IFT6390 HW2

Jonathan Lim (jonathan.siu.chi.lim@umontreal.ca)
Aasheesh Singh (aasheesh.singh@umontreal.ca)

26 November 2022

1. 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:

   $\mathbb{E}[(h_D(x^{'}) - y')^2]$
   $= \mathbb{E}[(h_D(x^{'}) - \mathbb{E}[h_D(x^{'})] + \mathbb{E}[h_D(x^{'})] - y^{'})^2]$
   $= \mathbb{E}[(h_D(x^{'}) - \mathbb{E}[h_D(x^{'})])^2 + (\mathbb{E}[h_D(x^{'})] - y^{'})^2 - 2(h_D(x^{'}) - \mathbb{E}[h_D(x^{'})])(\mathbb{E}[h_D(x^{'})] - y^{'})]$
   $= \mathbb{E}[(h_D(x^{'}) - \mathbb{E}[h_D(x^{'})])^2] + \mathbb{E}[(\mathbb{E}[h_D(x^{'})] - y^{'})^2] - 2\mathbb{E}[h_D(x^{'}) - \mathbb{E}[h_D(x^{'})]]\mathbb{E}[\mathbb{E}[h_D(x^{'})] - y^{'}]$
   Since $\mathbb{E}[h_D(x^{'}) - \mathbb{E}[h_D(x^{'})]] = \mathbb{E}[h_D(x^{'})] - \mathbb{E}[h_D(x^{'})] = 0$
   $\mathbb{E}[(h_D(x^{'}) - y')^2]$
   $= \mathbb{E}[(h_D(x^{'}) - \mathbb{E}[h_D(x^{'})])^2] + \mathbb{E}[(\mathbb{E}[h_D(x^{'})] - y^{'})^2]$
   $= \mathbb{E}[(h_D(x^{'}) - \mathbb{E}[h_D(x^{'})])^2] + \mathbb{E}[(\mathbb{E}[h_D(x^{'})])^2 - 2(f(x^{'}) + \epsilon)\mathbb{E}[h_D(x^{'})] + (f(x^{'}) + \epsilon)^2]$
   $= \mathbb{E}[(h_D(x^{'}) - \mathbb{E}[h_D(x^{'})])^2] + \mathbb{E}[\mathbb{E}[h_D(x^{'})]^2 - 2\mathbb{E}[h_D(x^{'}]f(x^{'}) + f(x^{i})^2] + \mathbb{E}[\epsilon^2] + 2\mathbb{E}[\epsilon]\mathbb{E}[f(x^{'}) - \mathbb{E}[h_D(x^{'}]]$
   $= \mathbb{E}[(h_D(x^{'}) - \mathbb{E}[h_D(x^{'})])^2] + (E[h_D(x^{'})] - f(x^{'}))^2$
   Since $\epsilon \sim \mathcal{N}(0, \sigma^2), \mathbb{E}[\epsilon] = 0$
   $\mathbb{E}[(h_D(x^{'}) - y')^2] = variance + bias^2 + \mathbb{E}[\epsilon^2]$

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

   Answer:

   Yes, the data can be linearly separable. Since the class labels are changing periodically we can use the below transformation to achieve linear separation:

   $$\phi(x) = \sin(\pi x)$$

and thus $\phi(x)$ would take the below values for any k $\in \mathbb{Z}$

$$\phi(x) = \begin{cases} > 0; & 2k < x < 2k+1 \\ < 0; & 2k+1 < x < 2k+2 \end{cases}$$

The line separating the two classes would be $\phi(x) = 0$.

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

Answer:

Yes, the following 1-D transformation would make the data linearly separable:

$$\phi(x) = x_1 x_2$$

$\phi(x)$ would vary as below according to values of x_1, x_2

$$\phi(x) = \begin{cases} > 0; & x_1, x_2 > 0 \quad \text{or} \quad x1, x2 < 0 \\ < 0; & x_1 > 0, x_2 < 0 \quad \text{or} \quad x1 < 0, x2 > 0 \end{cases}$$

The first case corresponds to points in (1st, 3rd) quadrant and second case corresponds to points in (2nd, 4th) quadrant. The line of separability will be $\phi(x) = 0$

(c) Using ideas from the above two datasets, can you suggest a transformation of the following dataset 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 you can find ϕ and do the dot product to find an expression for the kernel.

Answer:

The class labels are alternating between consecutive concentric circles, therefore we can transform our 2-D input features into 1-D by creating a new feature equivalent to the radial distance:

$$z = \sqrt{x_1^2 + x_2^2}$$

We can then define the transformation function as below:

$$\phi(x) : \mathbb{R}^2 \rightarrow \mathbb{R} = exp(-c \sin(\pi \sqrt{x_1^2 + x_2^2}))$$

where c is constant factor

The data is thus linearly separable after the transformation taking the below values for $\forall \quad k \in \mathbb{Z}$

$$\phi(x) = \begin{cases} < 0; & 2k < z < 2k + 1 \\ > 0; & 2k + 1 < z < 2k + 2 \end{cases}$$

The kernel K(x,y) can thus be expressed by the dot product of their individual transformations:

$$K(x, y) =< \phi(x).\phi(y) >$$

$$K(x, y) = exp(-c \sin(\pi\sqrt{x_1^2 + x_2^2}))exp(-c \sin(\pi\sqrt{y_1^2 + y_2^2}))$$

Arriving at the actual notation of the kernel involves computing the Taylor series expansion terms of the transformation function $\phi(x)$. Due to complexity, its left out of discussion in this problem.

Reference: Kernel Cookbook

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

Answer:

The expected or average squared difference between predicted $h(x)$ and actual y, where x and y are drawn from unknown distribution p.

(b) Let D' denote a dataset of size n-$\frac{n}{k}$. Show that $\mathbb{E}_{D\sim p}[error_{k-fold}] = \mathbb{E}_{D'\sim p,(x,y)\sim p}[(y - h_{D'(x)})^2]$ 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

Answer:

$$\mathbb{E}_{D\sim p}[error_{k-fold}] = \mathbb{E}[\frac{1}{k}\sum_{i=1}^{k}\frac{1}{\frac{n}{k}}\sum_{j\in ind[i]}\ell(h_{D\backslash i}(x_j), y_j)]$$

$$= \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{D\backslash i\sim p, j\in ind[i]}[y_j - h_{D\backslash i}(x_j)]$$

Since $|D| = n$ and $\sum_{i=1}^{k}|D_i| = n$ and $|D_i| = \frac{n}{k}$ and $|D_{\backslash i}| = |D^{'}| = n - \frac{n}{k}$,

every ith fold is $\frac{n}{k}$ in size, $D_{\backslash i}$ and $D^{'}$ are $n - \frac{n}{k}$ in size and $D_{\backslash i} \sim p$ and $D^{'} \sim p$

$$\mathbb{E}_{D\sim p}[error_{k-fold}] = \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{D\backslash i\sim p, j\in ind[i]}[(y_j - h_{D\backslash i(x_j)})^2]$$

$$= \mathbb{E}_{D'\sim p,(x,y)\sim p}[(y - h_{D'}(x))^2]$$

For every $h_{D\backslash i}$(x) it was trained on $D_{\backslash i}$ of the data, and D_i which was separately and independently sampled from p and does not contain any example that appears in $D_{\backslash i}$, was used to calculate $error_{k-fold}$, hence $error_{k-fold}$ is an (almost) unbiased estimate of the risk of h_D.

(c) Assuming that the time complexity of inverting a matrix of size mxm is in $O(m^3)$, what is the complexity of computing the solution of linear regression on the dataset D?

Answer:

Solution weights $w = (X^T X)^{-1} X^T y$

Time complexity of $X^T = O(nd)$

Time complexity of $X^T X = O(nd^2) + O(nd)$

Time complexity of $(X^T X)^{-1} = O(d^3) + O(nd^2) + O(nd)$

Time complexity of $(X^T X)^{-1} X^T = O(nd^2) + O(d^3) + O(nd^2) + O(nd)$

Time complexity of $(X^T X)^{-1} X^T y = O(nd) + O(nd^2) + O(d^3) + O(nd^2) + O(nd)$

$= 2O(nd^2) + 2O(nd) + O(d^3)$

$= O(d^3 + nd^2 + nd)$

$= O(d^3 + nd^2)$

(d) Let $X_{-i} \in \mathbb{R}^{(n-\frac{n}{k})xd}$ 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?

Answer:

$$error_{k-fold} = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\frac{n}{k}}||\mathbf{y_i} - \mathbf{X_i}\mathbf{W_{-i}}||^2$$

$$= \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\frac{n}{k}}||\mathbf{y_i} - \mathbf{X_i}[(\mathbf{X_{-i}^T}\mathbf{X_{-i}})^{-1}\mathbf{X_{-i}^T}\mathbf{y_{-i}}]||^2$$

From (c):

Time complexity to compute $\mathbf{W_{-i}} = O(d^3 + (n - \frac{n}{k})d^2 + (n - \frac{n}{k})d)$

Time complexity to compute $\mathbf{y_i} - \mathbf{X_i}\mathbf{W_{-i}} = O(\frac{n}{k}d + \frac{n}{k}d^2) + O(d^3 + (n - \frac{n}{k})d^2 + (n - \frac{n}{k})d)$

Time complexity to compute $error_{k-fold} = k \times O(\frac{n}{k}d + \frac{n}{k}d^2) + k \times O(d^3 + (n - \frac{n}{k})d^2 + (n - \frac{n}{k})d)$

$$= O(nd + nd^2) + O(kd^3) + O((nk-n)d^2) + O((nk-n)d)$$

$$= O(kd^3 + knd^2 + knd)$$

$$= O(kd^3 + knd^2)$$

(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 linear regression we have $error_{k-fold} = \frac{1}{k}\sum_{i=1}^{k}||\frac{\mathbf{y_i} - \mathbf{X_i}\mathbf{w}^*}{\mathbf{I} - \mathbf{X_i}(\mathbf{X^T}\mathbf{X})^{-1}\mathbf{X_i^T}}||^2$ where $\mathbf{w}^* = (\mathbf{X^T}\mathbf{X})^{-1}\mathbf{X^T}\mathbf{y}$ is the solution of linear regression computed on the whole dataset D, the notation $\frac{\mathbf{A}}{\mathbf{B}}$ means $\mathbf{A}\mathbf{B}^{-1}$ and $\mathbf{X_i} \in \mathbb{R}^{\frac{\mathbf{n}}{\mathbf{k}}\times\mathbf{d}}$ are the data matrix and output vector obtained by keeping only the rows corresponding to the ith fold of the data. What is the complexity of evaluating this formula?

Answer:

With reference to: http://www.cs.cmu.edu/~awm/15781/assignments/hw3_sol.pdf

$$\mathbf{\hat{y}} = \mathbf{H}\mathbf{y}, \text{ where } \mathbf{H} = \mathbf{X}(\mathbf{X^T}\mathbf{X})^{-1}\mathbf{X^T}$$

$$\hat{y}_i = \sum_{j=1}^{n} H_{ij}y_j$$

Let $\hat{y}^{-i}$ be predicted y after removing ith example x_i

$$\begin{aligned} \hat{y}^{-i} &= argmin \sum_{\substack{j=1\\ j\neq i}}^{n} (y_j - \hat{y}_j^{-i})^2 \\ &= argmin \sum_{j=1}^{n} (Z_j - \hat{y}_j^{-i})^2, \text{ where } Z_j = \begin{cases} y_j, \text{when } j \neq i, \\ \hat{y}_i^{-i}, \text{when } j = i, \end{cases} \end{aligned}$$

Since $\hat{y}_i = \sum_{j=1}^{n} H_{ij} y_j$ and $\hat{y}_i^{-i} = \sum_{j=1}^{n} H_{ij} Z_j$

$$\begin{aligned} \hat{y}_i - \hat{y}_i^{-i} &= \sum_{j=1}^{n} H_{ij}(y_j - Z_j) \\ &= H_{ii}(y_i - \hat{y}_i^{-i}) \\ \hat{y}_i^{-i} &= \hat{y}_i - H_{ii} y_i + H_{ii} \hat{y}_i^{-i} \\ y_i - \hat{y}_i^{-i} &= y_i - (\hat{y}_i - H_{ii} y_i + H_{ii} \hat{y}_i^{-i}) \\ y_i - H_{ii} y_i - \hat{y}_i^{-i} + H_{ii} \hat{y}_i^{-i} &= y_i - \hat{y}_i \\ y_i - \hat{y}_i^{-i} &= \frac{y_i - \hat{y}_i}{1 - H_{ii}} \end{aligned}$$

Substituting into $error_{k-fold} = \frac{1}{k} \sum_{i=1}^{k} (y_i - \hat{y}_i^{-i})^2$

$$\begin{aligned} error_{k-fold} &= \frac{1}{k} \sum_{i=1}^{k} \left(\frac{y_i - \hat{y}_i}{1 - H_{ii}} \right)^2 \\ &= \frac{1}{k} \sum_{i=1}^{k} \left\| \frac{\mathbf{y_i} - \mathbf{X_i} \mathbf{w}^*}{\mathbf{I} - \mathbf{X_i} (\mathbf{X^T X})^{-1} \mathbf{X_i^T}} \right\|^2 \end{aligned}$$

Since we can compute $\mathbf{w}^*$ and $(X^T X)^{-1}$ once and save it to memory to be used for all points $\mathbf{X_i}$, $i \in \{1, 2, ...n\}$ for $k = n$

Time complexity of $error_{k-fold} = O(d^3 + nd^2)$

4. (a) Show that the cost function associated with logistic regression is convex.

Answer:

Given the definitions for the logistic function and the associated cost function:

$$\sigma(wx) = \frac{1}{1+e^{-wx}} \quad ; x, w \in R$$

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

Using the below definition for convexity:

$$\frac{d^2L(w)}{dw^2} \geqslant 0 \quad \forall \quad w$$

$$\frac{dL(w)}{dw} = -\left[y\frac{1}{\sigma(wx)} \cdot \sigma'(wx) + (1-y)\frac{-\sigma'(wx)}{(1-\sigma(wx)}\right]$$

Computing the derivative of $\sigma(wx)$, wrt w we get

$$\begin{aligned}\sigma'(wx) &= \frac{-1}{(1+e^{-wx})^2} \cdot (e^{-wx}) \cdot -x \\ &= \frac{xe^{-wx}}{(1+e^{-wx})^2} \\ &= \frac{x}{(1+e^{-wx})}\left(1-\frac{1}{1+e^{-wx}}\right)\end{aligned}$$

Therefore we get,

$$\boxed{\sigma'(wx) = x\sigma(wx)(1-\sigma(wx))} \tag{1}$$

plugging this back in $L'(w)$ equation we get,

$$\begin{aligned}&\frac{dL(w)}{dw} = -\left[\frac{yx}{\sigma(wx)} \cdot \sigma(wx) \cdot (1-\sigma(wx)) - \frac{(1-y)x\sigma(wx)}{(1-\sigma(wx))} \cdot (1-\sigma(wx))\right] \\ &= -[xy(1-\sigma(wx)) - (1-y)\sigma(wx)x] \\ &= -[xy - xy\sigma(wx) - \sigma(wx)x + xy\sigma(wx)] \\ &= -[xy - x\sigma(wx) + 0]\end{aligned}$$

$$\boxed{\frac{dL(w)}{dw} = x(\sigma(wx) - y)} \tag{2}$$

Now computing the double derivative of loss wrt w we get,

$$\frac{d^2L(w)}{dw^2} = x\sigma'(wx) - 0$$
$$= x^2\sigma(wx)(1 - \sigma(wx))$$

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

since, $0 < \sigma(wx) < 1$
therefore; $0 < 1 - \sigma(wx) < 1$
and $x^2 \geqslant 0$

Since all terms are individually positive, therefore $\frac{d^2L}{dw^2} \geqslant 0 \quad \forall w$
Hence the cost function associated with logistic regression is convex.

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

Answer:
Referencing the results derived in the part-a**(1)** of this question for $\sigma'(wx)$, we get:

$$\begin{aligned}\sigma'(wx) &= \frac{-1}{(1+e^{-wx})^2} \cdot (e^{-wx}) \cdot -x \\ &= \frac{xe^{-wx}}{(1+e^{-wx})^2} \\ &= \frac{x}{(1+e^{-wx})}\left(1 - \frac{1}{1+e^{-wx}}\right)\end{aligned}$$

Therefore we get:

$$\boxed{\sigma'(wx) = x\sigma(wx)(1-\sigma(wx))} \tag{3}$$

All the terms appearing in the gradient function : x, $\sigma(wx)$ are scalars, hence the dimension of gradient of 1-D logistic function is a scalar or a vector with 1-d: (1,).

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

Answer:

By definition, at the stationary points of a function, the derivative of the function is zero. Therefore $\frac{dL(w)}{dw} = 0$ for stationary points.

Re-using expressions for $\frac{dL(w)}{dw}$ derived in part-a of this question**(2)**, we get:

$$\frac{dL(w)}{dw} = x(\sigma(wx) - y) \tag{4}$$

For stationary points, the gradient will be zero, hence:

$$x(\sigma(wx) - y) = 0$$

Therefore, either $x = 0$, or $\sigma(wx) = y$. For $x = 0$ we don't get any solution for a stationary point wrt w. Using the latter, we get the below expression for w:

$$\frac{1}{1+e^{-wx}} = y; \quad x \neq 0 \quad y \in \{0,1\}$$

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

$$w = \frac{1}{x}\log(1 - \frac{1}{y}); \quad x \neq 0 \quad y \in \{0,1\}$$

for both values of y $\in$ {0,1}, no solution exists for w.
Hence no real stationary point of L(w) exists wrt to w.

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

Answer:

We can compute w_1 using previous estimate w_0 as following:

$$w_1 = w_0 - \alpha \frac{dL(w)}{dw}$$

Substituting value of $\frac{dL}{dw}$ from **(2)** we get:

$$w_1 = w_0 - \alpha x(\sigma(wx) - y)$$

where $\alpha \in R$ is the learning rate.

5. (a) Compute the derivative $f^{'}(x)$ for $f(x) = -5log(x^5)sin(x^2)$

Answer:

$$\begin{aligned} f^{'}(x) &= -5log(x^5)sin(x^2) \\ &= -5log(x^5)cos(x^2)2x + sin(x^2)(-5\frac{1}{x^5}5x^4) \\ &= -10xlog(x^5)cos(x^2) - \frac{25}{x}sin(x^2) \end{aligned}$$

(b) Compute the derivative $f^{'}(x)$ for $f(x) = 3exp(\frac{-5}{3\sigma}(x-\mu)^2)$

Answer:

$$\begin{aligned} f^{'}(x) &= 3exp(\frac{-5}{3\sigma}(x-\mu)^2) \\ &= 3(\frac{-5}{3\sigma}2(x-\mu))exp(\frac{-5}{3\sigma}(x-\mu)^2) \\ &= \frac{-10}{\sigma}(x-\mu)exp(\frac{-5}{3\sigma}(x-\mu)^2) \end{aligned}$$

(c)(i) What are the dimensions of $\frac{\partial f_i}{\partial x}$?

Answer:

$$\frac{\partial f_1}{\partial x} \text{ will have dimension of } 1 \times 2$$
$$\frac{\partial f_2}{\partial x} \text{ will have dimension of } 1 \times \text{n}$$
$$\frac{\partial f_3}{\partial x} \text{ will have dimension of } \text{n} \times \text{n} \times \text{n}$$

(c)(ii) Compute the jacobians.

Answer:

$$
\begin{aligned}
\frac{\partial f_1}{\partial x_1} &= 2\cos(2x_1)\cos(3x_2) \\
\frac{\partial f_1}{\partial x_2} &= -3\sin(2x_1)\sin(3x_2) \\
\mathbf{J}_{\mathbf{f}_1}(x1, x2) &= [\frac{\partial f_1}{\partial x_1} \quad \frac{\partial f_1}{\partial x_2}] \in \mathbb{R}^{1\times 2} \\
&= [2\cos(2x_1)\cos(3x_2) \quad -3\sin(2x_1)\sin(3x_2)]
\end{aligned}
$$

$$
\begin{aligned}
f_2(x, y) &= 3x^T y = 3\sum_{i=1}^{n} x_i y_i \\
\frac{\partial f_2(x, y)}{\partial x_i} &= y_i \ , \qquad \frac{\partial f_2(x, y)}{\partial y_i} = x_i
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial f_2(x, y)}{\partial x} &= 3[\frac{\partial f_2(x, y)}{\partial x_1} \quad \frac{\partial f_2(x, y)}{\partial x_2} \quad \ldots \quad \frac{\partial f_2(x, y)}{\partial x_n}] \\
&= 3[y_1 \quad y_2 \quad \ldots \quad y_n] \\
\frac{\partial f_2(x, y)}{\partial x} &= 3[\frac{\partial f_2(x, y)}{\partial y_1} \quad \frac{\partial f_2(x, y)}{\partial y_2} \quad \ldots \quad \frac{\partial f_2(x, y)}{\partial y_n}] \\
&= 3[x_1 \quad x_2 \quad \ldots \quad x_n] \\
\mathbf{J}_{\mathbf{f}_2}(x, y) &= \begin{bmatrix} \frac{\partial f_2(x,y)}{\partial x} \\ \frac{\partial f_2(x,y)}{\partial y} \end{bmatrix} \in \mathbb{R}^{2\times n} \\
&= 3\begin{bmatrix} y_1 & y_2 & \ldots & y_n \\ x_1 & x_2 & \ldots & x_n \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
f_3(x) &= -4xx^T \\
&= -4\begin{bmatrix} x_1x_1 & x_1x_2 & x_1x_3 & \ldots & x_1x_n \\ x_2x_1 & x_2x_2 & x_2x_3 & \ldots & x_2x_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_nx_1 & x_nx_2 & x_nx_3 & \ldots & x_nx_n \end{bmatrix} \\
\frac{\partial f_3(x)}{\partial x_i} &\in \mathbb{R}^{1\times n\times n}, i \in \{1, 2, \ldots, n\} \\
\frac{\partial f_3(x)}{\partial x_i} &= -4A^i, \text{ where } A^i_{j,k} = \begin{cases} x_k, \text{when } i = j, \\ x_j, \text{when } i = k, \\ 2x_k, \text{when } i = j = k, \\ 0, \text{otherwise} \end{cases} \\
&\qquad \text{and } j \in \{1, 2, \ldots, n\}, k \in \{1, 2, \ldots, n\} \\
\mathbf{J}_{\mathbf{f}_3}(x) &= -4\begin{bmatrix} \frac{\partial f_3}{\partial x_1} \\ \frac{\partial f_3}{\partial x_2} \\ \vdots \\ \frac{\partial f_3}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{n\times n\times n}
\end{aligned}
$$

(d) Compute the derivatives $\frac{df}{dx}$ of the following functions. Provide the dimensions of every derivative

(i) **Answer:**

$$f(z) = 2e^{-\dfrac{z^2}{2}}$$

$$\frac{df}{dz} = 2e^{-\dfrac{z^2}{2}}(-z)$$

Since f(z), z ∈ R therefore, $\dfrac{df}{dz} \in R$ and dimension = 1x1

$$z(y) = y^T S^{-1} y$$

$$\frac{dz}{dy} = y^T(S^{-1} + (S^{-1})^T)$$

if S^{-1} is symmetric, then $\dfrac{dz}{dy} = 2y^T S^{-1}$. Dimension of $\dfrac{dz}{dy} \in R^D$: 1xD

$$y(x) = \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \\ .. \\ x_n - \mu_n \end{bmatrix}$$

Since x ∈ R^D, therefore derivative is matrix with each individual x_i:

$$\frac{dy}{dx} = \begin{bmatrix} 1 & 0 & ..0 \\ 0 & 1 & ..0 \\ . & . & ... \\ 0 & 0 & ..1 \end{bmatrix}$$

$$\frac{dy}{dx} = \mathbb{I}_D$$

dimension of $\dfrac{dy}{dx}$ = DxD

combining all partial derivatives we get,

$$\frac{df}{dx} = \frac{df}{dz}\frac{dz}{dy}\frac{dy}{dx}$$

$$\frac{df}{dx} = -2zexp(-\frac{z^2}{2})y^T(S^{-1} + (S^{-1})^T)\mathbb{I}_{\mathbb{D}}$$

dimension of $\frac{df}{dx}$ = 1xD

(ii) **Answer:**

$$f(x) = \operatorname{tr}\left(xx^\top + \sigma I\right)$$

$$f(x) = \sum_{i=1}^{D} x_i^2 + \sigma$$

$$\frac{df}{dx} = \begin{bmatrix} 2x_1 \\ 2x_2 \\ ... \\ 2x_D \end{bmatrix} = 2x$$

thus the dimension of $\frac{df}{dx}$: Dx1

(iii). **Answer:**

$$f(z) = \tanh^2(z)$$

$$f(z) = \begin{bmatrix} \tanh^2 z_1) \\ \tanh^2 z_2 \\ .. \\ \tanh^2 z_M \end{bmatrix}$$

where $f(z) \in \mathbb{R}^{\mathbb{M}}$ and z $\in \mathbb{R}^M$, therefore $\frac{df}{dz} \in \mathbb{R}^{MxM}$ and can be computed as below:

$$\frac{df}{dz} = \begin{bmatrix} 2\tanh^2 z_1 sech^2 z_1 & 0 & .. & 0 \\ 0 & 2\tanh^2 z_1 sech^2 z_1 & .. & 0 \\ .. & .. & .. & .. \\ 0 & 0 & .. & 2\tanh^2 z_M sech^2 z_M \end{bmatrix}$$

$$z = Ax + b$$

where z $\in R^M$ and x $\in R^N$, therefore we get $\frac{dz}{dx} \in R^{MxN}$

$$\frac{dz}{dx} = A$$

$$\frac{df}{dx} = \frac{df}{dz}\frac{dz}{dx}$$

$$\frac{df}{dx} = diag(2tanh^2(z)sech^2(z))A$$

Thus the dimensions of each derivative term is:

(a) $\dfrac{df}{dz} = \text{MxM}$

(b) $\dfrac{dz}{dx} = \text{MxN}$

(c) $\dfrac{df}{dx} = \text{MxN}$