MACHINE LEARNING

ASSIGNMENT - 02

<Q = 1> Image data :-

Dimension of image vector = D.

No. of observations = N.

D >> N.

Given data is mean Removed.

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i = 0.$$

in order to apply PCA.

Covariance Matrix,

$$S = \frac{1}{N} \sum_{i=1}^{N} x_i x^T ,$$

st; $X = [x_1 \; x_2 \; \dots \; x_N]^T$.

Size of Data Matrix = $N \times D$.

where each image is a row vector of size D,

Show that.

$$S = \frac{1}{N} X^T X$$

Let $x_1, x_2 \ldots x_n$ be a set of $1 \times D$ vectors.

$$x_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{iD} \end{bmatrix}$$

Let X be a $n \times D$ matrix, with columns.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Let $Q = X^T X$, be the $D \times D$ matrix.

$$S = Q = X^T X = \begin{bmatrix} (x_1)^T & (x_2)^T & \cdots & (x_n)^T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

$\Rightarrow$ Q is square

$\Rightarrow$ Q is symmetric.

$\Rightarrow$ Q is the covariance matrix.

$\Rightarrow$

$$S = \frac{1}{N} \begin{bmatrix} \sum_i x_{1i}^2 & & \\ & & \\ & & \sum_{i=1}^{D} x_{n}^2 \end{bmatrix}$$

$$= \frac{1}{N} \begin{bmatrix} var(x_1, x_1) & cov(x_1, x_N) \\ \vdots & \vdots \\ cov(x_N, x_1) & var(x_N, x_N) \end{bmatrix}$$

1.(b) To proof any eigen vector of u of S can be obtained using the eigen vector v of $\hat{S} = XX^T$ as $u = X^T v$.

Proof:

For S.

the eigen vector, eigen value equation is

$$\frac{1}{N} X^T X u_i = \lambda_i u_i$$

Now pre-multiply both sides by X to give.

$$\frac{1}{N} X X^T (X u_i) = \lambda_i (X u_i)$$

if now, we define $v_i = X u_i$, we obtain.

$$\frac{1}{N} X X^T v_i = \lambda_i v_i$$

which is an eigen vector/value equation.

we multiply both sides by X^T.

$$\left(\frac{1}{N} X^T X\right) (X^T v_i) = \lambda_i (X^T v_i).$$

we see that $X^T v_i$ is an eigen vector of S with eigen value λ_i,

it gives.

$$u_i = \frac{1}{(N\lambda_i)^{1/2}} X^T v_i$$

Hence Proved.

1. (c)

In many cases it could occur that the no. of dimensions of a single sample image is much larger than the no. of images in dataset.

In such cases if we have to calculate the PCA of a D×D covariance matrix, the computational complexity is higher.

On the other hand if we first compute PCA of a N×N matrix and then derive the eigen vectors by multiplying with X^T, the overall computational complexity is O(N^2) + O(D).

<2>

The attached files contain :-

(a) main.m :- the main script of the program.
↳ calls to → load an image
↳ PCA, Lda, and plotting
the results.

(b) loadimage.m :- Loads the training set.

(c) pca.m :- implements pca

(d) Lda.m :- implements LDA.

* The normalized eigen value will be printed after the program is executed.

* The reconstructed image is recognizable using one eigen vector.

* Using value of $K=10$ is good.
we can reduce it below that, by Following the residue error.

* Prediction Rate was 70 %

For the given ~~training~~ Test set.