ROUGH WORK

Q1: For the following questions, circle the right answer. (None, One or More is Correct). [30 min; $10 \times 3 = 30$ points]

1. Consider X to be a square matrix of size $n \times n$ and $X = UDV^T$. (i.e., SVD)
   (a) Both X^TX and XX^T have the same eigenvalues
   (b) Both X^TX and XX^T have the same eigenvectors
   (c) X, XX^TX and XX^T have the same eigenvalues
   (d) D^2 contains the eigenvalues of X^TX on its diagonal
   (e) D contains the eigenvalues of X^TX on its diagonal

2. Consider X to be a square matrix of size $n \times n$ and $X = UDV^T$. Then:
   (a) If rank(X) = n, D has all non-zero entries in diagonal.
   (b) If rank(X) = k, D has k zeros in diagonal
   (c) If rank(X) = k, D has $n - k$ zeros in diagonal
   (d) if rank(X) = n but $|A|$ is a very small number then, D takes the form $D = diag(d_1, d_2, .., \epsilon)$ where ϵ is a very small number
   (e) None of these

3. Given a set of 2D points X on the vertical line $x_2 = 5$, $X = \{[1,5]^T, [2,5]^T, [3,5]^T, [4,5]^T, [5,5]^T\}$
   We compute the covariance matrix, and its eigen values and eigen vectors. Then:
   (a) $\lambda_1 \geq \lambda_2$
   (b) μ is on the same line.
   (c) Σ is singular
   (d) Σ is diagonal
   (e) None of the above.

4. (Use notations and conventions from the class) Consider the problem of linear regression where we minimize the loss
$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N} \alpha_i (y_i - \mathbf{w}^T\mathbf{x}_i)^2 + \lambda_1 g(\mathbf{w})$$
   where $g()$ is a regularization term. We also write the loss in matrix form as
$$\mathcal{L}_2 = \frac{1}{N}[Y - \mathbf{X}\mathbf{w}]^T A[Y - \mathbf{X}\mathbf{w}] + \lambda_2 g(\mathbf{w}).$$
   Now: see $\mathcal{L}_2$ closely,
   (a) When A is a diagonal matrix, this is equivalent to weighing each sample independently as in $\mathcal{L}_1$
   (b) When A is not a diagonal matrix, this loss does not make any sense. Don't ever use.
   (c) When A is PD, we can do Cholesky decomposition of A as LL^T and an equivalent formulation is possible in $\mathcal{L}_1$ is each sample getting transformed as $\mathbf{L}^T\mathbf{x}_i$
   (d) When A is a rank deficient matrix, an equivalent formulation is possible in $\mathcal{L}_1$ with a dimensionality reduction of all the $\mathbf{x}_i$.
   (e) None of the above

5. If $\mathbf{A}$ is a $n \times n$ matrix, with every pair of columns orthogonal i.e., $\mathbf{a_i \cdot a_j = 0}\ \forall i, j$ and $||\mathbf{a_i}|| = 1$. Then:
   (a) $\mathbf{A^{-1} = A}^T$.
   (b) $\mathbf{AA}^T = \mathbf{I}$
   (c) $\mathbf{AA}^T$ has only one 1 in every column and all others zero.
   (d) $\mathbf{A}^{-1}$ has only one 1 in every column and all others zero.
   (e) none of the above
6. Overfitting can be reduced if
   (a) we can have more training data
   (b) we can reduce the amount of training data
   (c) we reduce the dimensionality of all samples (i.e., $\mathbf{x}$)
   (d) we reduce the learning rate (η) in gradient descent
   (e) we increase the learning rate (η) in gradient descent.
7. A covariance matrix is always:
   (a) square
   (b) full rank (i.e., rank =d)
   (c) triangular
   (d) PSD
   (e) PD
   (f) has at least one eigen value negative
   (g) no eigen value is imaginary
   (h) all eigen values are non-negative
8. Let $\mathbf{x}_i \in R^3$. A set of N points $\mathbf{x}_i$ are on a line. Then rank of a matrix $A = \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T$ is:

   (a) 1 (b) 2 (c) 3 (d) N (e) ∞
9. Gradient Descent rule (for minimizing a differentiable loss)
   (a) says that the solution needs to be updated along the negative gradient of loss function.
   (b) says that the solution needs to be updated along the gradient of loss function.
   (c) can be used for solving non-convex optimization problems (as in neural networks)
   (d) can be used for solving convex optimization problems (as in linear regression)
   (e) all of the above.
10. Identify the true statements:
   (a) K Nearest Neighbour is a lazy algorithm for classification
   (b) All real data is truly multivariate Gaussian
   (c) L_∞ regression provides the best sparse solution
   (d) L2 regularization is popularly used because it provides differentiability

Q2. Answer briefly in the space provided [30 min; 5 × 6 = 30 points]

1. A test for a certain rare disease is assumed to be correct 95% of the time: if a person has (does not have) the disease, the test results are positive (negative) with probability 0.95. A random person drawn from a certain population has probability 0.001 of having the disease. Given that the person just tested positive, what is the probability that the person has the disease?

   Solution:

   Given:
   $P($ testing positive $|$ person has the disease $) = 0.95$
   $P($ testing negative $|$ person does not have the disease $) = 0.95$
   $\implies P($ testing positive $|$ person does not have the disease $) = 1 - 0.95 = 0.05$
   $P($ having the disease $) = 0.001$
   $\implies P($ not having the disease $) = 1 - 0.001 = 0.999$

   Bayes theorem states:
   $P(B|A) = \frac{P(B)P(A|B)}{P(A)}$
   $P(A) = P(B)P(A|B) + P(\text{not } B)P(A|\text{not } B)$

   In the given question:
   A = testing positive
   B = having the disease

   Substituting in the equation we get:
   $P($ having the disease $|$ testing positive $) = \frac{0.001 \times 0.95}{(0.001 \times 0.95) + (0.999 \times 0.5)}$
   $= \frac{0.00095}{0.0509} \approx 0.0187$
   Marks are deducted for:

   - Not writing the Bayesian formula
   - Incorrect substitution
   - Early approximation of values (0.04995 to 0.05, etc)
   - Missing intermediate steps or final answer (0.01866)

2. Consider a two class classification problem with respective means as μ_A and μ_B. Both are univariate Gaussian with variance σ^2. The optimal threshold/classifier is θ. If prior probability of class B is twice as that of class A, will θ be close to μ_A or μ_B? (one word answer). Draw a neat simple diagram and demonstrate.

   Solution:

   μ_A

   - 2 marks for correct answer
   - 1 mark for drawing 2 Gaussians
   - 1 mark for same width of both curves (same variance)
   - 1 mark for height difference
   - 1 mark for showing correct decision boundary

   Any of the below diagrams is accepted:

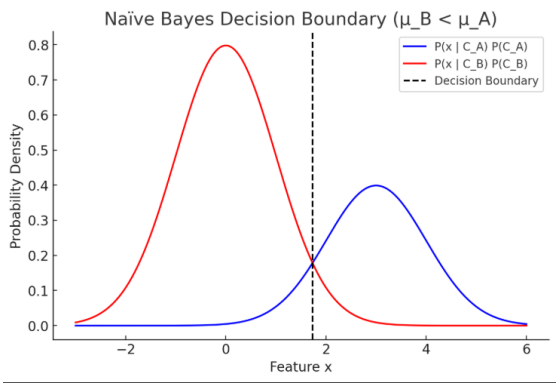

Figure 1

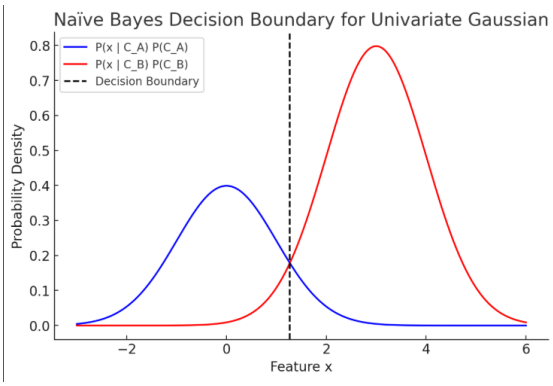

Figure 2

3. Consider the SVD of $\mathbf{A}$ as $\mathbf{A} = \mathbf{UDV^T}$, Derive a relationship between eigen values of A^TA and Singular values of A.

   Solution:
   $A = UDV^T$
   $\Rightarrow A^TA = (UDV^T)^T(UDV^T)$
   $\Rightarrow A^TA = VD^TU^TUDV^T$
   $\Rightarrow A^TA = VD^TDV^T$ [Since U is orthogonal, $UU^T = U^TU = I$]
   By properties of SVD decomposition
   D is diagonal matrix of singular values of A
   $\Longrightarrow$ D^TD is diagonal matrix of squared singular values of A
   Thus $A^TA = V(D^TD)V^T$ forms the eigenvalue decomposition
   i.e. D^TD is diagonal matrix of eigenvalues of A^TA
   From the above two statements, we get the relation -
   The eigenvalues of A^TA are the same as the squared singular values of A

4. Derive a gradient descent update equation for the minimizing the below loss function

$$\mathcal{L}(\mathbf{w}) = \mathbf{w}^TA\mathbf{w} - 8(\mathbf{w^T}\mathbf{w} - 1) + 25$$

   Solution:

   Compute the Gradient of the Loss Function

   To derive the gradient compute the partial derivatives of $L(w)$ with respect to w:

   1. The gradient of w^TAw with respect to w is:

$$\nabla_w \left(w^TAw\right) = 2Aw$$

   (Note that A is assumed to be a symmetric matrix).

   2. The gradient of w^Tw with respect to w is $2w$, so the gradient of this term is:

$$\nabla_w \left(-8(w^Tw - 1)\right) = -16w$$

   3. The gradient of a constant term is zero.

   Combining the Gradients,

$$\nabla_w L(w) = 2Aw - 16w$$

   Equation to update parameters:

$$w_{\text{new}} = w - \eta \nabla_w L(w)$$

where η is the learning rate.

$$w_{\text{new}} = w - \eta(2Aw - 16w)$$

Simplifying:

$$w_{\text{new}} = w - \eta\left((2A - 16I)w\right)$$

where I is the identity matrix of the same dimension as A.

- 1 mark for correct gradient descent formula
- 3 marks for calculating derivative of each term correctly
- 1 mark for correct final update equation
- 1 mark for overall understanding
- *(This question has been marked on a deductive basis)*

5. We know often matrices have low rank. We use this effectively in machine learning
   (a) Complete the matrix A if it is of rank-1

$$A = \begin{bmatrix} ?, ?, 2, ?, ? \\ ?, 4, ?, 6, ? \\ ?, 2, ?, ?, 1 \\ 3, ?, ?, ?, ? \\ 1, ?, 1, ?, 1 \end{bmatrix} \Rightarrow$$

   (b) Find the rank 1 approximation (nearest rank 1 matrix) of A [1]

$$A = \begin{bmatrix} 1 & -1 & 3 \\ 3 & 1 & 1 \end{bmatrix}$$

Solution:
Note - the marks distribution for this question is 2 and 4 respectively for parts (a) and (b) (a) Matrix of rank 1 implies that all the rows are scalar multiples of each other
Pairwise compare rows and use proportion to compute the missing values of the row
For example, comparing the 3rd and the 5th row -
$\frac{?}{1} = \frac{2}{?} = \frac{?}{1} = \frac{?}{?} = \frac{1}{1}$
$\implies$ 3rd row = 5th row = $\begin{bmatrix} 1 & 2 & 1 & ? & 1 \end{bmatrix}$

Final answer:

$$A = \begin{bmatrix} 2, 4, 2, 6, 2 \\ 2, 4, 2, 6, 2 \\ 1, 2, 1, 3, 1 \\ 3, 6, 3, 9, 3 \\ 1, 2, 1, 3, 1 \end{bmatrix}$$

[1] SVD of $A = \begin{bmatrix} 1 & -1 & 3 \\ 3 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 0 & \sqrt{6} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{bmatrix}$

(b) Given the SVD decomposition

$$A = \begin{bmatrix} 1 & -1 & 3 \\ 3 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 0 & \sqrt{6} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{bmatrix}$$

For the rank 1 approximation, we will take the maximum singualr value of A as σ, and its corresponding vectors from U and V of the SVD decomposition as u and v respectively
$\therefore$ rank 1 approximation $A' = \sigma_1 u v^T$

$$= 4 \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$= 4 \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 2 \\ 2 & 0 & 2 \end{bmatrix}$$

Q3. Eigen Face Algorithm: Here, question as well as answer is provided!!. You only need to fill in the blanks; You are given many options to choose from at the end [2]

If the given options are insufficient (which surely is) or steps are missing or reordering, please add/edit precisely. If there is a step missing, add at the end and edit numbering (eg. add a step 7b, if a step is missing 7 and 8) [30 min; 30 point]

You are given 1000 images of frontal face cropped from 200 individuals (5 distinct faces per person). Each image is of size 200×200 pixel size. We are now given a query face image $\mathbf{a}$ of size 200×200. We need to compute the most similar 10 faces to $\mathbf{a}$.

Answer Key:

1. Flatten all the images into a vector $\mathbf{x}_i$ of $d =$ 40K. There are a total of $N =$ 1000 samples for computing the dimensionality reduction matrix A for PCA.
2. Compute mean $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$ from these samples. Subtract mean from all samples to normalize $\mathbf{x}' = \mathbf{x} - \mu$
3. Arrange all the mean subtracted samples as columns and create a data matrix X of size $d \times N$.
4. Compute covariance matrix Σ from X as $\frac{1}{N}XX^T$. Note that the dimension of Σ is $d \times d$.
5. This Σ matrix can have a maximum of $N = 1000$ non-zero eigenvalues.
6. Computing eigenvectors of Σ is costly; to simplify the computation, we first find a smaller matrix C of size $N \times N$ from X as $C = X^T X$.
7. We know that the eigen values of C and Σ are related: they are the same.
8. We compute eigen vectors of C as $\mathbf{u}$ of size $N \times 1$, and compute eigen vectors $\mathbf{v}$ of Σ of size $d \times 1$. This is done using a simple multiplication as $\mathbf{v} = X\mathbf{u}$.
9. We choose k prominent eigen vectors by analyzing the eigen value spectrum. We found $k = 100$
10. We arrange the k eigen vectors as rows and form a matrix A of size $k \times d$.
11. New reduced-dimensional vectors: $\mathbf{p}_i = A\mathbf{x}'_i$.

[2] $(\cdot)10(\cdot)100(\cdot)10^4(\cdot)10^6(\cdot)40K(\cdot)N\times N(\cdot)d\times d(\cdot)N\times d(\cdot)d\times N(\cdot)\frac{1}{N}XX^T(\cdot)\frac{1}{N}X^TX(\cdot)\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i(\cdot)$ same$(\cdot)$scaled versions$(\cdot)k\times N(\cdot)k\times d(\cdot)A\mathbf{x}_i(\cdot)A\mathbf{x}'_i$

12. We now have N reduced dimensional samples $\mathbf{p}_1, \ldots, \mathbf{p}_N$ of size $\underline{k}$.
13. Similarly we compute the reduced dimensional representation of $\mathbf{a}$: $\mathbf{q} = \underline{A(\mathbf{a} - \mu)}$.
14. We compute the distance of $\mathbf{q}$ from all $\mathbf{p}_i$ as $d_i = \underline{\|q - p_i\|^2}$.
15. We sort d_i and find the $\mathbf{p}_i$ corresponding to the smaller $\underline{10}$ distances, and return the corresponding faces.

ROUGH WORK