Ans 1: MATLAB Code:

```
clear;clc;
load('teapots.mat')
teapot_data = teapotImages;
m = mean(teapot_data);
X = teapot_data - m;
C = cov(X);
[V, D] = eig(C);
[d, ind] = sort(diag(D),'descend');
d = d(1:3,:);
v = V(:,ind(1:3));
c = X*v;
X_hat = m+c*v';

%10 images
for i = 11:20
    figure(i);
    colormap gray;
    subplot(1,2,1);
    imagesc(reshape(teapot_data(i,:),38,50));
    title('Before Recon');
    axis image;
    subplot(1,2,2)
    imagesc(reshape(X_hat(i,:),38,50));
    title('After Recon');
    axis image;
end
norm(teapot_data-X_hat)
```

--

```
clear;clc;

load('teapots.mat')

X = teapotImages;
[coefficient_of_3, score3] = pca(X,'Algorithm','eig','Rows','all','NumComponents',3);
Xhat3 = mean(X)+score3*coefficient_of_3';
[coefficient_of_6, score6] = pca(X,'Algorithm','eig','Rows','all','NumComponents',6);
Xhat6 = mean(X)+score6*coefficient_of_6';
[coefficient_of_32, score32] =
pca(X,'Algorithm','eig','Rows','all','NumComponents',32);
Xhat32 = mean(X)+score32*coefficient_of_32';


figure(1);
colormap gray;
subplot(2,2,1);
imagesc(reshape(data(10,:),38,50));
title('Before');
axis image;
subplot(2,2,2)
imagesc(reshape(Xhat3(10,:),38,50));
title('TOP3');
axis image;
```

```
subplot(2,2,3)
imagesc(reshape(Xhat6(10,:),38,50));
title('TOP6');
axis image;
subplot(2,2,4)
imagesc(reshape(Xhat32(10,:),38,50));
title('TOP32');
axis image;
```

--

Python code for determining Eigenvalues:

```
import numpy as np

import matplotlib.pyplot as plt
from numpy.linalg import eig
from scipy.io import loadmat

if __name__=='__main__':
    data = loadmat('teapots.mat')
    X = data['teapotImages']
    u = np.mean(X, axis=0).reshape(1,1900)
    x = X - np.repeat(u,100,axis=0)
    C = x.T.dot(x)/(x.shape[0]-1)
    D, V = eig(C)
    print(x.shape)
    print(C)
    print(D[:3])
    print(V[:,:3])
```

Writeup:

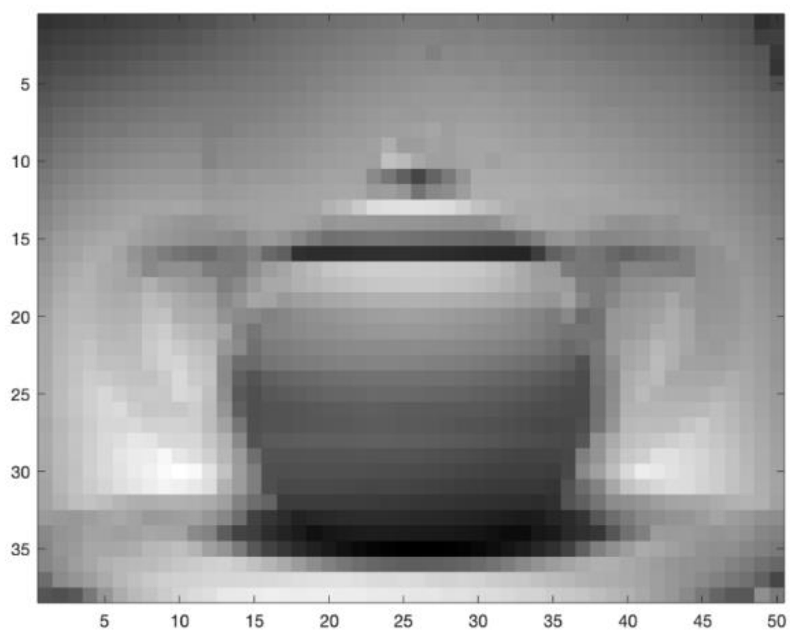
(1) We use PCA on image data provided to reconstruct it. We determine top 3 most significant components of image data, encode them with mean of the data for reconstruction.

First, compute mean of image data of shape (100, 1900) ~~using pyd~~ Then minus the mean from all data points to get decentered values

$$\boxed{\mu = mean(X)}$$

$$\boxed{x = X - \mu}$$

Mean of image data is shown as follows



Then, find covariance matrix of data of shape (100, 1900)

$$C = \mathrm{cov}(x) = x^T x$$ which has dimensions (1900, 1900).

Apply Eigenvalue Decomposition to covariance matrix to find

V = Eigenvector matrix

Matrix with eigenvalue on the diagonal Λ

$$C = V \Lambda V^{-1}$$

→ Determine most significant value in eigenvalues

As cov matrix is symmetric and positive-definite, eigenvalues are all non-negative

Top 3 most significant eigenvalues are:

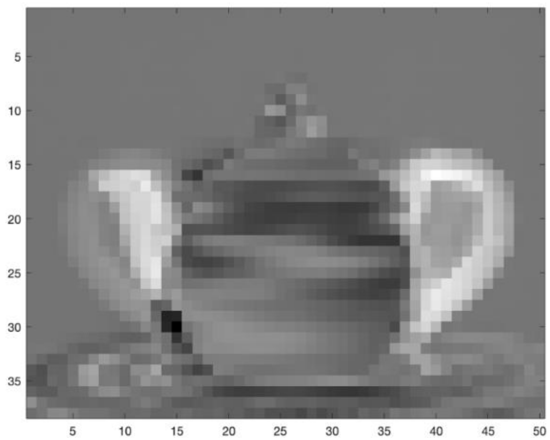
[4.2150, 3.0168, 2.0993]

each with corresponding eigenvector

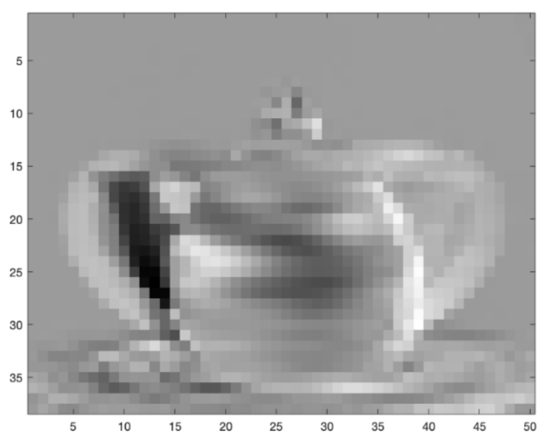
Each eigenvector can be shown as an image, depicted below

Eigen Value=4.2150

Eigen Value: 3.0168



Eigen Value: 2.0993



→ Encode by calculating coefficient matrix

Coeff matrix $(c_{ij}) = (X_i - \mu)^T V'_j = x_i^T V'_j$,

is inner product of x and top-3 eigenvectors

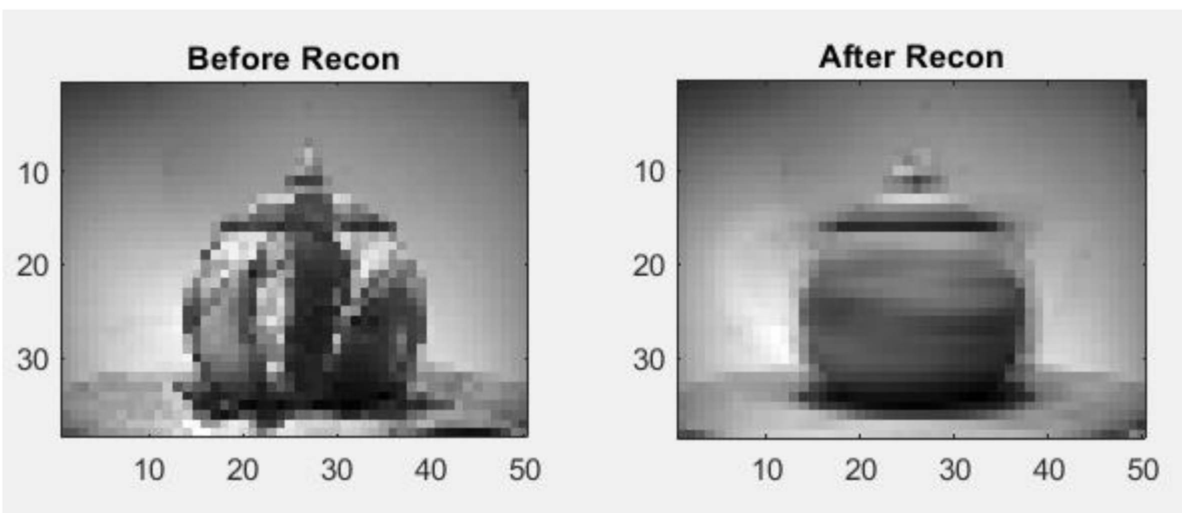
$$\boxed{c = xV'}$$

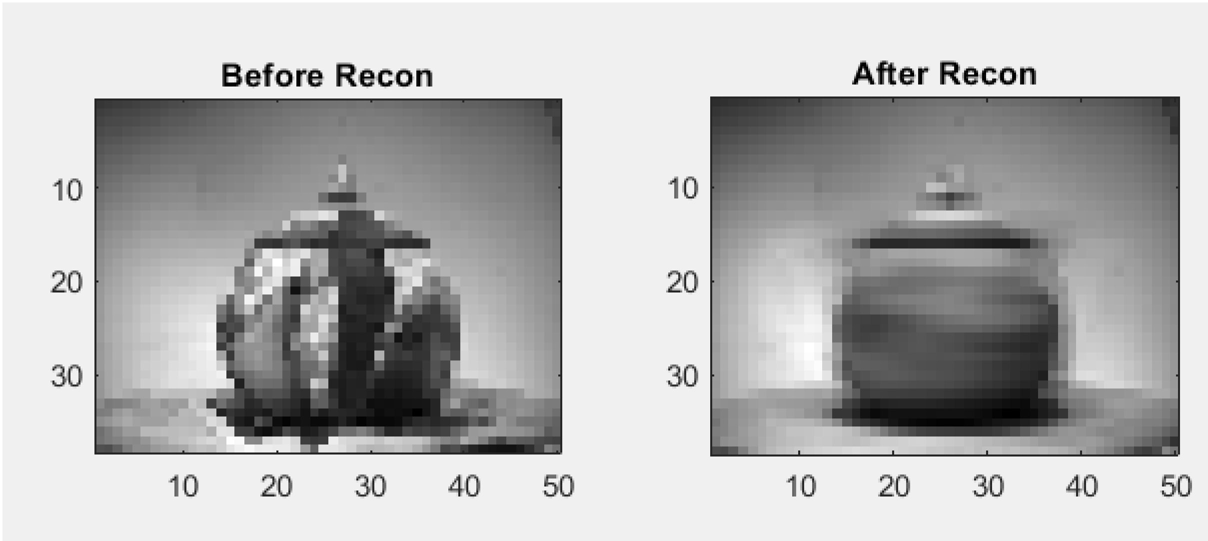
Decode and construct new image $\hat{x}$

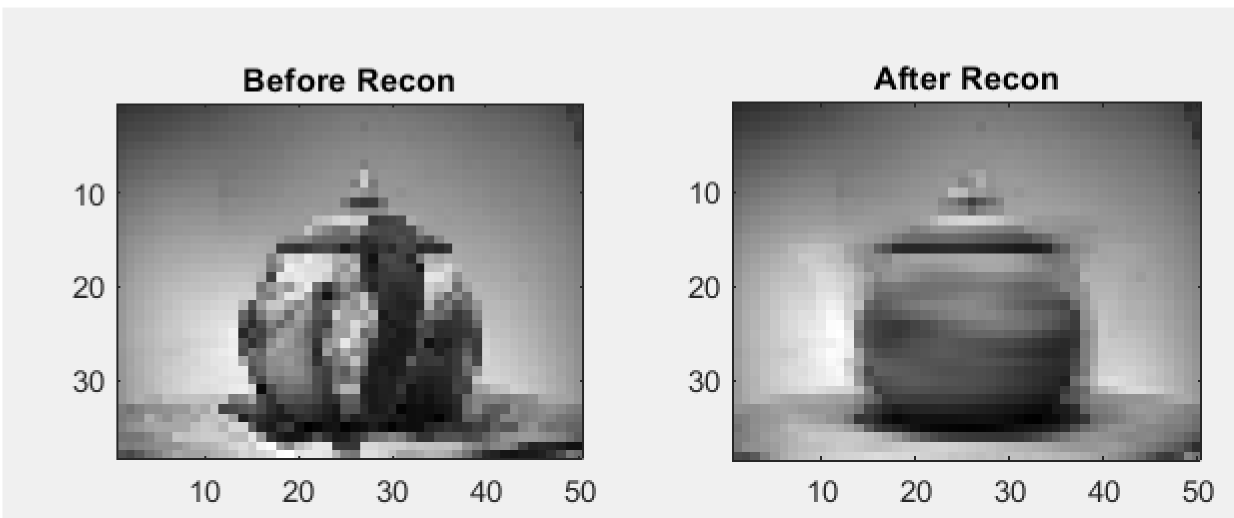
$$\boxed{\hat{X}_i = \mu + \sum_{j=1}^{r} c_{ij} V'_j}$$

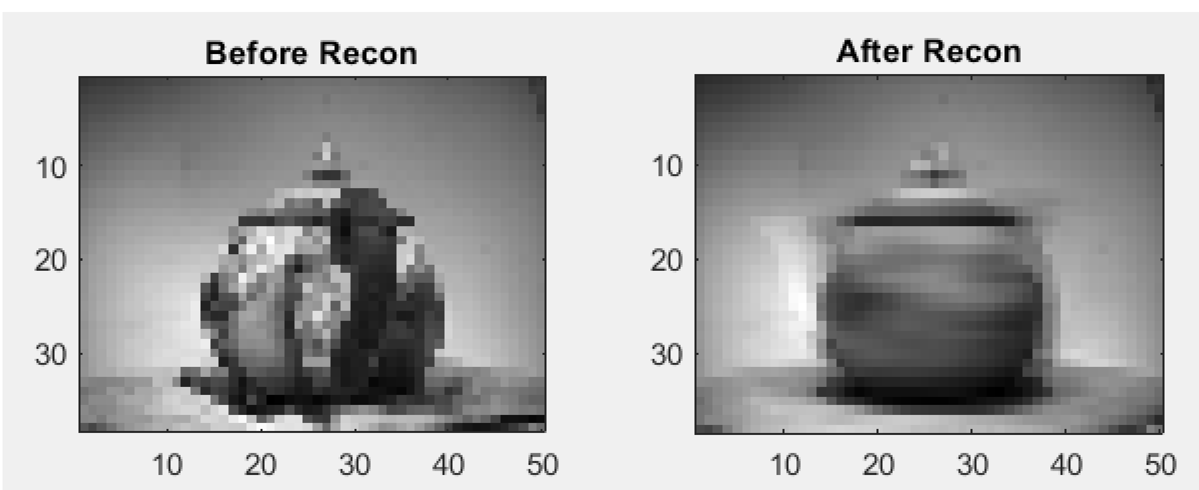
Rewritten → $\hat{X} = \mu + CV'$

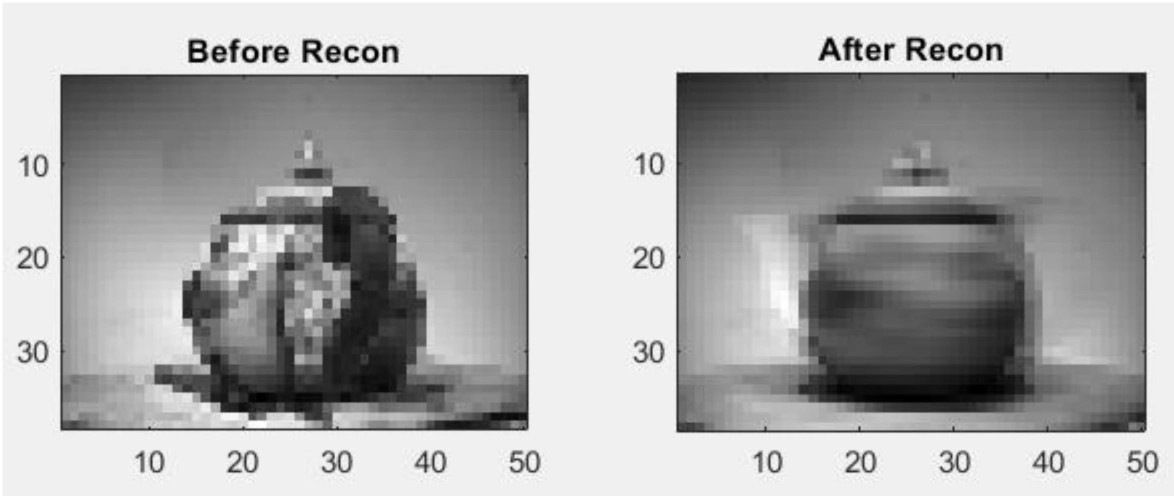
Picked images randomly $i = 10:20$ to reconstruct using PCA. Following are results

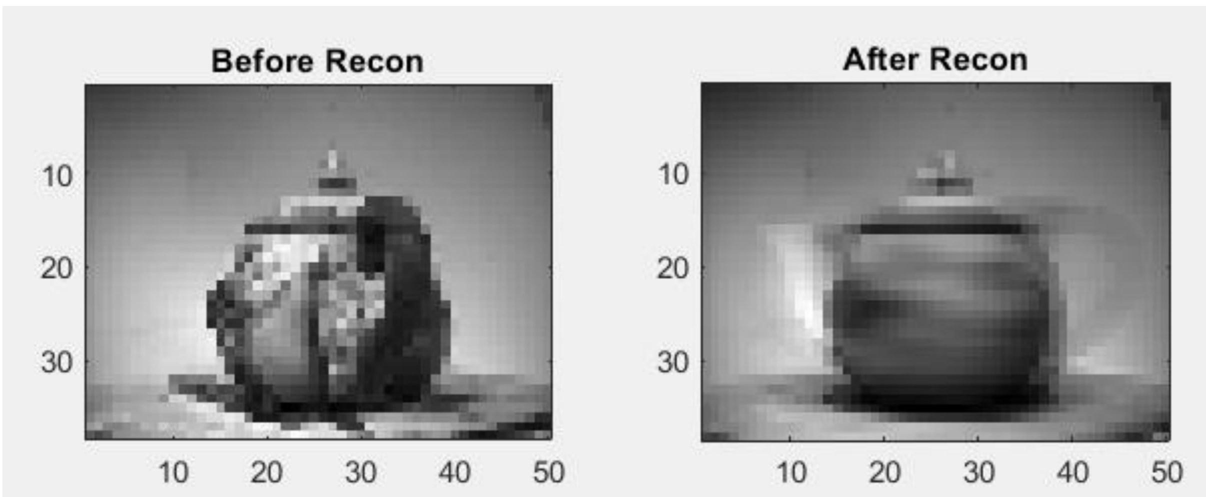


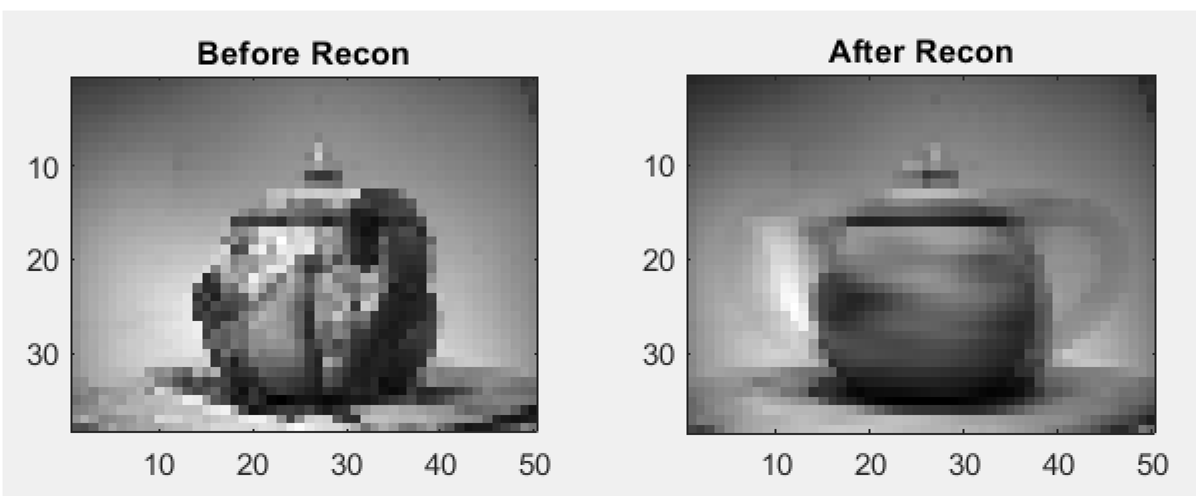

Before Recon
After Recon
10
20
30
10
20
30
40
50

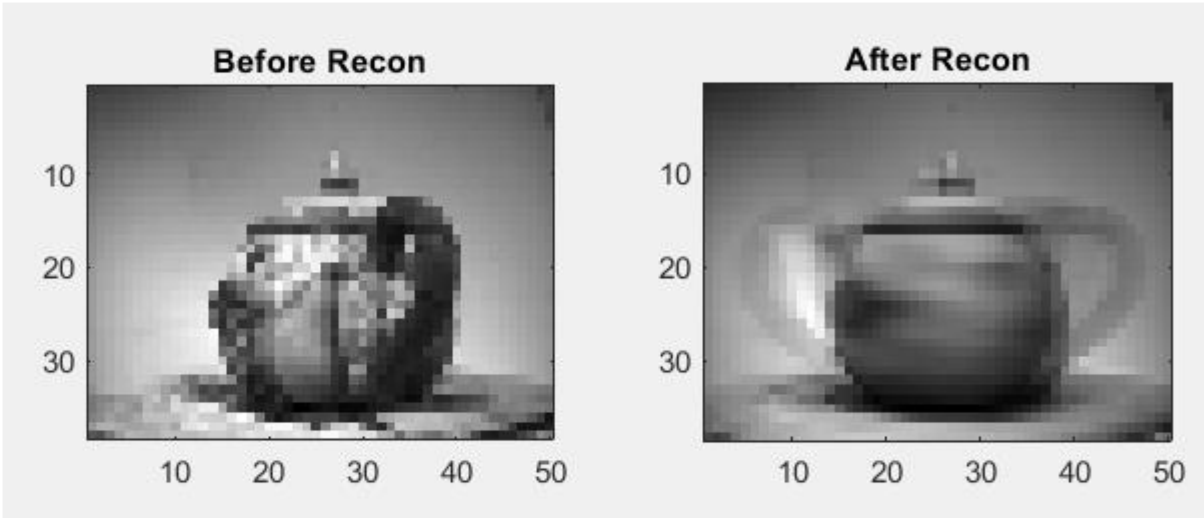

Before Recon
After Recon
10
20
30
10
20
30
40
50

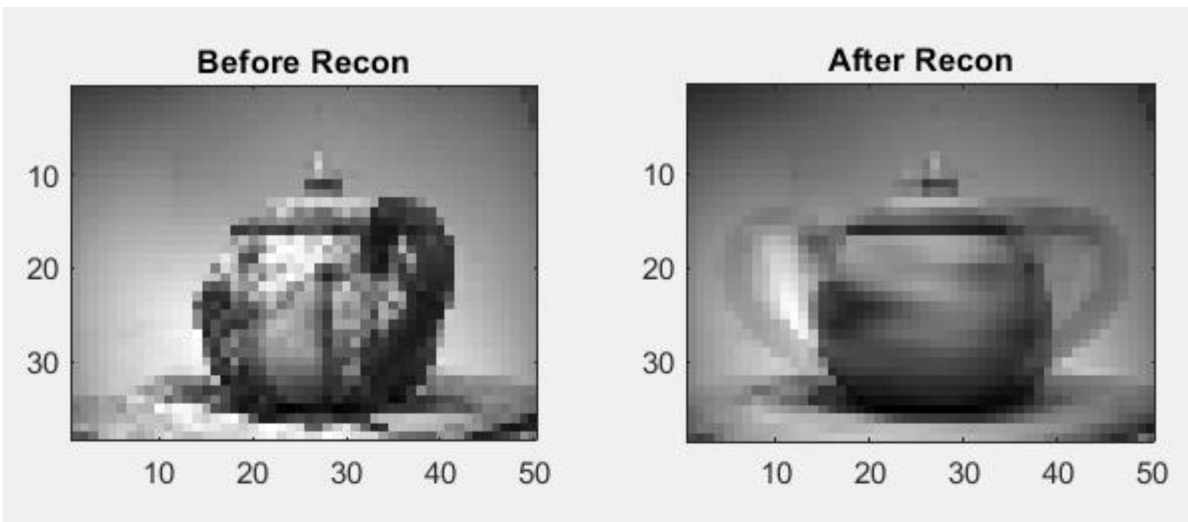

Before Recon
After Recon
10
20
30
10
20
30
40
50

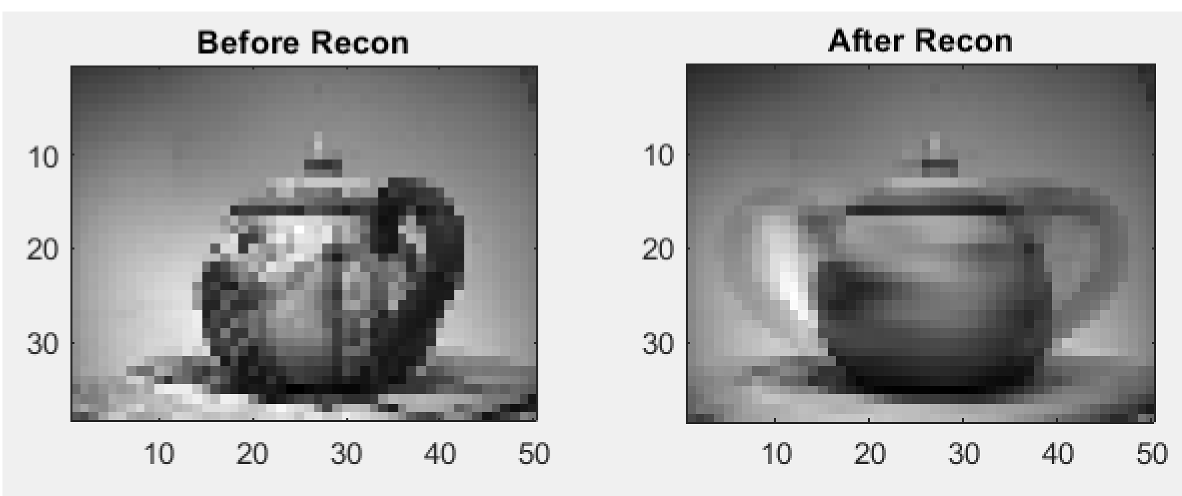

Before Recon
After Recon
10
20
30
10
20
30
40
50


Before Recon
After Recon
10
20
30
10
20
30
40
50


Before Recon
After Recon
10
20
30
10
20
30
40
50


Before Recon
After Recon
10
20
30
10
20
30
40
50


Before Recon
After Recon
10
20
30
10
20
30
40
50

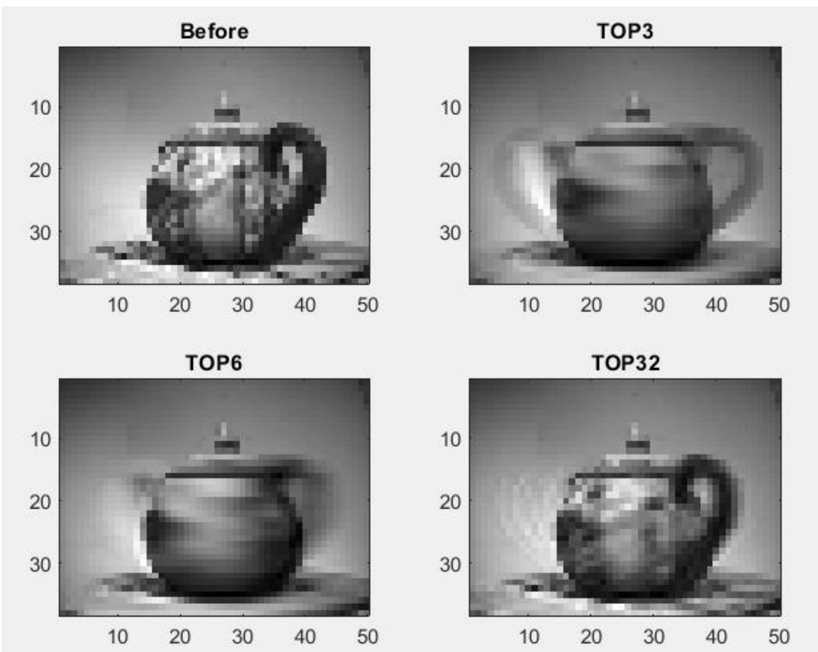

Before Recon
After Recon
10
20
30
10
20
30
40
50

As observed, some images are well reconstructed and distinguishable but many others appear to be not that well reconstructed. Only top 3 components are sharp/aggressive. As per Eigenvalue decomposition, 32 values are more than 0.1, and 6 values are higher than 1. Decoding with more components could get a better reconstruction.

Performance of PCA encoder is determined by computing L-2 norm of X and X hat. The outcome number is 13.6262.

By picking top-6 components, norm value goes down to 9.8303; by picking top 32 component, norm reduces to 3.1382.

Following are snips of image data encode/decode with top 3, top 6 and top 32 most significant components.



Encoding more components into reconstruction gives better outcome with detailing. Deciding on how many most significant components we pick comes down to the requirements of the problem, item in question and how main feature is defined.

Ans 2:

1

VB2182 HW4

(2) Box 1: 8 apples, 4 oranges | Box 2: 10 apples, 2 oranges

Apple selected $\Rightarrow$ Probability of selecting a box

$$P(E_1) = P(E_2) = 1/2$$

A: Selecting Apple from Box

$$P(A/E_1) = 8/12 = 2/3 \; ; \; P(A/E_2) = 10/12 = 5/6$$

Applying Bayes' theorem:

$$P(E_1/A) = \frac{P(E_1)\,P(A/E_1)}{P(E_1)\times P(A/E_1) + P(E_2)\times P(A/E_2)}$$

$$= \frac{\frac{1}{2}\times\frac{2}{3}}{\frac{1}{2}\times\frac{2}{3} + \frac{1}{2}\times\frac{5}{6}} = \frac{1/3}{\frac{1}{3} + \frac{5}{12}} = \frac{1}{1+\frac{5}{4}} = 4/9$$ Ans

Ans 3:

(1) Each class is Gaussian

$\theta = \{\alpha, \mu_1, \Sigma_1, \mu_2, \Sigma_2\}$

Class probability via Bernoulli distribution

$$P(y|\theta) = \alpha^y (1-\alpha)^{1-y}$$

Probability of data $P(x|y,\theta) = \mathcal{N}(x|\mu_y, \Sigma_y)$.

Use max likelihood for α, μ's and Σ's.

~~$\mu_{MLE} = \text{argmax}\, \mathcal{N}($~~

sol Likelihood function $\mathcal{L} = P(x|y,\theta) = \mathcal{N}(x|\mu_y, \Sigma_y)$

For, MLE of Gaussian model, we need to find good estimates of μ & Σ

$$\mu_{MLE} = \underset{\mu}{\text{argmax}}\, \mathcal{N}(x^i|\mu_y, \Sigma_y)$$

$$\Sigma_{MLE} = \underset{\Sigma}{\text{argmax}}\, \mathcal{N}(x|\mu_y, \Sigma_y)$$

we need to get best parameter θ for a Gaussian

$$\theta_{MLE} = \cancel{\text{argument log}}\ \cancel{\text{argmax} \log(\mathcal{N}(x|y,\theta}$$

$$= \underset{\theta}{\text{argmax}} \log(\mathcal{N}(x|y,\theta))$$

$$\sum_{n=1}^{N} \log(\mathcal{N}(x|y,\theta)) = \sum_{n=1}^{N} \log(\mathcal{N}(x|\mu_y, \Sigma_y))$$

Assume Gaussian have diagonal covariance matrices so full covariance matrix Σ gets replaced by a diagonal variance vector σ^2

$$\sum_{n=1}^{N} \log(\mathcal{N}(x|\mu_y, \Sigma_y)) = \sum_{n=1} \log(\mathcal{N}(x|\mu_y, \sigma_y^2))$$

$$\Rightarrow \sum_{n=1}^{N} \log(\mathcal{N}(x|\mu_y, \sigma_y^2)) = \sum_{n=1}^{N} \log\left(\frac{1}{\sqrt{2\pi\sigma_y^2}} \exp^{-\frac{1}{2}\left(\frac{x_n-\mu_y}{\sigma_y}\right)^2}\right)$$

log likelihood $LL = \sum_{n=1}^{N} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{1}{2}\left(\frac{(x_n-\mu_y)^2}{\sigma_y^2}\right)}\right)$

$$= \sum_{n=1}^{N}\left(\log\left(\frac{1}{\sqrt{2\pi\sigma_y^2}}\right) + \log\left(\exp^{-\frac{1}{2}\left(\frac{(x_n-\mu_y)^2}{\sigma_y^2}\right)}\right)\right)$$

$$\Rightarrow LL = \sum_{n=1}^{N}\left(-\log(\sqrt{2\pi\sigma_y^2}) + \left(-\frac{1}{2}\right)\left(\frac{(x_n-\mu_y)^2}{\sigma_y^2}\right)\right)$$

$$= -\frac{1}{2}\sum_{n=1}^{N} \log(2\pi\sigma_y^2) + \sum_{n=1}^{N} -\frac{1}{2}\frac{(x_n-\mu_y)^2}{\sigma_y^2}$$

$$\boxed{LL = -\frac{N}{2}\log(2\pi\sigma_y^2) - \frac{1}{2\sigma_y^2}\sum_{n=1}^{N}(x_n-\mu_y)^2} \quad \text{— (I)}$$

solve for μ_y & σ_y^2 one by one;

$\frac{\partial LL}{\partial \mu} = 0$ for $\underset{\mu}{\operatorname{argmax}} LL(x|\mu_y, \sigma_y^2)$

$$\Rightarrow \frac{\partial LL}{\partial \mu} = \frac{\partial}{\partial \mu}\left[-\frac{N}{2}\log(2\pi\sigma_y^2)\right]^{\nearrow 0} + \frac{\partial}{\partial \mu}\left[-\frac{1}{2\sigma_y^2}\sum_{n=1}^{N}(x_n-\mu_y)^2\right]$$

$$= -\frac{1}{2\sigma_y^2}\sum_{n=1}^{N}\frac{\partial}{\partial \mu}(x_n-\mu_y)^2$$

$$= \frac{\cancel{-}1}{\cancel{2}\sigma_y^2} \times (\cancel{-2}) \sum_{n=1}^{N}(x_n-\mu_y) = 0$$

$$\Rightarrow \sum_{n=1}^{N} x_n - \sum_{n=1}^{N}\mu_y = 0$$

$$\sum_{n=1}^{N} x_n = N\mu_y$$

$$\Rightarrow \boxed{\mu_y = \frac{1}{N}\sum_{n=1}^{N} x_n} \quad \text{— (II)}$$

MLE of σ^2

$$\frac{\partial LL}{\partial \sigma_y^2} = \frac{\partial}{\partial \sigma_y^2}\left(-\frac{N}{2}\log(2\pi\sigma_y^2)\right) + \frac{\partial}{\partial \sigma_y^2}\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu_y)^2\right)$$

$$= \frac{-N \times \cancel{2\pi}}{2\times(\cancel{2\pi}\sigma_y^2)} + \sum_{n=1}^{N}\frac{-1\times(-1)}{2\sigma_y^2\cdot\sigma_y^2}(x_n-\mu_y)^2$$

$$= \frac{-N}{2\sigma_y^2} + \frac{1}{2\sigma_y^4}\sum_{n=1}^{N}(x_n-\mu_y)^2 = 0$$

$$\Rightarrow \boxed{\sigma_y^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_y)^2}$$

$$\Rightarrow \boxed{\sigma_y = \left[\frac{\sum(x_n-\mu_y)^2}{N}\right]^{1/2}} \quad \text{— III}$$

now $P(y/\theta) = \alpha^y(1-\alpha)^{1-y}$; where $y = 0, 1$

0 ; everywhere else

$$L(y/\theta) = \prod_{i=1}^{N}\alpha^{y_i}(1-\alpha)^{1-y_i}$$

$$\log L(y/\theta) = \sum_{i=1}^{N}\log\alpha^{y_i} + \log(1-\alpha)^{1-y_i}$$

$$\boxed{LL(\theta) = \sum_{i=1}^{N}y_i(\log\alpha) + \sum_{i=1}^{N}(1-y_i)\log(1-\alpha) = \log\alpha\sum_{i=1}^{N}y_i + \left(N-\sum_{i=1}^{N}y_i\right)\log(1-\alpha)}$$

$\cancel{\frac{\partial LL(\alpha)}{\partial\alpha}}$ = Let say $\sum_{i=1}^{N}y_i = Y$

So $\boxed{LL(\theta) = (\log\alpha)Y + (N-Y)\log(1-\alpha)}$ — IV

$$\frac{\partial LL(\alpha)}{\partial\alpha} = Y\times\frac{1}{\alpha} + (N-Y)\frac{-1}{1-\alpha} = 0$$

$$\Rightarrow \boxed{\hat{\alpha} = \frac{Y}{N} = \frac{\sum_{i=1}^{N}y_i}{m}} \quad \text{— V}$$

PDF, ~~Bayes Naive~~ ~~Lecture~~ ~~Probability~~ =)

~~[illegible]~~ Decision boundary is set of pts x;

$$P(Y=1 | X=x) = P(Y=0 | X=x)$$

For 2D Gaussian $N(\mu_y, \Sigma_y)$ class conditional feature distribution $P(X=x | Y=y)$

$$P(X=x | Y=y) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_y|}} \exp\left(-\frac{(x-\mu_y)\Sigma_y^{-1}(x-\mu_y)'}{2}\right)$$

$$\frac{P(Y=1 | X=x)}{P(Y=0 | X=x)} = \frac{P(X=x | Y=1)P(Y=1)}{P(X=x | Y=0)P(Y=0)}$$

$$\boxed{= \sqrt{\frac{|\Sigma_1|}{|\Sigma_2|}} \exp\left(-\frac{(x-\mu_2)\Sigma_2^{-1}(x-\mu_2)'}{2} + \frac{(x-\mu_1)\Sigma_1^{-1}(x-\mu_1)'}{2}\right] \frac{\theta}{1-\theta}} \quad \text{— (VI)}$$

(VI) is a quadratic equation in x

But if $\Sigma_1 = \Sigma_2$, then quadratic part cancels out, and decision boundary becomes linear

=) For $\Sigma_1 = \Sigma_2$, (VI) become

$$= 1 \times \exp\left[\frac{-(x-\mu_2)\Sigma_1^{-1}(x-\mu_2)' + (x-\mu_1)\Sigma_1^{-1}(x-\mu_1)'}{2}\right] \frac{\theta}{1-\theta}$$

$x^2\ \Sigma_1^{-1}$ terms cancel out leaving maximum x degree to be 1

$$= \exp\left[\frac{-x\Sigma_1^{-1}(x-\mu_2) + \mu_2}{2}\right]$$

(3) $\theta = \{\alpha, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$

$P(y|\theta) = \alpha^y (1-\alpha)^{1-y}$; $P(x|y,\theta) = N(x|\mu_y, \Sigma_y)$

Max likelihood estimator θ_{MLE}

$$L(\theta) = \prod_{i=1}^{N} p(x_i, y_i|\theta) = p(data|\theta)$$

$$l(\theta) = \log(L(\theta)) = \log(p(data|\theta)) = \sum_{i=1}^{N} \log(x_i, y_i|\theta)$$

$$= \sum_{i=1}^{N} \log p(y_i|\theta) + \sum_{i=1}^{N} \log(p(x_i|y_i, \theta)) \quad [\text{Bayes rule}]$$

$$l(\theta) = \underbrace{\sum_{i=1}^{N} \log(p(y_i|\alpha))}_{\text{Part 1}} + \underbrace{\sum_{y_i \in 0} \log(p(x_i|\mu_1, \Sigma_1))}_{\text{Part 2}} + \underbrace{\sum_{y_i \in 1} \log(p(x_i|\mu_2, \Sigma_2))}_{\text{Part 3}}$$

Part 1: $\sum_{i=1}^{N} \log p(y_i|\alpha) = \sum_{i=1}^{N} \log \alpha^{y_i} \cdot (1-\alpha)^{1-y_i} \equiv l(\alpha)$

$$\frac{\partial}{\partial \alpha} l(\alpha) = 0 = \frac{\partial}{\partial \alpha}\left(\sum_{i=1}^{N} y_i \log \alpha + (1-y_i)\log(1-\alpha)\right) = 0$$

$$\frac{\partial}{\partial \alpha} \sum_{i \in 1} \log \alpha + \sum_{i \in 0} \log(1-\alpha) = 0 \Rightarrow \frac{N_1}{\alpha} - \frac{N_0}{1-\alpha} = 0$$

$$\boxed{\alpha_{MLE} = \frac{N_1}{N_1 + N_0}}$$

$$\vec{\mu_1} = \mu_1 ; \vec{\mu_2} = \mu_2$$

Part 2 $\quad \partial(\mu_1, \Sigma_1) = \sum_{y_i \in 0} \log(p(x_i|\mu_1, \Sigma_1))$

$$l(\mu_1, \Sigma_1) = \sum_{\substack{i=1 \\ y_i \in 0}}^{N} \log \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_1|}} \exp\left(-\tfrac{1}{2}(\vec{x_i} - \vec{\mu_1})^T \Sigma_1^{-1} (\vec{x_i} - \vec{\mu})\right)$$

$$\frac{\partial}{\partial \mu_1} l(\mu_1, \Sigma_1) = \frac{\partial}{\partial \mu_1}\left(\sum_{\substack{y_i \in 0 \\ i=1}}^{N} -D/2 \log 2\pi - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(\vec{x_i} - \vec{\mu_1})^T \Sigma_1^{-1} (\vec{x_i} - \vec{\mu_1})\right) = 0$$

$$\Rightarrow \sum_{\substack{i=1 \\ y_i \in 0}}^{N} (\vec{x_i} - \mu_1)^T \Sigma^{-1} = 0 \Rightarrow \boxed{\mu_1 = \frac{1}{N_0} \sum_{y_i \in 0} \vec{x_i}}$$

$$\frac{\partial \ell(\mu_1, \Sigma_1)}{\partial \Sigma_1} = 0 = \frac{\partial}{\partial \Sigma_1}\left(\sum_{\substack{i=1 \\ y_i \in 0}} -\frac{D}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_1| - \underbrace{\frac{1}{2}(\vec{x}_i - \vec{\mu}_1)^T \Sigma_1^{-1}(\vec{x}_i - \vec{\mu}_1)}_{EQ1}\right) = 0$$

Say $A_1 = \Sigma_1^{-1}$

$$\frac{\partial}{\partial A_1} * \left(-\frac{N_0 D}{2}\log 2\pi + \frac{N_0}{2}\log|A_1| - \frac{1}{2}\sum_{\substack{i=1 \\ y_i \in 0}}^{N} t_r(EQ1)\right) = 0$$

$$\left.\begin{array}{l} \frac{\partial \log(|A|)}{\partial A} = (A^{-1})^T \;\Big|\; \frac{\partial\, tr[BA]}{\partial A} = B^T \\ \{\text{Applying } xx^T A = tr(x^T A x) = \vec{x} A x \end{array}\right\} \to \text{Equation FIRST}$$

$$\frac{\partial}{\partial A_1}\left(-\frac{N_0 D \log 2\pi}{2} + \frac{N_0}{2}\log(A_1) - \frac{1}{2}\sum_{\substack{i=1 \\ y_i \in 0}}^{N} tr\left[(\vec{x}_i - \vec{\mu}_1)(\vec{x}_i - \vec{\mu}_1)^T A_1\right] = 0\right.$$

$$\frac{\partial \ell(\mu_1, \Sigma_1)}{\partial A_1} = 0 \Rightarrow -0 + \frac{N_0}{2}(A_1^{-1})^T - \frac{1}{2}\sum_{\substack{i=1 \\ y_i \in 0}}^{N}\left[(\vec{x}_i - \vec{\mu}_1)(\vec{x}_i - \vec{\mu}_1)^T\right]^T = 0$$

$$\Rightarrow \frac{N_0}{2}\Sigma_1 - \frac{1}{2}\sum_{\substack{i=1 \\ y_i \in 0}}^{N}(\vec{x}_i - \vec{\mu}_1)(\vec{x}_i - \vec{\mu}_1)^T = 0$$

$$\Rightarrow \boxed{\Sigma_1 = \frac{1}{N_0}\sum_{y_i \in 0}(\vec{x}_i - \vec{\mu}_1)(\vec{x}_i - \vec{\mu}_1)^T}$$

~~Similar~~ Also, in a similar way,

$$\boxed{\Sigma_2 = \frac{1}{N_1}\sum_{y_i \in 1}(\vec{x}_i - \vec{\mu}_2)\cdot(\vec{x}_i - \vec{\mu}_2)}$$

$$\boxed{\mu_2 = \frac{1}{N_1}\sum_{y_i \in 1}\vec{x}_i}$$

Bayes' Optimal decision

$\hat{y} = \underset{y \in \{0,1\}}{\operatorname{argmax}}\; p(y|x)$ which is $\hat{y} = 0.5$ as decision boundary

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x,y)}{\sum_y p(x,y)} = \frac{p(x,y)}{p(x,y=0) + p(x,y=1)}$$

$$p(y=1|x) = \frac{p(x,y)=1}{p(x,y=0) + p(x,y=1)}$$

$$= \underline{\alpha N(x|\mu_2, \Sigma_2)}$$

$$(1-\alpha)\,N(x|\mu_1,\Sigma_1)+\alpha N(x|\mu_2,\Sigma_2)$$

3

Case 1st: $\Sigma_1=\Sigma_2=\Sigma$

$$p(y=1|x)=p=0.5=\alpha\cdot \underbrace{N(x|\mu_2,\Sigma)}_{N_2^*}$$

$$(1-\alpha)\underbrace{N(x|\mu_1,\Sigma)}_{N_1^*}+\alpha N(x|\mu_2,\Sigma)$$

$$\left(\frac{1}{2}-\frac{\alpha}{2}\right)N_1^*+\frac{\alpha}{2}N_2^*=\alpha N_2^*$$

$$(1-\alpha)N_1^*=\alpha N_2^*$$

$$\Rightarrow \log(1-\alpha)+\log N_1^*=\log\alpha+\log N_2^*$$

Assume $\log(1-\alpha)=c_1$ and $\log\alpha=c_2$; $c_1-c_2=c$

$$\boxed{c+\log N_1^*=\log N_2^*}$$

$$\Rightarrow c+\log\left(\frac{1}{2\pi^{D/2}\sqrt{|\Sigma|}}\right)-\frac{1}{2}\sum_i^N(\vec{x}_i-\mu_1)\Sigma^{-1}(\vec{x}_i-\mu_1)^T$$

$$=\log\left(\frac{1}{2\pi^{D/2}\sqrt{|\Sigma|}}\right)-\frac{1}{2}\sum_{i=1}^N(\vec{x}_i-\mu_2)\Sigma^{-1}(\vec{x}_i-\mu_2)^T$$

$$\Rightarrow c=\frac{1}{2}\sum_{i=1}^N\left((\vec{x}_i-\mu_1)\Sigma^{-1}(\vec{x}_i-\mu_1)^T-(\vec{x}_i-\mu_2)\Sigma^{-1}(\vec{x}_i-\mu_2)^T\right)$$

Apply trace formula

$$\Rightarrow c=\frac{1}{2}\sum_{i=1}^N tr\left(\left((\vec{x}_i-\mu_1)(\vec{x}_i-\mu_1)^T-(\vec{x}_i-\mu_2)(\vec{x}_i-\mu_2)^T\right)\Sigma^{-1}\right)$$

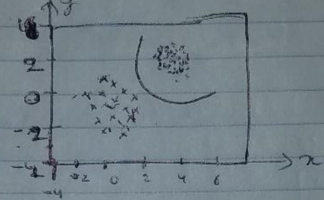
$$\Rightarrow c=\frac{1}{2}\sum_{i=1}^N tr\left(\left(2(\mu_2-\mu_1)\vec{x}_i+\mu_1^2-\mu_2^2\right)\Sigma^{-1}\right)\Big]$$ linear eq in x_i

Case 2nd, when $\Sigma_1\neq\Sigma_2$ then $x_i x_i^T$ will have Σ_1 & Σ_2 coefficients, which won't cancel each other. So, the final equation is quadratic in terms of x_i in decision boundary.

4

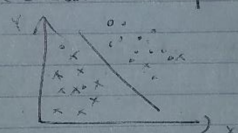
⇒ When covariances are equal, distribution of both the classes is similar and as a result, they can be linearly dissected.

When covariances are unequal, one class dominates the other, has a wider distribution, while the other has a more compact distribution:



If covariances are different, quadratic decision boundary

Linear decision example



Data classes {x, ·} have same covariances (uniform distribution) and hence have linear decision boundary