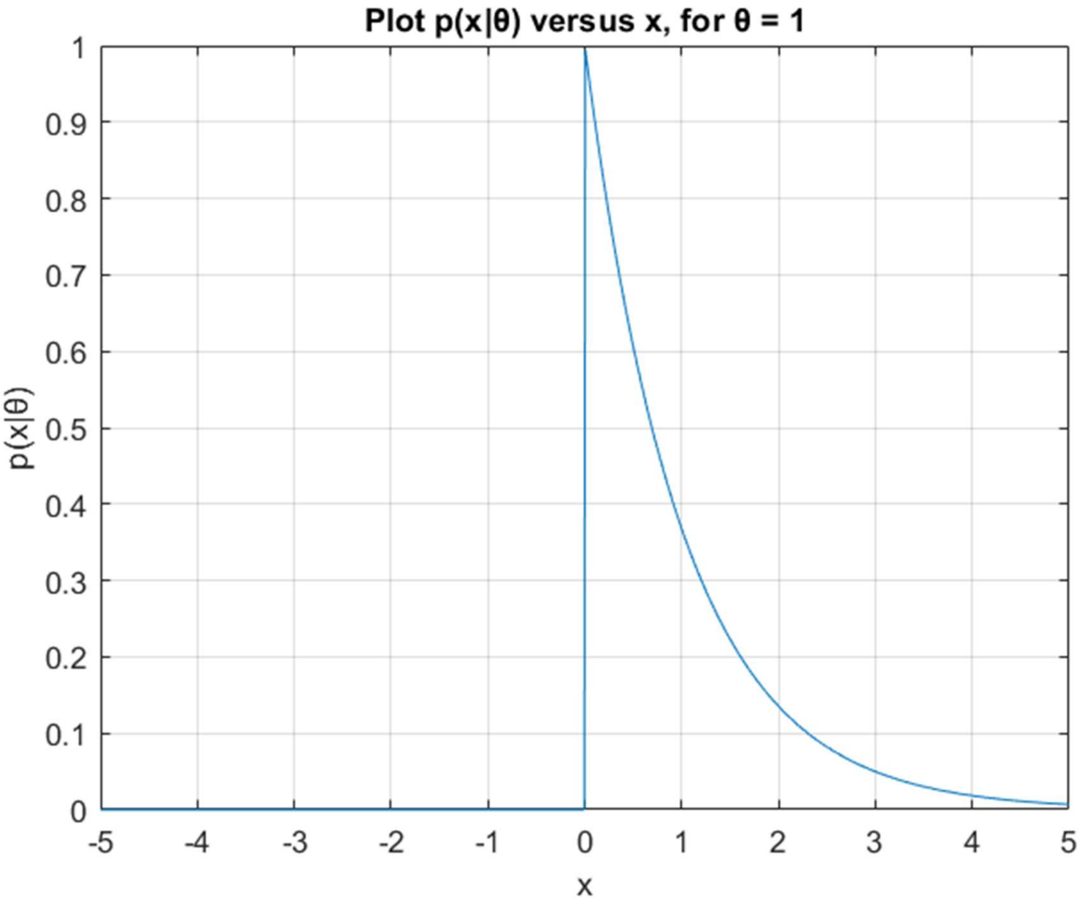
PATTERN RECOGNITION – Homework 2

Solution

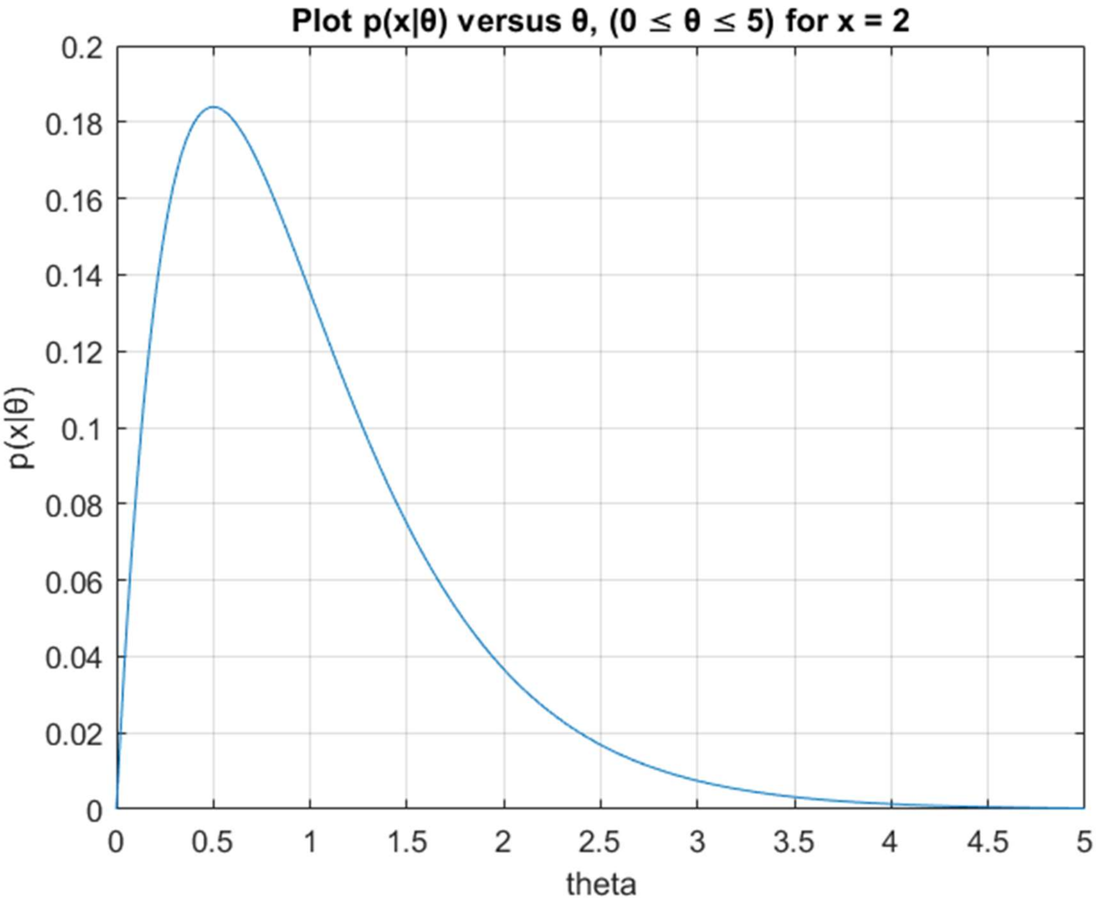
Problem 1.

a/

```
%% a. Plot p(x|?) versus x, for ? = 1
theta = 1;
x = linspace(-5, 5, 1000);
px_theta = zeros(size(x));
px_theta(x>=0) = theta * exp(-theta * x(x>=0));
figure;
plot(x, px_theta);
grid on;
title("Plot p(x|?) versus x, for ? = 1");
xlabel("x"); ylabel("p(x|?)");

%% a. Plot p(x|?) versus ?, (0 ? ? ? 5) for x = 2
theta = linspace(0, 5, 1000);
px_theta = zeros(size(theta));
px_theta = theta .* exp(-theta * 2);
figure;
plot(theta, px_theta);
grid on;
title("Plot p(x|?) versus ?, (0 ? ? ? 5) for x = 2");
xlabel("theta"); ylabel("p(x|?)");
```



Plot p(x|θ) versus x, for θ = 1
p(x|θ)
x
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-5
-4
-3
-2
-1
0
1
2
3
4
5



b/

Given data samples $D = \{x_1, x_2, \ldots, x_n\}$ drawn independently

We have: $p(D|\theta) = p(x_1|\theta) . p(x_2|\theta) \ldots p(x_n|\theta)$

$\Rightarrow$ log-likelihood:

$$l(\theta) = \ln p(D|\theta) = \sum_{k=1}^{n} \ln p(x_k|\theta)$$

$$\Rightarrow \nabla_\theta l(\theta) = \nabla_\theta \left[\sum_{k=1}^{n} \ln p(x_k|\theta)\right] = \sum_{k=1}^{n} \nabla_\theta \ln p(x_k|\theta)$$

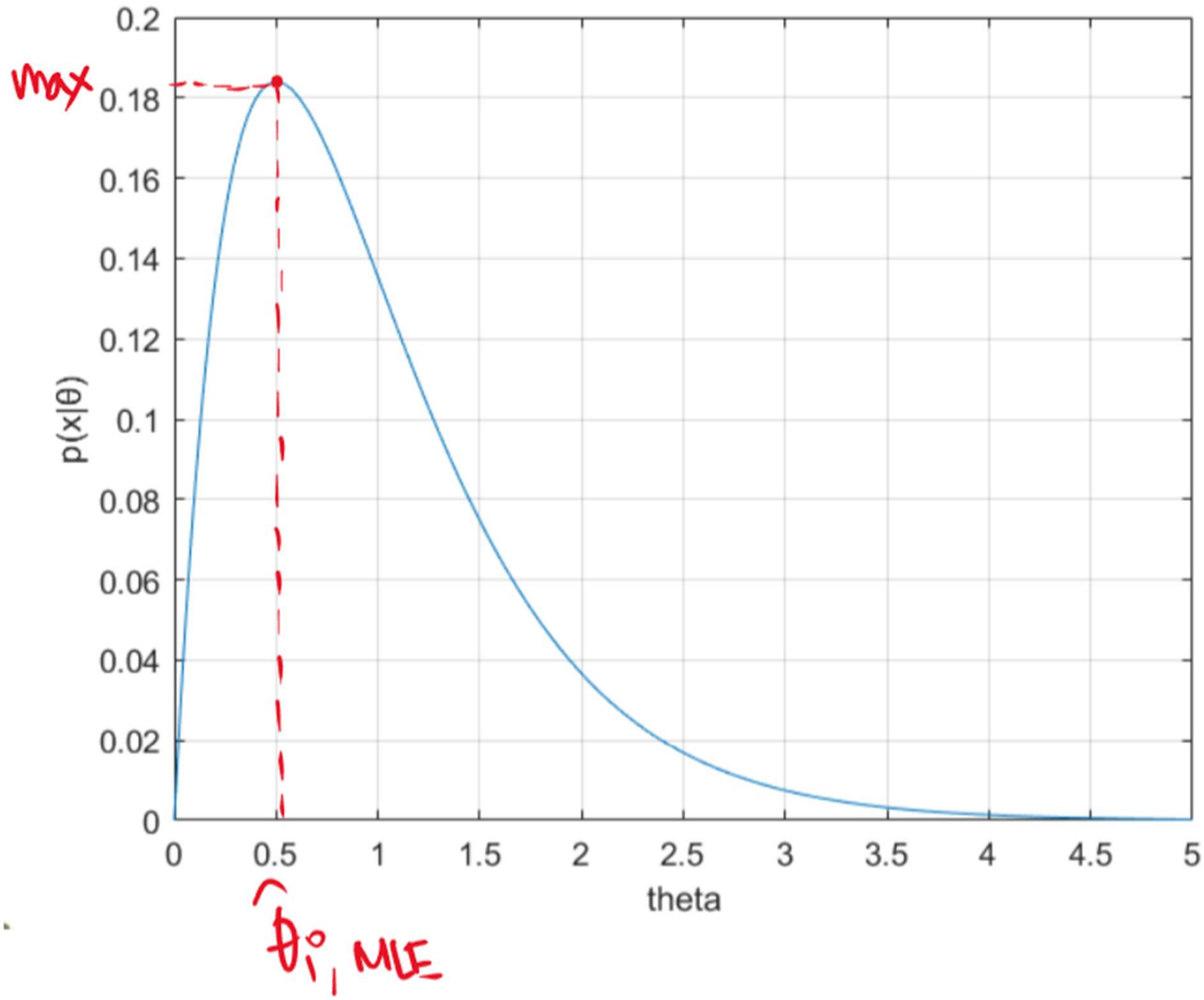
$$= \sum_{k=1}^{n} \nabla_\theta \left[\ln\theta - \theta . x_k\right] = \sum_{k=1}^{n} \left[\frac{1}{\theta} - x_k\right]$$

$$= \frac{n}{\theta} - \sum_{k=1}^{n} x_k$$

$\hat{\theta}_{MLE}$ can be obtained by solving $\nabla_\theta l(\theta) = 0$.

Then, $\dfrac{n}{\hat{\theta}} - \sum_{k=1}^{n} x_k = 0 \Rightarrow \hat{\theta} = \dfrac{n}{\sum_{k=1}^{n} x_k} = \dfrac{1}{\frac{1}{n}\sum_{k=1}^{n} x_k}$

c/



Problem 2.

a/

$$y_i = w^T x_i \quad \rightarrow \mu_i = E[w^T x_i] = w^T \mu_i$$

$$\sigma_i^2 = E\left[(y_i - \mu_i)^2\right] = w^T \Sigma_i w$$

To obtain w that maximize $J_1(w)$, we take derivative of $J_1(w)$ w.r.t w by chain rule and set to 0:

$$\frac{\partial J_1}{\partial w} = \frac{\partial J_1}{\partial \mu_1} \cdot \frac{\partial \mu_1}{\partial w} + \frac{\partial J_1}{\partial \mu_2} \cdot \frac{\partial \mu_2}{\partial w} + \frac{\partial J_1}{\partial \sigma_1^2} \cdot \frac{\partial \sigma_1^2}{\partial w} + \frac{\partial J_1}{\partial \sigma_2^2} \cdot \frac{\partial \sigma_2^2}{\partial w} = 0$$

where:

$$\frac{\partial J_1}{\partial \mu_1} = \frac{2(\mu_1 - \mu_2)}{\sigma_1^2 + \sigma_2^2} \quad ; \quad \frac{\partial \mu_1}{\partial w} = \mu_1$$

$$\frac{\partial J_1}{\partial \mu_2} = -\frac{2(\mu_1 - \mu_2)}{\sigma_1^2 + \sigma_2^2} \quad ; \quad \frac{\partial \mu_2}{\partial w} = \mu_2$$

$$\frac{\partial J_1}{\partial \sigma_1^2} = -\frac{(\mu_1 - \mu_2)^2}{(\sigma_1^2 + \sigma_2^2)^2} \quad ; \quad \frac{\partial \sigma_1^2}{\partial w} = (\Sigma_1 + \Sigma_1^T)w = 2\Sigma_1 w$$

$$\frac{\partial J_1}{\partial \sigma_2^2} = -\frac{(\mu_1 - \mu_2)^2}{(\sigma_1^2 + \sigma_2^2)^2} \quad ; \quad \frac{\partial \sigma_2^2}{\partial w} = (\Sigma_2 + \Sigma_2^T)w = 2\Sigma_2 w$$

Then,

$$\frac{\partial J_1}{\partial w} = \frac{2(\mu_1 - \mu_2)}{\sigma_1^2 + \sigma_2^2} \cdot (\mu_1 - \mu_2) - \frac{2(\mu_1 - \mu_2)^2}{(\sigma_1^2 + \sigma_2^2)^2}(\Sigma_1 + \Sigma_2)w = 0$$

$$\Rightarrow w = \frac{\sigma_1^2 + \sigma_2^2}{\mu_1 - \mu_2} \cdot (\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2)$$

Since this is constant & doesn't affect the direction of separation, $J_1(w)$ is maximized by $w = (\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2)$

b/

$$\frac{\partial J_2}{\partial w} = \frac{\partial J_2}{\partial \mu_1}\cdot\frac{\partial \mu_1}{\partial w} + \frac{\partial J_2}{\partial \mu_2}\cdot\frac{\partial \mu_2}{\partial w} + \frac{\partial J_2}{\partial \sigma_1^2}\cdot\frac{\partial \sigma_1^2}{\partial w} + \frac{\partial J_2}{\partial \sigma_2^2}\cdot\frac{\partial \sigma_2^2}{\partial w} = 0$$

$$\frac{\partial J_2}{\partial \mu_1} = \frac{2(\mu_1 - \mu_2)}{P(\omega_1)\sigma_1^2 + P(\omega_2)\sigma_2^2} \quad ; \quad \frac{\partial \mu_1}{\partial w} = \mu_1$$

$$\frac{\partial J_2}{\partial \mu_2} = -\frac{2(\mu_1 - \mu_2)}{P(\omega_1)\sigma_1^2 + P(\omega_2)\sigma_2^2} \quad ; \quad \frac{\partial \mu_2}{\partial w} = \mu_2$$

$$\frac{\partial J_2}{\partial \sigma_1^2} = -\frac{P(\omega_1)(\mu_1 - \mu_2)}{(P(\omega_1)\sigma_1^2 + P(\omega_2)\sigma_2^2)^2} \quad ; \quad \frac{\partial \sigma_1^2}{\partial w} = 2\Sigma_1 w$$

$$\frac{\partial J_2}{\partial \sigma_2^2} = -\frac{P(\omega_2)(\mu_1 - \mu_2)}{(P(\omega_1)\sigma_1^2 + P(\omega_2)\sigma_2^2)^2} \quad ; \quad \frac{\partial \sigma_2^2}{\partial w} = 2\Sigma_2 w$$

Then,

$$\frac{\partial J_2}{\partial w} = \frac{2(\mu_1 - \mu_2)}{P(\omega_1)\sigma_1^2 + P(\omega_2)\sigma_2^2}\cdot(\mu_1 - \mu_2) - \frac{2(\mu_1 - \mu_2)^2}{(P(\omega_1)\sigma_1^2 + P(\omega_2)\sigma_2^2)^2}\cdot\left(P(\omega_1)\Sigma_1 + P(\omega_2)\Sigma_2\right)w = 0$$

$$\Rightarrow w = \frac{P(\omega_1)\sigma_1^2 + P(\omega_2)\sigma_2^2}{\mu_1 - \mu_2}\cdot\left(P(\omega_1)\Sigma_1 + P(\omega_2)\Sigma_2\right)^{-1}\cdot(\mu_1 - \mu_2)$$

doesn't affect the direction of separation

$\Rightarrow J_2(w)$ is maximized by $w = \left(P(\omega_1)\Sigma_1 + P(\omega_2)\Sigma_2\right)^{-1}\cdot(\mu_1 - \mu_2)$

c/

WITHIN-CLASS SCATTER

Thus, $(1/n)(\tilde{s}_1^2 + \tilde{s}_2^2)$ is an estimate of the variance of the pooled data, and $\tilde{s}_1^2 + \tilde{s}_2^2$ is called the total *within-class scatter* of the projected samples. The *Fisher linear discriminant* employs that linear function $\mathbf{w}^t\mathbf{x}$ for which the criterion function

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \tag{96}$$

The $J_1(w) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$ is more closely related to the eq. 96 since for $J_1(w)$, we use exact mean & variance while in eq. 96, we use sample data.

Problem 3.

a/

We have $m_1 = \frac{1}{n_1}\sum_i y_i$; $m_2 = \frac{1}{n_2}\sum_j y_j$;

$$s_1^2 = \sum_i (y_i - m_1)^2 \ ; \ s_2^2 = \sum_i (y_j - m_2)^2 .$$

Eq1. $$J_1 = \frac{1}{n_1 n_2}\sum_{y_i \in Y_1}\sum_{y_j \in Y_2}(y_i - y_j)^2 = \frac{1}{n_1 n_2}\sum_i\sum_j (y_i^2 - 2y_i y_j + y_j^2)$$

$$= \frac{1}{n_1 n_2}\left[\sum_i\sum_j y_i^2 - 2\sum_i\sum_j y_i y_j + \sum_i\sum_j y_j^2\right]$$

$$= \frac{1}{n_1 n_2}\cdot\left[\sum_i n_2 y_i^2 - 2\sum_i y_i \sum_j y_j + n_1\sum_j y_j^2\right]$$

$$= \frac{1}{n_1}\sum_i y_i^2 - \frac{2}{n_1 n_2}\sum_i y_i \sum_j y_j + \frac{1}{n_2}\sum_j y_j^2 \quad (*)$$

Eq2. $$J_1 = (m_1 - m_2)^2 + \frac{1}{n_1}s_1^2 + \frac{1}{n_2}s_2^2$$

$$= m_1^2 - 2m_1 m_2 + m_2^2 + \frac{1}{n_1}\sum_i (y_i - m_1)^2 + \frac{1}{n_2}\sum_j (y_j - m_2)^2$$

$$= m_1^2 - 2m_1 m_2 + m_2^2 + \frac{1}{n_1}\sum_i y_i^2 - \frac{2}{n_1}\sum_i y_i m_1 + \frac{1}{n_1}\sum_i m_1^2 + \dots$$

$$\dots + \frac{1}{n_2}\sum_j y_j^2 - \frac{2}{n_2}\sum_j y_j m_2 + \frac{1}{n_2}\sum_j m_2^2$$

$$= m_1^2 - 2m_1 m_2 + m_2^2 + \frac{1}{n_1}\sum_i y_i^2 - 2m_1^2 + m_1^2 + \dots$$

$$\dots + \frac{1}{n_2}\sum_j y_j^2 - 2m_2^2 + m_2^2$$

$$= \frac{1}{n_1}\sum_i y_i^2 + \frac{1}{n_2}\sum_j y_j^2 - 2m_1m_2$$

$$= \frac{1}{n_1}\sum_i y_i^2 + \frac{1}{n_2}\sum_j y_j^2 - 2.\frac{1}{n_1n_2}\sum_i y_i \sum_j y_j \quad (**)$$

Since $(*)$ & $(**)$ are equivalent, we can conclude that

$J_1 = \frac{1}{n_1 n_2}\sum_{y_i \in Y_1}\sum_{y_j \in Y_2}\left(y_i - y_j\right)^2$ can be written as $J_1 = \left(m_1 - m_2\right)^2 + \frac{1}{n_1}s_1^2 + \frac{1}{n_2}s_2^2$

b/

c/

Problem 4.

a/

```
%%
function w = fisher_linear_discriminant(class_1, class_2)
    d = size(class_1, 2);
    n1 = length(class_1);
    n2 = length(class_2);
    % Mean vectors
    m1 = mean(class_1); % 1xd
    m2 = mean(class_2);
    % Within-class scatter
    S1 = calculate_Si(class_1);
    S2 = calculate_Si(class_2);
    Sw =  S1 + S2;
    % Between-class scatter
    Sb = (m1' - m2') * (m1' - m2')'; % m->dx1
    disp(Sw);
    disp(Sb);
    % Optimal direction w
    [V, D] = eig(Sw^-1 * Sb); % solve eigenvalue problem
    [~, idx] = max(diag(D)); % find the index of largest eigenvalue
    w = V(:, idx); % optimal direction is the eigenvector, where eigenvalue
is largest
    w = w/norm(w); % normalize
    w2 = Sw^-1 * (m1'-m2');
    w2 = w2/norm(w2);
    disp(w2);
end

function Si = calculate_Si(data)
    d = size(data, 2); % dim
    Si = zeros(d, d);
    m = mean(data);
    for i=1:length(data)
```

```
        Si = Si + (data(i,:)'-m')*(data(i,:)'-m')';
    end
end

function proj_data = project_data(data, direction)
    w = direction; % dx1 => w' in 1xd
    proj_data = zeros(size(data)); %nxd
    for i=1:length(data)
        proj_data(i, :) = dot(data(i, :), w') / norm(w')^2 * w';
    end
end
```

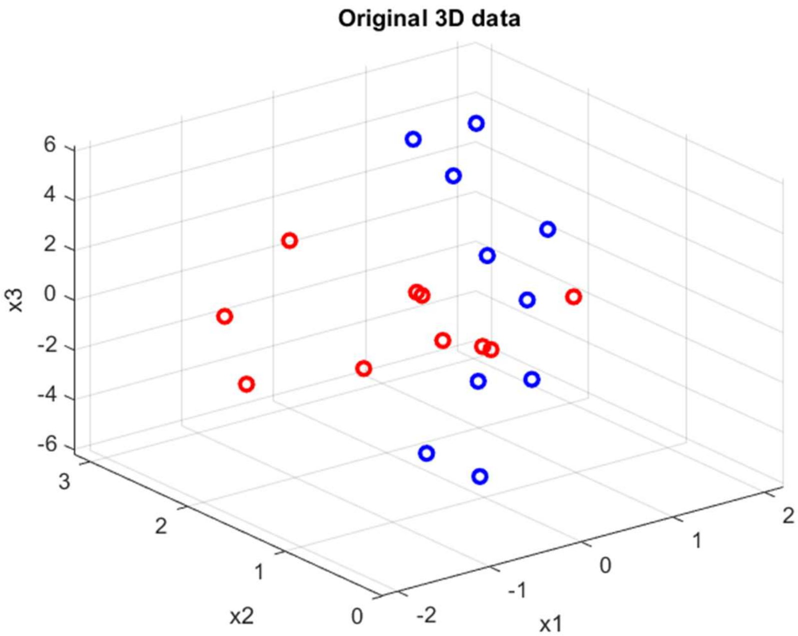
b/

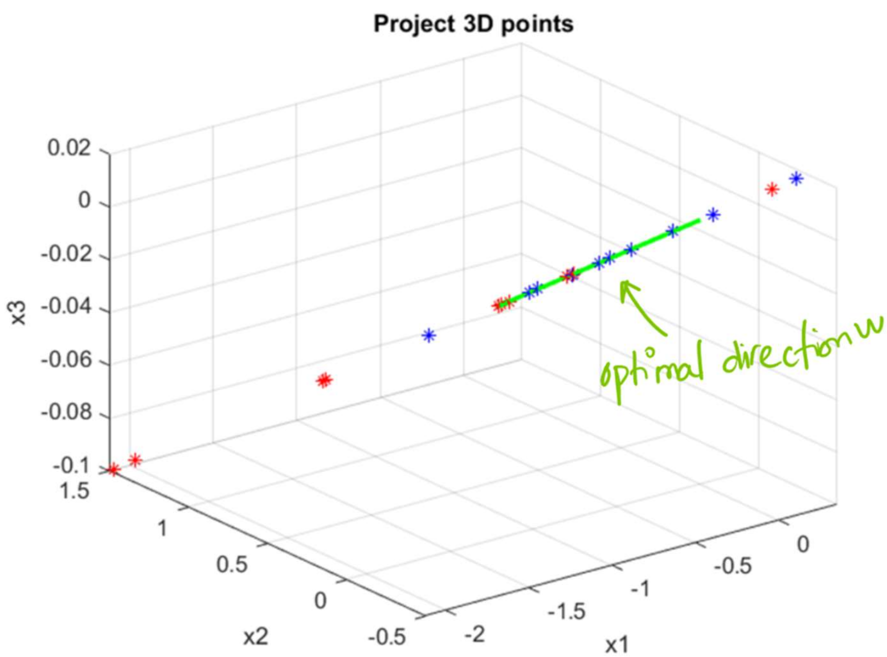
```
%% Data
data_1 = ...
    [0.28 1.31 -6.2;
    0.07 0.58 -0.78;
    1.54 2.01 -1.63;
    -0.44 1.18 -4.32;
    -0.81 0.21 5.73;
    1.52 3.16 2.77;
    2.20 2.42 -0.19;
    0.91 1.94 6.21;
    0.65 1.93 4.38;
    -0.26 0.82 -0.96];
data_2 = ...
    [0.011 1.03 -0.21
    1.27 1.28 0.08
    0.13 3.12 0.16
    -0.21 1.23 -0.11
    -2.18 1.39 -0.19
    0.34 1.96 -0.16
    -1.38 0.94 0.45
    -0.12 0.82 0.17
    -1.44 2.31 0.14
    0.26 1.94 0.08];
data_3 = ...
    [1.36 2.17 0.14
    1.41 1.45 -0.38
    1.22 0.99 0.69
    2.46 2.19 1.31
    0.68 0.79 0.87
    2.51 3.22 1.35
    0.60 2.44 0.92
    0.64 0.13 0.97
    0.85 0.58 0.99
    0.66 0.51 0.88];

%% a + b. Calculate the optimal direction w
w = fisher_linear_discriminant(data_1, data_2);
```

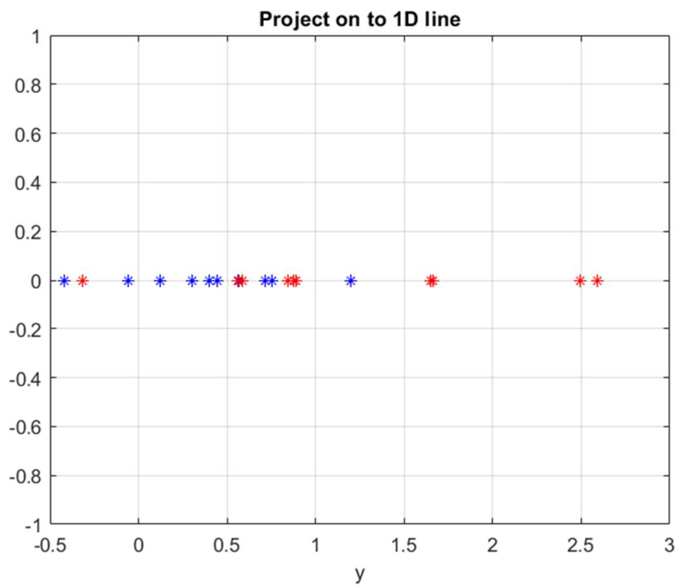
Result: optimal direction w = [-0.821 0.569 -0.0382]’

c/

```
%% c. Plot w & mark on it the projections of the projected points
figure;
plot3(data_1(:,1), data_1(:,2), data_1(:,3), 'bo', 'LineWidth', 2);
grid on; hold on;
plot3(data_2(:,1), data_2(:,2), data_2(:,3), 'ro', 'LineWidth', 2);
xlabel("x1"); ylabel("x2"); zlabel("x3");
title("Original 3D data");

figure;
quiver3(0, 0, 0, w(1), w(2), w(3), "g-", 'LineWidth', 2);
xlabel("x1"); ylabel("x2"); zlabel("x3");
grid on; hold on;

% Project data to w
y_1 = w' * data_1';
y_2 = w' * data_2';
% proj_1 = zeros(size(data_1));
% for i=1:length(data_1')
%     proj_1(i, :) = y_1(i) * w;
% end
% plot3(proj_1(:,1), proj_1(:,2), proj_1(:,3), 'y*');

proj_data_1 = project_data(data_1, w);
proj_data_2 = project_data(data_2, w);

plot3(proj_data_1(:,1), proj_data_1(:,2), proj_data_1(:,3), 'b*');
plot3(proj_data_2(:,1), proj_data_2(:,2), proj_data_2(:,3), 'r*');
title("Project 3D points");

% Project on to 1D line
figure;
plot(y_1, zeros(length(y_1), 1), 'b*');
grid on; hold on;
plot(y_2, zeros(length(y_2), 1), 'r*');
xlabel("y"); title("Project on to 1D line");
```

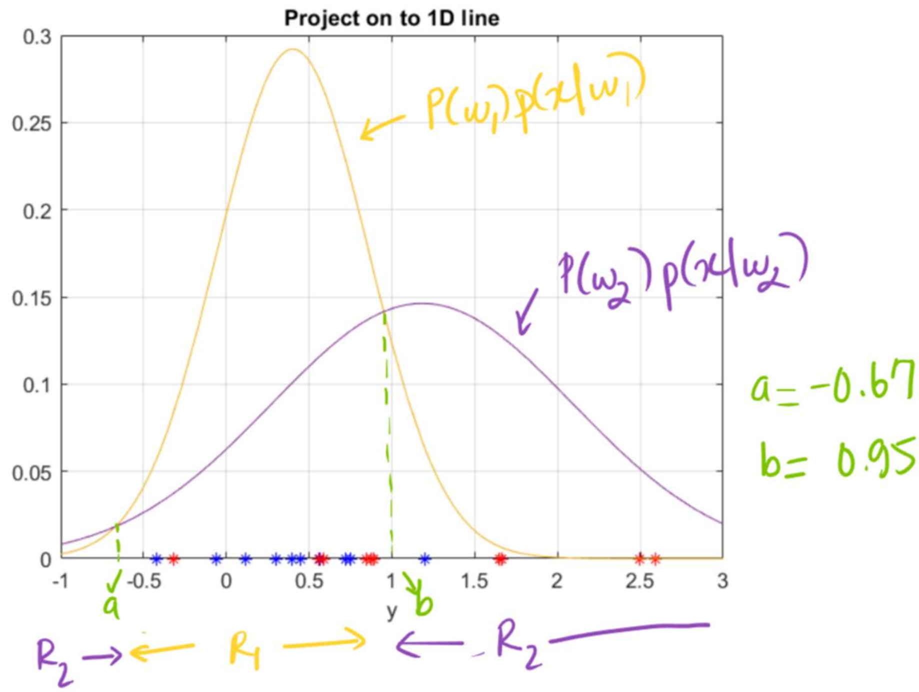

Original 3D data
x1
x2
x3


Project 3D points
x1
x2
x3
optimal direction w



d/

```
%% d. Find decision boundary
mu_1 = mean(y_1);
mu_2 = mean(y_2);
sigma_1 = std(y_1);
sigma_2 = std(y_2);
% Decision boundary
syms x u sigma;
gaussian = exp(-((x - u)^2) / (2 * sigma^2)) / (sigma * sqrt(2 * pi));
p_x_w1 = subs(gaussian, {u, sigma}, {mu_1, sigma_1});
p_x_w2 = subs(gaussian, {u, sigma}, {mu_2, sigma_2});
p_w1 = 1/3;
p_w2 = 1/3;
fplot(p_x_w1 * p_w1, [-1, 3]);
fplot(p_x_w2 * p_w2, [-1, 3]);
intersection = double(solve(p_x_w1 == p_x_w2, x));
a = min(intersection);
b = max(intersection);
```



e/

```
%% e. Training error
y = [y_1 y_2];
label = [ones(1,length(y_1)) 2*ones(1,length(y_2))];
decision = (y < a | y > b) + 1;
error_rate = sum(decision ~= label)/length(y);
disp(error_rate);
```

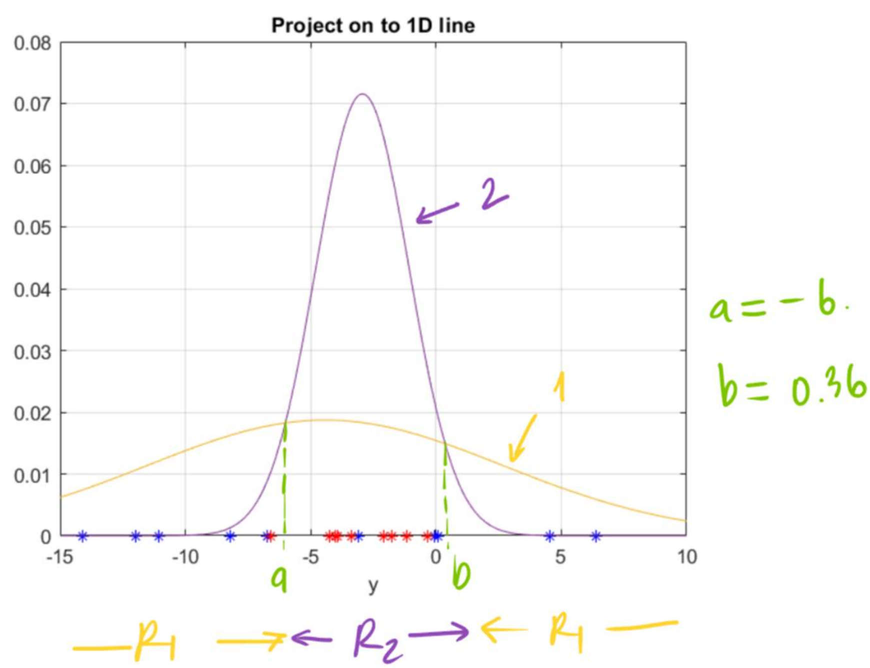
Result: error_rate = 0.3500

f/

```
%% f. Use non-optimal direction w = -[1.0 2.0 1.5]'
w = -[1.0 2.0 1.5]';
% Project data to w
y_1 = w' * data_1';
y_2 = w' * data_2';
% Project on to 1D line
figure;
plot(y_1, zeros(length(y_1), 1), 'b*');
grid on; hold on;
plot(y_2, zeros(length(y_2), 1), 'r*');
xlabel("y"); title("Project on to 1D line");
% Find decision boundary
mu_1 = mean(y_1);
mu_2 = mean(y_2);
sigma_1 = std(y_1);
sigma_2 = std(y_2);
syms x u sigma;
gaussian = exp(-((x - u)^2) / (2 * sigma^2)) / (sigma * sqrt(2 * pi));
```

```
p_x_w1 = subs(gaussian, {u, sigma}, {mu_1, sigma_1});
p_x_w2 = subs(gaussian, {u, sigma}, {mu_2, sigma_2});
p_w1 = 1/3;
p_w2 = 1/3;
fplot(p_x_w1 * p_w1, [-15, 10]);
fplot(p_x_w2 * p_w2, [-15, 10]);
intersection = double(solve(p_x_w1 == p_x_w2, x));
a = min(intersection);
b = max(intersection);

%Training error
y = [y_1 y_2];
label = [ones(1,length(y_1)) 2*ones(1,length(y_2))];
decision = (y < a | y > b) + 1;
error_rate = sum(decision ~= label)/length(y);
disp(error_rate);
```

By using new w (non optimal), we have:



Compared to the projected data given the optimal direction, we can see that the non-optimal w does not separate the data well.

Problem 5.

a/

```
u = [5; 5];
u1 = u + [-3; 7];
```

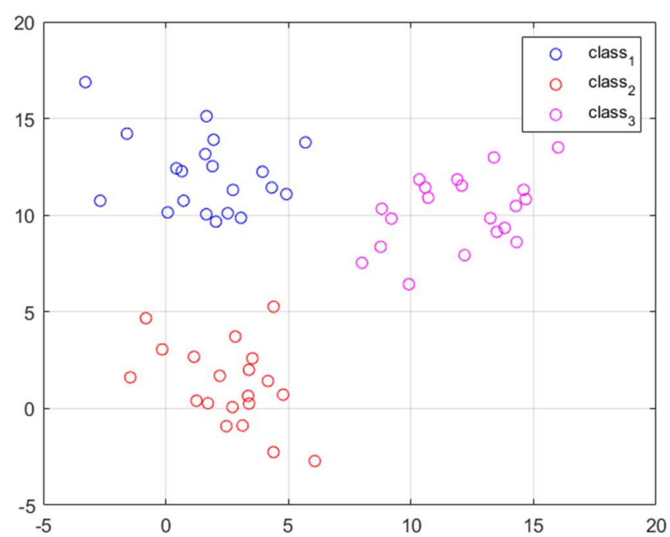
```
u2 = u + [-2.5; -3.5];
u3 = u + [7; 5];

S1 = [5 -1; -1 3];
S2 = [4 0; 0 4];
S3 = [3.5 1; 1 2.5];
%% a.
% Within-class scatter matrix
Sw = S1 + S2 + S3;
% Between-class scatter matrix
n1 = 1; n2 = 1; n3 = 1;
Sb = n1 * (u1 - u)*(u1 - u)' + n2 * (u2 - u)*(u2 - u)' + n3 * (u3 - u)*(u3 -
u)';
% Solve the generalized eigenvalue problem
[V, D] = eig(Sw^-1 * Sb);
% Find eigenvectors and eigenvalues
tmp = diag(D);
[~, idx] = sort(tmp(tmp~=0)); % all non-zero eigenvalues index
eigenvalues = diag(D(idx, idx));
eigenvectors = V(:, idx);
W = eigenvectors;
```

Eigenvalues = [4.239; 9.980]

Eigenvectors = [-0.896 -0.352; 0.444 -0.936]

b/

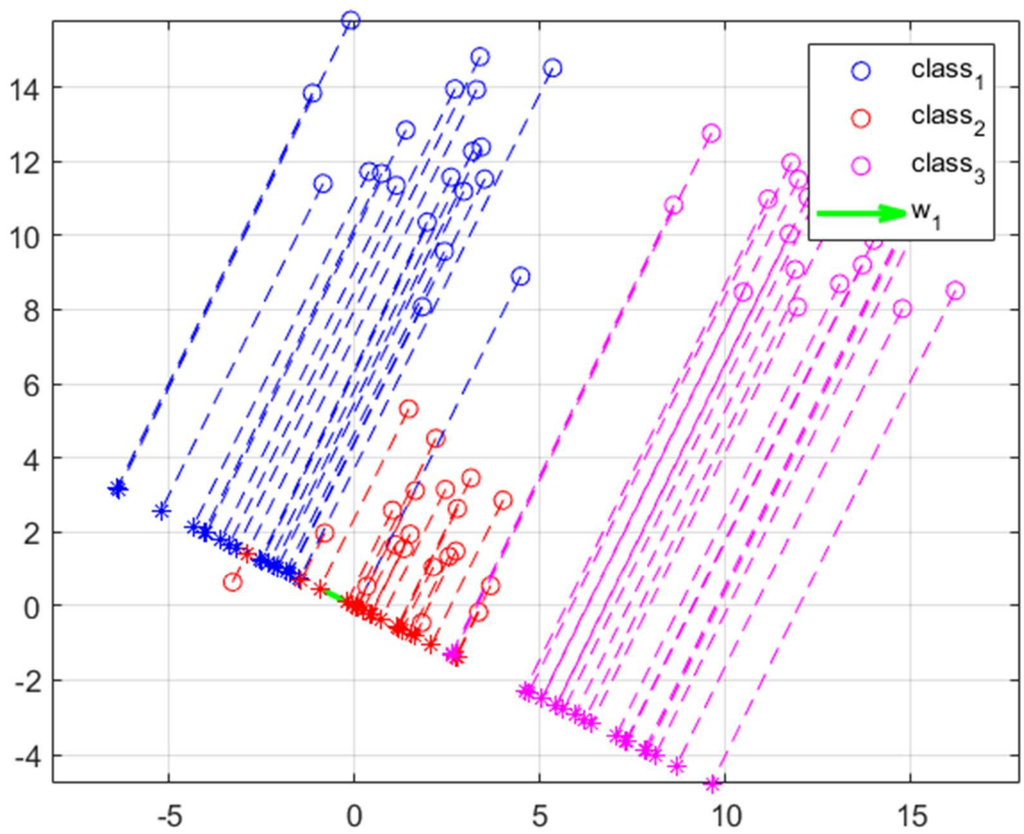
```
%% b. Generate 20 samples for each class
n = 20;
data_class_1 = mvnrnd(u1', S1, n);
data_class_2 = mvnrnd(u2', S2, n);
data_class_3 = mvnrnd(u3', S3, n);
```

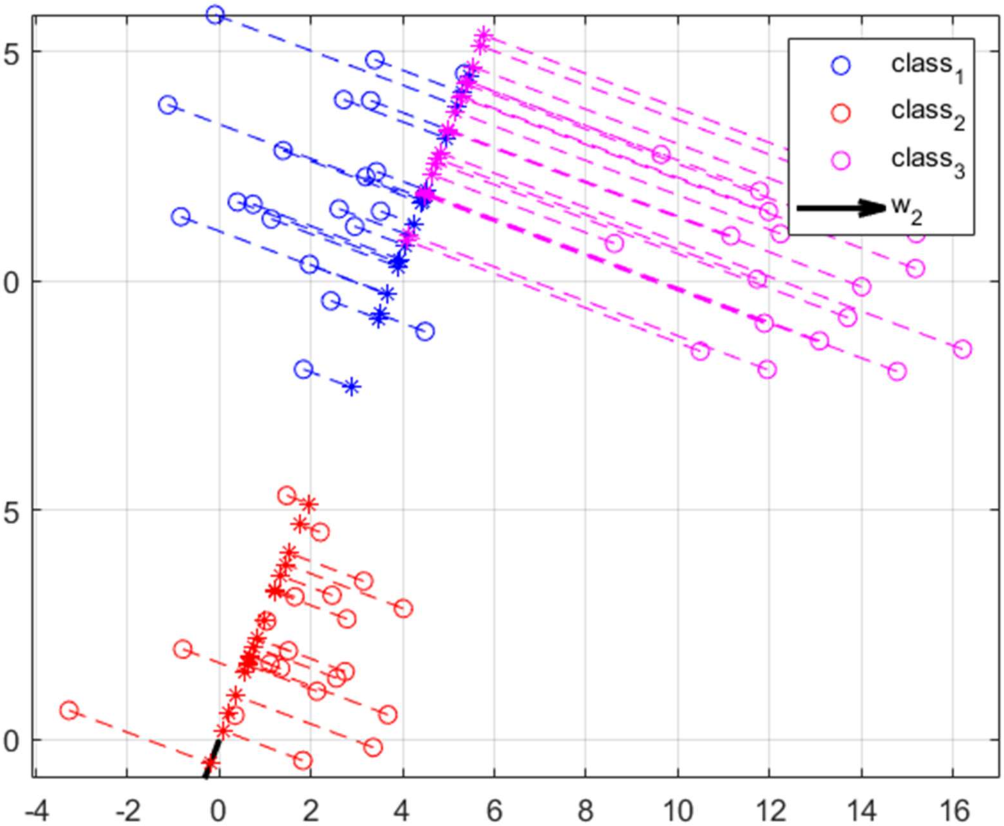


c/

```
%% c.
plot(data_class_1(:,1), data_class_1(:,2), 'bo');
grid on; hold on;
plot(data_class_2(:,1), data_class_2(:,2), 'ro');
plot(data_class_3(:,1), data_class_3(:,2), 'mo');

% Projection
quiver(0, 0, W(1,1), W(2,1), "g-", 'LineWidth', 2);

proj_data_1 = project_2d_data(data_class_1, W(:,1));
plot(proj_data_1(:,1), proj_data_1(:,2), 'b*');
proj_data_2 = project_2d_data(data_class_2, W(:,1));
plot(proj_data_2(:,1), proj_data_2(:,2), 'r*');
proj_data_3 = project_2d_data(data_class_3, W(:,1));
plot(proj_data_3(:,1), proj_data_3(:,2), 'm*');

for i=1:length(data_class_1)
    x = [data_class_1(i,1); proj_data_1(i,1)];
    y = [data_class_1(i,2); proj_data_1(i,2)];
    plot(x, y, 'b--');
end
for i=1:length(data_class_2)
    x = [data_class_2(i,1); proj_data_2(i,1)];
    y = [data_class_2(i,2); proj_data_2(i,2)];
    plot(x, y, 'r--');
end
for i=1:length(data_class_1)
    x = [data_class_3(i,1); proj_data_3(i,1)];
    y = [data_class_3(i,2); proj_data_3(i,2)];
    plot(x, y, 'm--');
end
axis equal;
legend("class_1", "class_2", "class_3", "w_1");

%%
figure;
plot(data_class_1(:,1), data_class_1(:,2), 'bo');
grid on; hold on;
plot(data_class_2(:,1), data_class_2(:,2), 'ro');
plot(data_class_3(:,1), data_class_3(:,2), 'mo');

quiver(0, 0, W(1,2), W(2,2), "k-", 'LineWidth', 2);

proj_data_1 = project_2d_data(data_class_1, W(:,2));
plot(proj_data_1(:,1), proj_data_1(:,2), 'b*');
proj_data_2 = project_2d_data(data_class_2, W(:,2));
plot(proj_data_2(:,1), proj_data_2(:,2), 'r*');
proj_data_3 = project_2d_data(data_class_3, W(:,2));
plot(proj_data_3(:,1), proj_data_3(:,2), 'm*');

for i=1:length(data_class_1)
    x = [data_class_1(i,1); proj_data_1(i,1)];
    y = [data_class_1(i,2); proj_data_1(i,2)];
    plot(x, y, 'b--');
end
```

```
for i=1:length(data_class_2)
    x = [data_class_2(i,1); proj_data_2(i,1)];
    y = [data_class_2(i,2); proj_data_2(i,2)];
    plot(x, y, 'r--');
end
for i=1:length(data_class_1)
    x = [data_class_3(i,1); proj_data_3(i,1)];
    y = [data_class_3(i,2); proj_data_3(i,2)];
    plot(x, y, 'm--');
end

axis equal;
legend("class_1", "class_2", "class_3", "w_2");

function projected_data = project_2d_data(data, direction)
    w = direction;
    projected_data = zeros(size(data));
    for i=1:length(data)
        projected_data(i, :) = dot(data(i, :), w') / norm(w')^2 * w';
    end
end
```

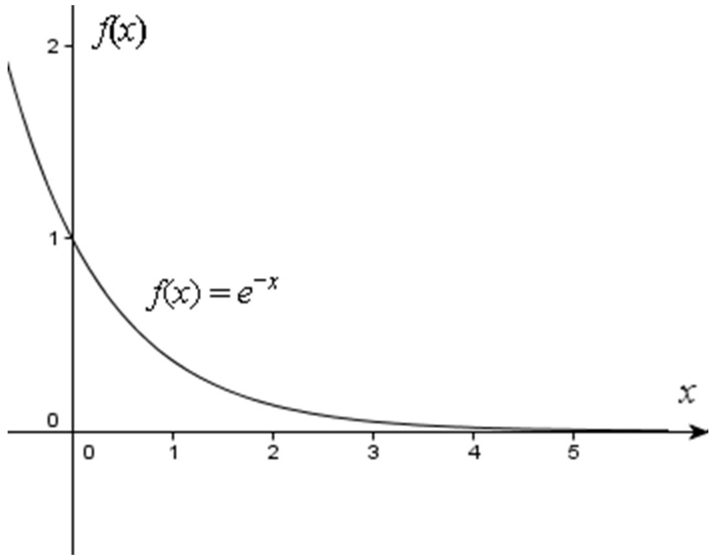




It is clear that the first project line is better for better separation of the projections for the generated samples.

Problem 6.

a/



$$\phi(x) = \begin{cases} e^{-x} & ; \ x > 0 \\ 0 & , \ x \le 0 \end{cases}$$

In this problem, the volume $V_n = h_n$.

$$E[p_n(x)] = \frac{1}{n}\sum_{i=1}^{n} E\left[\frac{1}{V_n}\phi\left(\frac{x - x_i}{h_n}\right)\right] = \frac{1}{n}\sum_{i=1}^{n} E\left[\frac{1}{h_n}e^{-\left(\frac{x-x_i}{h_n}\right)}\right]$$

* For $x < 0$, $p_n(x) = 0 \rightarrow E[p_n(x)] = 0$

* For $0 \le x \le a$,

$$E[p_n(x)] = \int_0^x \frac{1}{a}\cdot\frac{1}{h_n}e^{-\left(\frac{x-s}{h_n}\right)}ds = \frac{1}{a}\cdot\left[e^0 - e^{-x/h_n}\right] = \frac{1}{a}\left(1 - e^{-x/h_n}\right)$$

* For $x \ge a$,

$$E[p_n(x)] = \int_0^a \frac{1}{a}\frac{1}{h_n}e^{-\left(\frac{x-s}{h_n}\right)}ds = \frac{1}{a}\left[e^{-\frac{(x-a)}{h_n}} - e^{-\frac{x}{h_n}}\right]$$

$$= \frac{1}{a}\cdot e^{-x/h_n}\cdot\left(e^{a/h_n} - 1\right)$$

b/

```
x = linspace(-5, 5, 1000);
p_mean = pn_x(x, 1, 1);
plot(x, p_mean);
grid on;
xlabel("x"); ylabel("E[p_n(x)]");
```

```
hold on;
p_mean = pn_x(x, 1, 1/4);
plot(x, p_mean);

hold on;
p_mean = pn_x(x, 1, 1/16);
plot(x, p_mean);
legend("h_n = 1", "h_n = 1/4", "h_n = 1/16");

function f = pn_x(x, a, hn)
    f = zeros(size(x));
    for i=1:length(x)
        if (x(i) < 0)
            f(i) = 0;
        elseif ((x(i) >= 0) && (x(i) <= a))
            disp(f(i));
            f(i) = (1 - exp(-x(i) / hn)) / a;
        else
            f(i) = exp(-x(i) / hn) * (exp(a / hn) - 1) / a;
        end
    end
end
```