, write down a formula of the k-fold CV error for linear regression. Specifically, substitute the loss expression with the actual loss obtained by using the analytical solution for linear regression. What is the complexity of evaluating this formula?

(e) It turns out that for the special case of linear regression, the k-fold validation error can be computed more efficiently. Show that in the case of linear regression we have

$$\text{error}_{k-fold} = \frac{1}{k}\sum_{i=1}^{k}\left(\frac{\mathbf{y}_i - \mathbf{X}_i\mathbf{w}^*}{\mathbf{I} - \mathbf{X}_i(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}_i^\top}\right)^2$$

where $\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$ is the solution of linear regression computed on the whole dataset D. What is the complexity of evaluating this formula?

Answer.

(a) $\boxed{\mathit{risk} = \sum_D (y - h(x))^2}$

(b)

$$\begin{aligned}
\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{k-fold}] &= \mathop{\mathbb{E}}_{D\sim p}\left[\frac{1}{k}\sum_{i=1}^{k}\frac{1}{n/k}\sum_{j\in\text{ind}[i]}\ell(h_{D\setminus\text{ind}[i]}(x_j), y_j)\right] \\
&= \frac{1}{k}\sum_{i=1}^{k}\mathop{\mathbb{E}}_{D\sim p}\left[\frac{1}{n/k}\sum_{j\in\text{ind}[i]}\ell(h_{D\setminus\text{ind}[i]}(x_j), y_j)\right] \\
&\approx \frac{1}{k}\sum_{i=1}^{k}\mathop{\mathbb{E}}_{\substack{D'\sim p,\\ (x,y)\sim p}}\left[\ell(h_{D'}(x), y)\right] \\
&= \mathop{\mathbb{E}}_{\substack{D'\sim p,\\ (x,y)\sim p}}\left[(y - h_{D'}(x))^2\right]
\end{aligned}$$

(c) Here we assume the complexity of multiplying an $a \times b$ matrix by a $b \times c$ matrix is $\mathcal{O}(abc)$.

Then calculating $\boldsymbol{\theta}^* = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}$ takes

$\mathcal{O}\left(dnd + d^3 + ddn + dn\right) = \boxed{\mathcal{O}\left(d^3 + d^2 n\right)}$ time.

(d)

$$\text{error}_{k-fold} = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n/k}\sum_{j\in\text{ind}[i]}\left(\mathbf{y}_i - \mathbf{X}_i\left[(\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1}\mathbf{X}_{-i}^\top \mathbf{y}_{-i}\right]\right)_j^2$$

Evaluating this takes

$$\begin{aligned}
&(\mathbf{X}_{-i}^\top \mathbf{X}_{-i}) \implies \mathcal{O}\left(d^2 n(1-\frac{1}{k})\right) \implies \mathcal{O}\left(d^2 n\right)\\
&(\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \implies \mathcal{O}\left(d^2 n + d^3\right)\\
&(\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1}\mathbf{X}_{-i}^\top \implies \mathcal{O}\left(d^2 n + d^3 + d^2 n(1-\frac{1}{k})\right) \implies \mathcal{O}\left(d^2 n + d^3\right)\\
&\mathbf{X}_i\left[(\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1}\mathbf{X}_{-i}^\top \mathbf{y}_{-i}\right] \implies \mathcal{O}\left(d^2 n + d^3 + dn(1-\frac{1}{k})\right) \implies \mathcal{O}\left(d^2 n + d^3\right)\\
&\frac{1}{n/k}\sum_{j\in\text{ind}[i]}(\mathbf{y}_i - \cdots)_j^2 \implies \mathcal{O}\left(d^2 n + d^3 + \frac{n}{k}\right) \implies \mathcal{O}\left(d^2 n + d^3\right)\\
&\text{error}_{k-fold} \implies \mathcal{O}\left(k\cdot(d^2 n + d^3)\right) \implies \mathcal{O}\left(d^2 nk + d^3 k\right)
\end{aligned}$$

(e)

$$\text{error}_{k-fold} = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n/k}\sum_{j\in\text{ind}[i]}\left(\mathbf{y}_i - \mathbf{X}_i\left[(\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1}\mathbf{X}_{-i}^\top \mathbf{y}_{-i}\right]\right)_j^2$$

$$
\begin{aligned}
&\mathbf{y}_i - \mathbf{X}_i \left[(\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i} \right] \\
&= \frac{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top}{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top} \cdot \left(\mathbf{y}_i - \mathbf{X}_i (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i} \right) \\
&= \boxed{\frac{\mathbf{y}_i - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top \mathbf{y}_i - \mathbf{X}_i (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i} + \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top \mathbf{X}_i (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i}}{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top}} \\
&= \boxed{\frac{\mathbf{y}_i - \mathbf{X}_i \left[(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top \mathbf{y}_i + (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i} - (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top \mathbf{X}_i (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i} \right]}{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top}} \\
&= \boxed{\frac{\mathbf{y}_i - \mathbf{X}_i \left[(\mathbf{X}^\top \mathbf{X})^{-1} \left(\mathbf{X}_i^\top \mathbf{y}_i + (\mathbf{X}^\top \mathbf{X}) (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i} - \mathbf{X}_i^\top \mathbf{X}_i (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i} \right) \right]}{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top}} \\
&= \frac{\mathbf{y}_i - \mathbf{X}_i \left[(\mathbf{X}^\top \mathbf{X})^{-1} \left(\mathbf{X}_i^\top \mathbf{y}_i + (\mathbf{X}^\top \mathbf{X} - \mathbf{X}_i^\top \mathbf{X}_i) (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i} \right) \right]}{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top} \\
&= \frac{\mathbf{y}_i - \mathbf{X}_i \left[(\mathbf{X}^\top \mathbf{X})^{-1} \left(\mathbf{X}_i^\top \mathbf{y}_i + (\mathbf{X}_{-i}^\top \mathbf{X}_{-i}) (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i} \right) \right]}{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top} \\
&= \frac{\mathbf{y}_i - \mathbf{X}_i \left[(\mathbf{X}^\top \mathbf{X})^{-1} \left(\mathbf{X}_i^\top \mathbf{y}_i + \mathbf{X}_{-i}^\top \mathbf{y}_{-i} \right) \right]}{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top} \\
&= \frac{\mathbf{y}_i - \mathbf{X}_i \left[(\mathbf{X}^\top \mathbf{X})^{-1} (\mathbf{X}^\top \mathbf{y}) \right]}{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top} \\
&= \frac{\mathbf{y}_i - \mathbf{X}_i \mathbf{w}^*}{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top}
\end{aligned}
$$

(Note: The boxes in the above formulas are artifacts of my formatting I have not been able to resolve.)

So,

$$
\begin{aligned}
\text{error}_{k-fold} &= \frac{1}{k} \sum_{i=1}^{k} \frac{1}{n/k} \sum_{j \in \text{ind}[i]} \left(\mathbf{y}_i - \mathbf{X}_i \left[(\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \mathbf{y}_{-i} \right] \right)_j^2 \\
&= \frac{1}{k} \sum_{i=1}^{k} \frac{1}{n/k} \sum_{j \in \text{ind}[i]} \left(\frac{\mathbf{y}_i - \mathbf{X}_i \mathbf{w}^*}{1 - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top} \right)_j^2
\end{aligned}
$$

Evaluating this formula takes

$$
\begin{aligned}
&(\mathbf{X}^\top\mathbf{X}) \implies \mathcal{O}\left(d^2n\right)\\
&(\mathbf{X}^\top\mathbf{X})^{-1} \implies \mathcal{O}\left(d^2n+d^3\right)\\
&\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \implies \mathcal{O}\left(d^2n+d^3+d^2n+dn\right) \implies \mathcal{O}\left(d^2n+d^3\right)\\
&(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}_i^\top \implies \mathcal{O}\left(d^2n+d^3+d^2\frac{n}{k}\right) \implies \mathcal{O}\left(d^2n+d^3\right)\\
&\mathbf{X}_i(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}_i^\top \implies \mathcal{O}\left(d^2n+d^3+d\left(\frac{n}{k}\right)^2\right) \implies \mathcal{O}\left(d^2n^2+d^3\right)\\
&\left(1-\mathbf{X}_i(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}_i^\top\right)^{-1} \implies \mathcal{O}\left(d^2n+d^3+\left(\frac{n}{k}\right)^3\right) \implies \mathcal{O}\left(d^2n+d^3+n^3\right)\\
&\frac{\mathbf{y}_i-\mathbf{X}_i\mathbf{w}^*}{1-\mathbf{X}_i(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}_i^\top} \implies \mathcal{O}\left(d^2n+d^3+\left(\frac{n}{k}\right)^3\right) \implies \mathcal{O}\left(d^2n+d^3+n^3\right)
\end{aligned}
$$

$\mathcal{O}\left(d^3+d^2k\right)$ to compute $(\mathbf{X}^\top\mathbf{X})^{-1}$ and $\mathbf{w}^*$, and $\mathcal{O}\left(d^2\right)$ per point for additional computation, so a total run time of $\boxed{\mathcal{O}\left(d^3+d^2n+n^3\right)}$, which is a factor of n better than in (d).

5. **Feature Maps** [8 points]

In this exercise, you will design feature maps to transform an original dataset into a linearly separable set of points. For the following questions, if your answer is '*yes*', write the expression for the proposed transformation; and if your answer is '*no*', write a brief explanation. You are expected to provide explicit formulas for the feature maps, and these formulas should only use common mathematical operations.

(a) [2 points] Consider the following 1-D dataset (Figure 1). Can you propose a 1-D transformation that will make the points linearly separable?

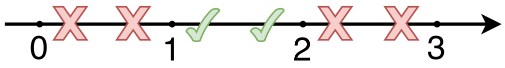

Figure 1: 1D dataset

(b) [2 points] Consider the following 2-D dataset (Figure 2). Can you propose a transformation into 1D that will make the data linearly separable?

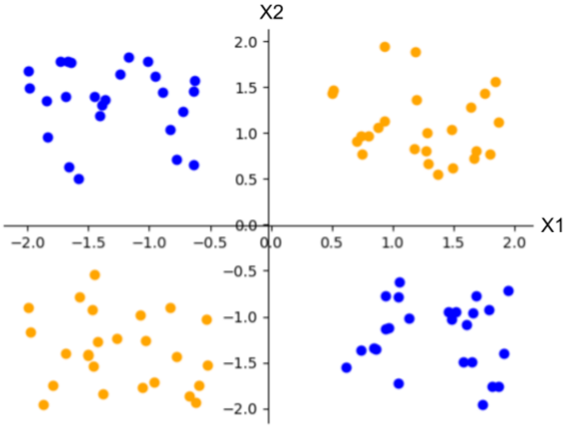

Figure 2: 2D dataset

(c) [4 points] Using ideas from the above two datasets, can you suggest a transformation of the following dataset (as shown in Figure 3) that makes it linearly separable? If '*yes*', also provide the kernel corresponding to the feature map you proposed. Remember that $K(x, y) = \phi(x) \cdot \phi(y)$, so find ϕ and do the dot product to get the kernel.

Answer.

(a) Yes

$\boxed{x' = (x - 1.5)^2}$

$\boxed{x' = (x - 2)(x - 1)}$

(b) Yes

$\boxed{x = X_1 * X_2}$

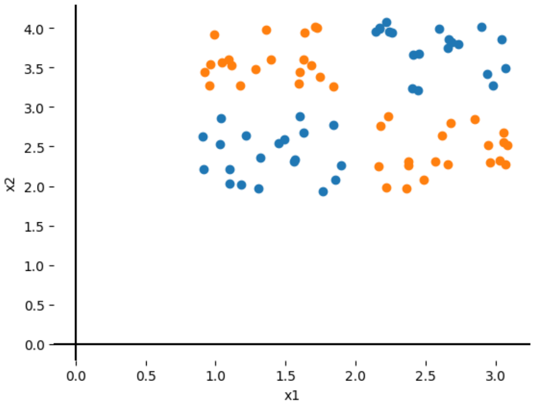

Figure 3: Another 2D dataset

(c) Yes $X' = (X_1 - 2) * (X_2 - 3)$ The corresponding kernel is

$$\begin{aligned}
K(X, Y) =& \phi(X)^T \phi(Y) \\
=& [(X_1 - 2) * (X_2 - 3)] * [(Y_1 - 2) * (Y_2 - 3)] \\
=& (X_1 X_2 - 3X_1 - 2X_2 + 6) * (Y_1 Y_2 - 3Y_1 - 2Y_2 + 6) \\
=& X_1 X_2 Y_1 Y_2 - 3X_1 X_2 Y_1 - 2X_1 X_2 Y_2 + 6X_1 X_2 - 3X_1 Y_1 Y_2 \\
& + 9X_1 Y_1 + 6X_1 Y_2 - 18X_1 - 2X_2 Y_1 Y_2 + 6X_2 Y_1 + 4X_2 Y_2 \\
& - 12X_2 + 6Y_1 Y_2 - 18Y1 - 12Y_2 + 36
\end{aligned}$$