8843 A MATH FOUNDATIONS OF ML HW #3

NAME : ANIRUDHA SUNDARESAN 903301401

GT ID

Q1. During the course of the last week, we have discussed about inner products in their respective vector spaces and how they are uniquely defined. A norm induced by the inner product will satisfy extra properties as compared to the other norms. A Grammian operator used in finding the point on a subspace closest to a given point employs the use of inner products. This is technically called linear approximation in a Hilbert space. This concept explores an easy method to calculate the closest distançe using the properties of inner products. Using this algorithm, we have found out the structure of a polynomial closest to the exponential function. We have also seen examples (the last example in the lecture 5 notes) showing how weights can be assigned as part of the algorithm (to reward/punish) during each iteration for finding the best fitting polynomial. This incentive-driven method will make sure that the machine learns faster and with greater accuracy. A set of bases need not be orthogonal to each other, but if they are (orthogonal bases), then these help save computational time in the algorithms of linear approximation by reducing the Grammian to an identity matrix. The orthogonal bases make calculations a lot easier.

The combined concepts of orthogonal bases, inner product and linear approximation will pave the way for a better intuitive understanding of machine learning. The exact use of linear algebra in machine learning is still very cloudy for me as we have not yet covered its applications. On googling, I found that if I can understand machine learning methods at the level of vectors and matrices, I will be able to improve my intuition for how and when they work in the case of machine learning. Matrices help us to look at all the data as a single entity. They also say that it has been observed through practice that representing large sets of data (in the form of vectors and matrices) help us visualize the data better. Linear algebra provides the computational engine for the majority of machine learning algorithms. Many of the problems in ML are about data fitting: Given a subspace S and a point p in a given scalar product space, find on S a point closest to p. Figuring out this 'best fit' is important for classifying unlabelled data points. With these in mind, I look forward to the next unit which deals with the concepts of linear estimation.

Q2.

(a) $G \rightarrow N \times N$

If the columns of G are linearly dependent, then, through the technique of elementary column transformation, we can get a column of zeros and hence, determinant becomes zero.

Inverse will not exist if determinant $=0$ because

$$G^{-1} = \frac{\text{adjoint}(G)}{|G|}$$

If $Gx=0 \;\forall\; x \neq 0$, x is in null space of G. Thus, G, maybe many-to-one mapping and thus non-invertible.

$\therefore$ For G^{-1} to exist, the columns must be linearly independent and $Gx \neq 0$ for all $x \neq 0$.

ANOTHER METHOD :

Let b element of vector space.

$\therefore b = x_1 v_1 + x_2 v_2 + \cdots + x_N v_N$

If x_1 is 0 and $x_2 = x_3 = \cdots x_N = 0. \Rightarrow b = $

If x_k is $\frac{1}{\alpha}$ and $x_1 = x_2 \cdots = x_N = 0$ except x_k $\Rightarrow b = \frac{1}{\alpha}$

Now, if say v_1 and v_K are linearly dependent, by the relation: $v_K = \alpha v_1$ then,

the coordinates $\{1, 0; 0 \ldots 0\}$ and $\{0, 0 \ldots \frac{1}{\alpha}, 0 \ldots 0\}$ ↑ at K^{th} position

corresponds to $v_1 (= \frac{1}{\alpha} v_k)$

Two sets of co-ordinates gives us same vector in space.

$\therefore$ Non-invertible system. if linearly dependent vectors.

$\therefore$ For G to be invertible, each vector should have unique co-ordinate. For this, v_k must be linearly independent.

(b)

$$G = \begin{bmatrix} \langle \varphi_1, \varphi_1 \rangle & \langle \varphi_2, \varphi_1 \rangle & \langle \varphi_3, \varphi_1 \rangle & \cdots & \langle \varphi_N, \varphi_1 \rangle \\ \langle \varphi_1, \varphi_2 \rangle & \langle \varphi_2, \varphi_2 \rangle & \cdots & & \langle \varphi_N, \varphi_2 \rangle \\ \vdots & & & & \\ \langle \varphi_1, \varphi_N \rangle & \langle \varphi_2, \varphi_N \rangle & \cdots & & \langle \varphi_N, \varphi_N \rangle \end{bmatrix}$$

From part (a), G matrix must have linearly independent column vectors.

Let us say for instance, φ_1 and φ_k are linearly dependent. $\varphi_k = \alpha \varphi_1$. Then $\langle \varphi_K, \varphi_1 \rangle = \alpha \langle \varphi_1, \varphi_1 \rangle$ and $\langle \varphi_K, \varphi_N \rangle = \alpha \langle \varphi_1, \varphi_N \rangle$

Two columns in G will be :

$$\begin{bmatrix} \langle \varphi_1, \varphi_1 \rangle & \cdots & \alpha \langle \varphi_1, \varphi_1 \rangle & \cdots & \langle \varphi_N, \varphi_1 \rangle \\ \langle \varphi_1, \varphi_2 \rangle & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots \\ \langle \varphi_1, \varphi_N \rangle & & \alpha \langle \varphi_1, \varphi_N \rangle & & \langle \varphi_N, \varphi_N \rangle \end{bmatrix}$$

$$\begin{matrix} \langle \varphi_k, \varphi_1 \rangle \\ \langle \varphi_k, \varphi_2 \rangle \\ \vdots \\ \langle \varphi_k, \varphi_N \rangle . \end{matrix}$$

column k and column 1 ~~...~~ thus ~~linearly~~ become linearly dependent and thus matrix becomes non-invertible (as det $= 0$)

$\therefore$ For G to be invertible, $\{\varphi_n\}$ must be linearly independent.

[NOTE : The proof is valid for φ_k and φ_l instead of φ_1 and φ_k].

$G_1, G_2, G_3 \rightarrow$ zero mean Gaussian random variables.

$R \rightarrow$ covariance matrix

$$R_{ij} = E[G_i G_j]$$

$$S = \text{span}(G_1, G_2, G_3)$$

Elements of $S \rightarrow$ zero mean Gaussian random variables.

(a) $\langle X, Y \rangle = E[XY]$

To prove that this is a valid inner product, it must satisfy the properties of symmetry, positivity, linearity and homogeneity.

For symmetry, $\langle f, g \rangle = \overline{\langle g, f \rangle}$

But here, since it is over real ($\mathbb{R}$), $\overline{\langle g, f \rangle} = \langle g, f \rangle$

$\langle X, Y \rangle = E[XY]$ and $\langle Y, X \rangle = E[YX] = E[XY]$

$\therefore$ Symmetry property proved as $\langle X, Y \rangle = \langle Y, X \rangle$

For positivity, $\langle f, f \rangle \geq 0$ always and equal to 0 if and only if $f = 0$

$\langle X, X \rangle = E[X^2] \rightarrow$ expectation of a set of positive elements.

$E(X^2) = \sum_x \underbrace{x^2 p_X(x)}_{\geq 0}$

Hence $E(X^2) \geq 0$

Hence $E[X^2] = \langle X, X \rangle \geq 0$.

If all values of the random variable $X = 0$, then $E(X^2) = 0$

$\therefore$ Positivity property proved.

For homogeneity, $\langle \alpha f, g \rangle = \alpha \langle f, g \rangle$

$\langle \alpha X, Y \rangle = E[\alpha XY] = \alpha E[XY] = \alpha \langle X, Y \rangle$

property of expectation

Since we get $\langle \alpha X, Y \rangle = \alpha \langle X, Y \rangle$

Homogeneity property proved.

For linearity:

$\langle f+g, h \rangle = \langle f, h \rangle + \langle g, h \rangle$

$\langle X+Y, Z \rangle = E((X+Y)Z) = E[XZ] + E[YZ]$

property of expectation

$\Rightarrow \langle X, Z \rangle + \langle Y, Z \rangle$

$\therefore$ Linearity property proved.

$\therefore$ $\langle X, Y \rangle = E[XY]$ is a valid inner product on the vector space S.

Now,

distance induced by inner product = norm.

Inner product of a vector with itself = $|\text{norm}|^2$

$\langle X, X \rangle = |\text{norm}|^2 = E[X^2]$

$\therefore |\text{norm}| = \sqrt{E[X^2]}$

$\Downarrow$

Distance induced by the inner product $= \sqrt{E[X^2]}$

Can be interpreted as the (square) root of mean (of) square error.

$\Rightarrow \sqrt{\int x^2 P_X(x)\, dx}$

OR

$\sqrt{\sum x^2 P_X(x)}$

$X = a_1 G_1 + a_2 G_2 + a_3 G_3 \qquad Y = b_1 G_1 + b_2 G_2 + b_3 G_3$

$$\langle X, Y\rangle = E(XY)$$
$$= E\left((a_1 G_1 + a_2 G_2 + a_3 G_3)(b_1 G_1 + b_2 G_2 + b_3 G_3)\right)$$

$$= E\left[a_1 b_1 G_1^2 + a_1 b_2 G_1 G_2 + a_1 b_3 G_1 G_3 + a_2 b_1 G_2 G_1 + a_2 b_2 G_2^2 + a_2 b_3 G_2 G_3 + a_3 b_1 G_3 G_1 + a_3 b_2 G_3 G_2 + a_3 b_3 G_3^2\right]$$

Using property of expectation

$$= \left[a_1 b_1 E(G_1^2) + a_1 b_2 E(G_1 G_2) + a_1 b_3 E(G_1 G_3) + a_2 b_1 E(G_2 G_1) + a_2 b_2 E(G_2^2) + a_2 b_3 E(G_2 G_3) + a_3 b_1 E(G_3 G_1) + a_3 b_2 E(G_3 G_2) + a_3 b_3 E(G_3^2)\right]$$

$$= \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} E(G_1)^2 & E[G_1 G_2] & E(G_1 G_3) \\ E[G_2 G_1] & E[G_2^2] & E[G_2 G_3] \\ E[G_3 G_1] & E[G_3 G_2] & E[(G_3)^2] \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

in matrix form.

$$= a^T R b \quad \text{where} \quad a = [a_1 \; a_2 \; a_3]^T, \quad b = [b_1 \; b_2 \; b_3]^T$$

$$\text{and} \quad R_{ij} = E[G_i G_j]$$

$$\therefore \langle X, Y\rangle = a^T R b$$

(C) Let $R = \begin{bmatrix} 1 & 0.4 & -0.2 \\ 0.4 & 1 & 0.4 \\ -0.2 & 0.4 & 1 \end{bmatrix}$

$X = G_1$, $Y = G_2$ and $Z = G_1 + G_2 + G_3$

$$\hat{Z} = \alpha_1 X + \alpha_2 Y$$

From R matrix, $E(G_1^2) = E(G_2^2) = E(G_3^2) = 1$

$E(G_2 G_1) = E(G_1 G_2) = 0.4$

$E(G_2 G_3) = E(G_3 G_2) = 0.4$

$E(G_3 G_1) = E(G_1 G_3) = -0.2$

X and Y are thus taken as the basis vectors.

$$\boxed{Ga = b}$$

$G = \begin{bmatrix} G_{11} & G_{21} \\ G_{12} & G_{22} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix}}_{2\times 2}$

since this is $\begin{bmatrix} \langle X.X \rangle & \langle Y.X \rangle \\ \langle X.Y \rangle & \langle Y.Y \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} E(G_1^2) & E(G_2 G_1) \\ E(G_1 G_2) & E(G_2^2) \end{bmatrix}$

$b = \underbrace{\begin{bmatrix} \langle Z.X \rangle \\ \langle Z.Y \rangle \end{bmatrix}}_{2\times 1}$

$b = \begin{bmatrix} E[(G_1 + G_2 + G_3)\cdot(G_1)] \\ E[(G_1 + G_2 + G_3)(G_2)] \end{bmatrix}$

$b = \begin{bmatrix} E(G_1^2) + E(G_1 G_2) + E(G_1 G_3) \\ E(G_1 G_2) + E(G_2^2) + E(G_2 G_3) \end{bmatrix} = \begin{bmatrix} 1 + 0.4 - 0.2 \\ 0.4 + 1 + 0.4 \end{bmatrix}$

$= \begin{bmatrix} 1.2 \\ 1.8 \end{bmatrix}$

And $a = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$

from $\underline{\underline{Ga = b}}$ $\Rightarrow \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 1.8 \end{bmatrix}$

$$\alpha_1 + 0.4\alpha_2 = 1.2$$
$$0.4\alpha_1 + \alpha_2 = 1.8$$

this, on solving gives $\boxed{\begin{aligned} \alpha_1 &= 0.572 \\ \alpha_2 &= 1.5712 \end{aligned}}$

$\therefore \underline{\underline{\hat{Z} = 0.572X + 1.5712Y}}$ for minimum MSE.

$\hookrightarrow$ closest point on subspace to Z.

Mean square error (for above α_1, α_2) $= E[(Z-\hat{Z})^2]$

$$E[(G_1 + G_2 + G_3 - \alpha_1 G_1 - \alpha_2 G_2)^2]$$

$$= E[((1-\alpha_1)G_1 + (1-\alpha_2)G_2 + G_3)^2]$$

$$= E[(0.428G_1 + (-0.5712)G_2 + G_3)^2]$$

$$= E[(0.428G_1 - 0.5712G_2 + G_3)(0.428G_1 - 0.5712G_2 + G_3)]$$

$= E[XY] \Rightarrow \langle X, Y \rangle = a^T R b.$ as from 5(b)

$$\therefore \Rightarrow \underset{1\times 3}{\begin{bmatrix} 0.428 & -0.5712 & 1 \end{bmatrix}} \underset{3\times 3}{[R]} \underset{3\times 1}{\begin{bmatrix} 0.428 \\ -0.5712 \\ 1 \end{bmatrix}}$$

$\Rightarrow \underset{(1\times 1)}{\underline{\underline{0.6857}}} \rightarrow$ Mean square error.

(d) Copy of the code given as printout.

(e) On implementing the script written for 5(d) with the values given in (e), we get the following:

$MSE_{Experimental}$ (from the function in 5(d)) = 0.2260

$MSE_{sampled}$ (from 5(e)) = 0.2249

From the function written in 5(d), I got

$$\boxed{\begin{aligned} \alpha_1 &= 0.4313 \\ \alpha_2 &= 0.2524 \end{aligned}}$$ and

We observe that $MSE_{function}$ & $MSE_{sampled}$ are almost same.

Variance (z) comes out to be <u>4.2686</u>

MSE <u>does not</u> compare favorably with Variance (z)