Statistical Pattern Recognition: Assignment 1
Praveen Kumar N (201082001)

January 18, 2021

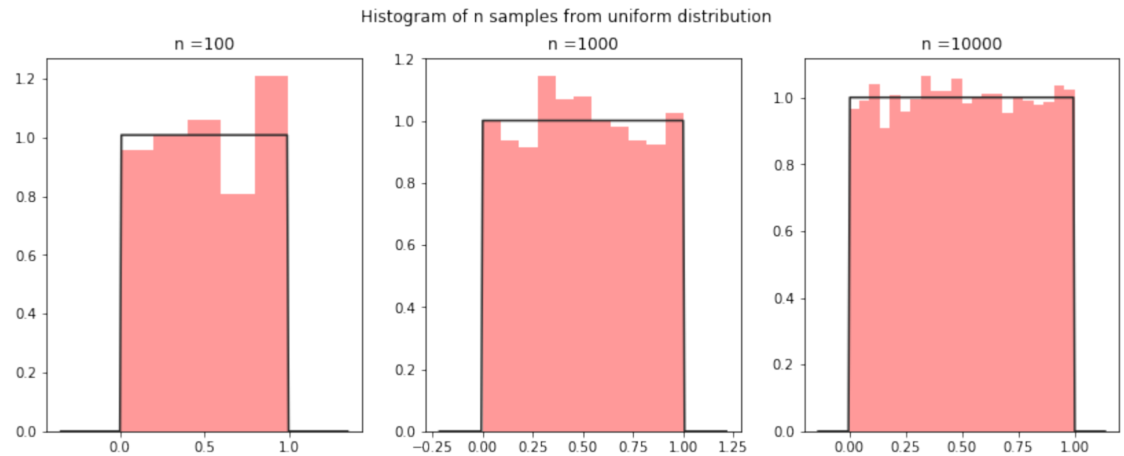
1 Problem 1

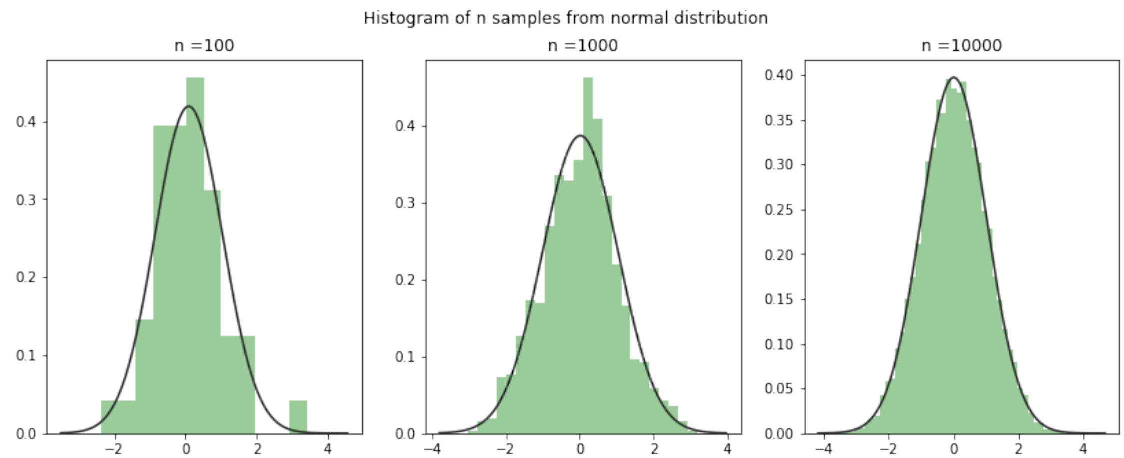
Sample n=100, 1000, 10000 points from (a) Uniform Distribution in 0 to 1. (b) Gaussian Distribution with mean 0 and variance 1. (c) Exponential Distribution with rate parameter = 1. Verify if the points are generated according to the respective distribution by plotting a histogram of the fraction of points in each case. Label graph properly.

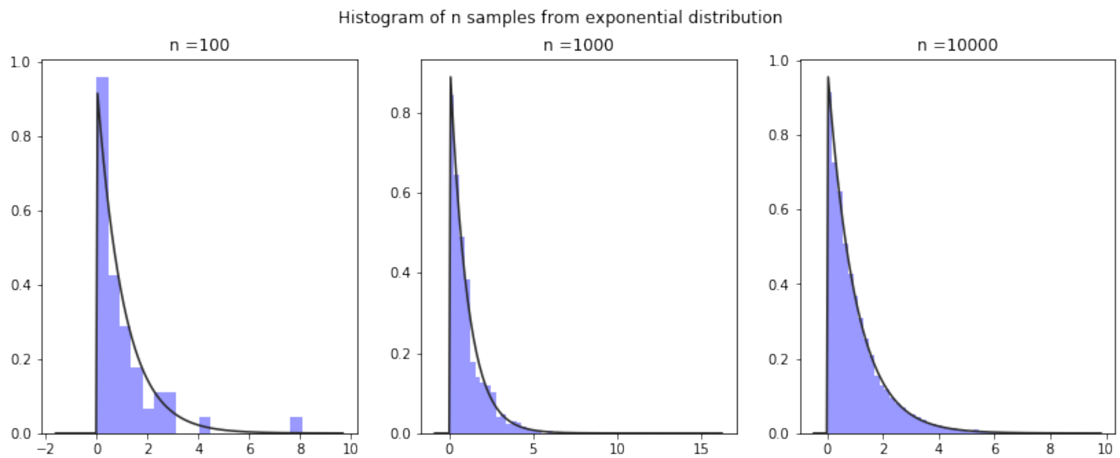
[89]:
```
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
```

[220]:
```
num_samp = [100,1000,10000]            # number of samples

# samples from uniform distribution in 0 to 1

fig,axes=plt.subplots(1,len(num_samp),figsize=(14,5))
fig.suptitle("Histogram of n samples from uniform distribution")

for i in range(len(num_samp)):
    n = num_samp[i]
    uni_samp = np.random.rand(n)       # n samples from unifrom distribution
    sns.distplot(uni_samp, ax=axes[i], color='r', fit=stats.uniform, kde=False)
    axes[i].set_title("n ={}".format(n))
plt.show()

# samples from Gaussian distribution with mean 0 and variance 1

fig,axes=plt.subplots(1,len(num_samp),figsize=(14,5))
fig.suptitle("Histogram of n samples from normal distribution")

for i in range(len(num_samp)):
    n = num_samp[i]
    gauss_samp = np.random.randn(n)       # n samples from normal distribution
    sns.distplot(gauss_samp, ax=axes[i], color='g', fit=stats.norm, kde=False)
    axes[i].set_title("n ={}".format(n))
```

```
plt.show()

# samples from Exponential distribution with parameter 1

fig,axes=plt.subplots(1,len(num_samp),figsize=(14,5))
fig.suptitle("Histogram of n samples from exponential distribution")

for i in range(len(num_samp)):
    n = num_samp[i]
    exp_samp = np.random.exponential(1,n)  # n samples from exponential␣
 ↪distribution
    sns.distplot(exp_samp, ax=axes[i], color='b', fit=stats.expon, kde=False)
    axes[i].set_title("n ={}".format(n))
plt.show()
```







2 Problem 2

Assume a unit circle centered at (0,0). Let n = 2; 3; 5 points be uniformly sampled from the circumference of the circle. Write a python program to estimate the probability of n = 2; 3; 5 points lie within some semi-circle.

[209]:
```
n_iter     = 50000   # Number of iterations to estimate the probability
n_samp     = [2,3,5] # number of points to be sampled from the circumference
est_probs  = []      # list of estimated probabilities
tru_probs  = []      # list of true probabilities

for n in n_samp:

    tru_probs.append((n/2**(n-1)))           # computing true probability, p =
 ↪n*(1/2)^(n-1)
    n_suc = 0                                # number of successes indicating
 ↪number of times
                                             # we get all n points lying on same
 ↪semicircle
    for _ in range(n_iter):

        samp_theta  = np.random.uniform(0,2*np.pi,n) # sampling n
 ↪points(represented by its angle)
                                                     # from the circumference
 ↪uniformly


        # Checking whether these n points lie on same semicircle
```

```
        samp_theta_min = np.min(samp_theta,initial=2*np.pi,where=samp_theta>np.
↪pi)
        samp_theta  = ((2*np.pi-samp_theta_min)+samp_theta)%(2*np.pi)
        samp_theta  = np.where(samp_theta>np.pi,samp_theta-2*np.pi,samp_theta)
        samp_theta.sort()
        theta_diffs = np.ediff1d(samp_theta)
        # for these n points to be in same semicircle sum of the angles
        # between consecutive points should not cross pi or 180 degrees
        if np.sum(theta_diffs)<=np.pi:
            n_suc = n_suc+1

    est_probs.append(n_suc/n_iter)            # estimated probability = no. of␣
↪successes/ no. of iterations

print("True probabilities:",tru_probs)
print("Estimated probabilities:",est_probs)
```

```
True probabilities: [1.0, 0.75, 0.3125]
Estimated probabilities: [1.0, 0.7488, 0.3123]
```

3 Problem 3

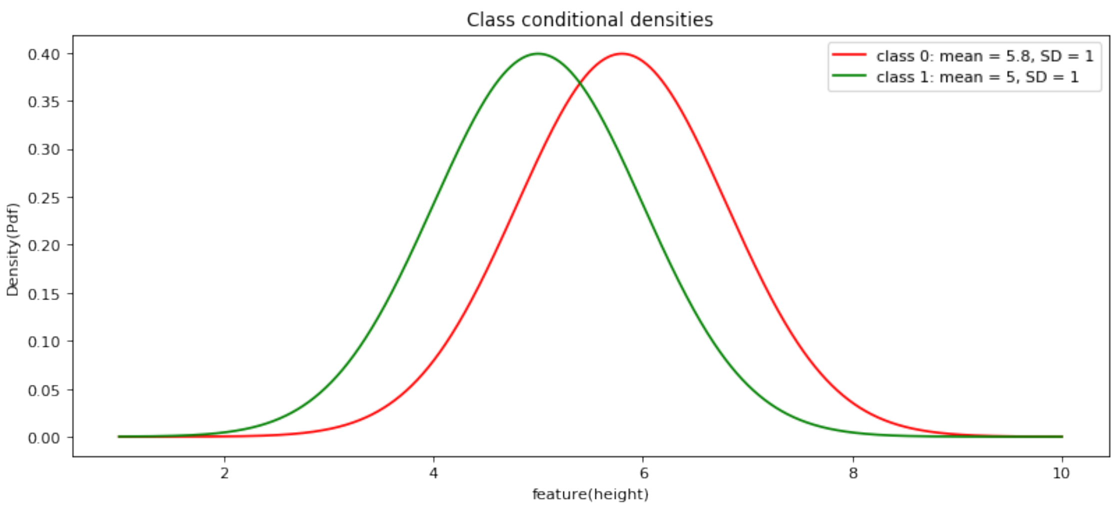
Assume two classes male and female. The height of the male class is distributed according to the normal distribution with a mean of 5.8 feet and a standard deviation of 1 feet and the height of the female class is distributed with a mean of 5 feet and a standard deviation of 1 feet. Assume following prior probabilities for two classes (a) For Male 0.5 and for Female 0.5. (b) For Male 0.1 and for Female 0.9. For each of the above cases, specify the priors, plot the class conditional densities and posterior probabilities of both the classes.Compute the misclassification error. Draw the decision boundary for the Bayes classifier.

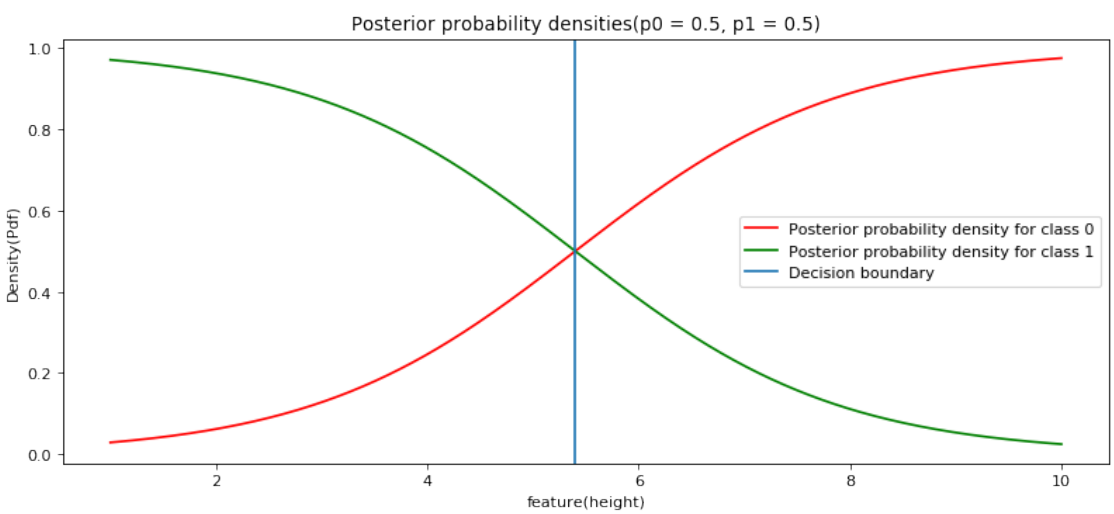
[219]:
```
# feature(x)                  : height
# classes(w)                  : male(class 0) , female(class 1)
# prior probablities          : p0 = Pr{class = 0} , p1 = Pr{class = 1}
# class conditional densities : f0(x) = fx(x/class = 0) ,
#                               f1(x) = fx(x/class = 1)
# posterior probabilities     : Pr{class = 0/x} = q0(x) = fx(x/class =␣
↪0)*Pr{class = 0}/fx(x) = p0*f0(x)/fx(x) ,
#                               Pr{class = 1/x} = q1(x) = fx(x/class =␣
↪1)*Pr{class = 1}/fx(x) = p1*f1(x)/fx(x)
# fx(x) = p0*f0(x) + p1*f1(x)
# classification              : x>= decision boundary => class = 1
#                               x<  decision boundary => class = 0
```

```
mean_m = 5.8  # mean height of male class
SD_m   = 1    # standard deviation of male class
mean_f = 5    # mean height of female class
SD_f   = 1    # standard deviation of female class

# plot of class conditional densities

x    = np.linspace(1,10,1000)          # taking 1000 features in the range␣
 ↪[1,10]
f0_x = stats.norm.pdf(x,mean_m,SD_m)   # class conditional density for class = 0
f1_x = stats.norm.pdf(x,mean_f,SD_f)   # class conditional density for class = 1
plt.figure(figsize=(12, 5), dpi=80)
plt.plot(x,f0_x,'r')
plt.plot(x,f1_x,'g')
plt.legend(["class 0: mean = {}, SD = {}".format(mean_m,SD_m),"class 1: mean =␣
 ↪{}, SD = {}".format(mean_f,SD_f)])
plt.title("Class conditional densities")
plt.xlabel("feature(height)"); plt.ylabel("Density(Pdf)")
plt.show()

#------------------------------ CASE A ----------------------------------------

# Prior probabilities
p0 = m_prior = 0.5 # prior probability of male class
p1 = f_prior = 0.5 # prior probability of female class

# computing posterior probabilities

q0_x = []
q1_x = []
class_0 = []
class_1 = []

fx_x = [p0*f0_x[i]+p1*f1_x[i] for i in range(1000)]


for i in range(1000):
    q0_x.append(f0_x[i]*p0/fx_x[i])
    q1_x.append(f1_x[i]*p1/fx_x[i])
    # splitting features based on classes they belong to
    if q0_x[i]>=q1_x[i]:
        class_0.append(x[i])
    else:
        class_1.append(x[i])
```

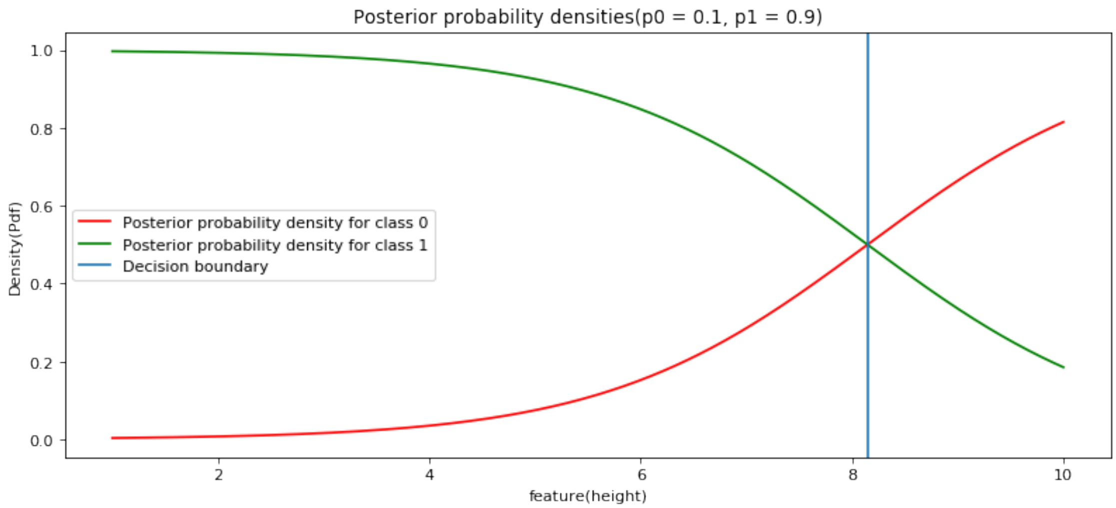
```
# Computing decision boundary

dec_boundry = (mean_m+mean_f)/2
# decision boundary for normal class conditional densities
# with different mean and same variance and p0=p1=1/2

# plot of posterior probabilities

plt.figure(figsize=(12, 5), dpi=80)
plt.plot(x,q0_x,'r')
plt.plot(x,q1_x,'g')
plt.axvline(dec_boundry)
plt.legend(["Posterior probability density for class 0","Posterior probability␣
 ↪density for class 1","Decision boundary"])
plt.title("Posterior probability densities(p0 = {}, p1 = {})".format(p0,p1))
plt.xlabel("feature(height)"); plt.ylabel("Density(Pdf)")
plt.show()


# Computing misclassification error

mis_error   = p1*stats.norm.sf(dec_boundry,loc=mean_f,scale=SD_f) + p0*stats.
 ↪norm.cdf(dec_boundry,loc=mean_m,scale=SD_m)
print("CASE A : p0 = {}, p1 = {}".format(p0,p1))
print("Decision boundary : {}".format(dec_boundry))
print("Missclassification error : {}".format(mis_error))

#------------------------------ CASE B ----------------------------------------

# Prior probabilities
p0 = m_prior = 0.1 # prior probability of male class
p1 = f_prior = 0.9 # prior probability of female class

# Computing posterior probabilities

q0_x = []
q1_x = []
class_0 = []
class_1 = []

fx_x = [p0*f0_x[i]+p1*f1_x[i] for i in range(1000)]


for i in range(1000):
    q0_x.append(f0_x[i]*p0/fx_x[i])
    q1_x.append(f1_x[i]*p1/fx_x[i])
    # splitting features based on classes they belong to
```

```
    if q0_x[i]>=q1_x[i]:
        class_0.append(x[i])
    else:
        class_1.append(x[i])


# Computing decision boundary

dec_boundry = (mean_m+mean_f)/2 - ((SD_m)**2 * np.log(p1/p0))/(mean_f-mean_m)
# decision boundary for normal class conditional densities
# with different mean and same variance and p0=0.1,p1=0.9

# plot of posterior probabilities

plt.figure(figsize=(12, 5), dpi=80)
plt.plot(x,q0_x,'r')
plt.plot(x,q1_x,'g')
plt.axvline(dec_boundry)
plt.legend(["Posterior probability density for class 0","Posterior probability␣
 ↪density for class 1","Decision boundary"])
plt.title("Posterior probability densities(p0 = {}, p1 = {})".format(p0,p1))
plt.xlabel("feature(height)"); plt.ylabel("Density(Pdf)")
plt.show()

# Computing misclassification error

mis_error   = p1*stats.norm.sf(dec_boundry,loc=mean_f,scale=SD_f) + p0*stats.
 ↪norm.cdf(dec_boundry,loc=mean_m,scale=SD_m)
print("CASE B : p0 = {}, p1 = {}".format(p0,p1))
print("Decision boundary : {}".format(dec_boundry))
print("Missclassification error : {}".format(mis_error))
```






```
CASE A : p0 = 0.5, p1 = 0.5
Decision boundary : 5.4
Missclassification error : 0.3445782583896759
```




```
CASE B : p0 = 0.1, p1 = 0.9
Decision boundary : 8.146530721670276
Missclassification error : 0.0997960343653087
```

DATE

STATISTICAL PATTERN RECOGNITION

ASSIGNMENT - 1 (due: 19/01/2021)

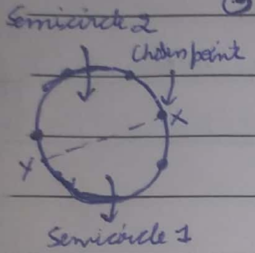
Praveen Kumar N
20108200l

② Problem: To find probability that n points sampled uniformly from the circumference of a circle lie on some semi circle

Solution: ① for $n=1$ & 2 we can easily see that we always get them on some semicircle since angle b/w any two points on circle $\leq 180°$

$\Rightarrow P=1$ for $n=1$ & 2

③ Now for $n>2$.



↳ randomly pick 1 out of n points (say X) here there are ${}^nC_1$ ways of choosing 1 point out of n points

↳ Now construct a semicircle XY starting from point X (i.e length $(XY) = \frac{1}{2}$ circumference) $\Rightarrow$ we have two semicircles XY & YX

↳ Now remaining $(n-1)$ can be in either in semicircle XY or YX with probability $(\frac{1}{2})$

↳ Now for all $(n-1)$ points to lie in one of the semicircles. The probability = probability that 1^{st} point lie in a semicircle $\times$ probability that 2^{nd} point lie in a semicircle $\times$

⋮

probability that $(n-1)^{th}$ point lie in a semicircle

$$= (\tfrac{1}{2}) \times (\tfrac{1}{2}) \cdots (\tfrac{1}{2})$$
$$= (\tfrac{1}{2})^{n-1}$$

↳ Now since while choosing 1 point out of n points we had ${}^nC_1 = n$ ways we get above probability for each of the ways

$\Rightarrow$ total probability $= n\,(\frac{1}{2})^{n-1}$ // $= p$

for $n=2$	$p=1$
$n=3$	$p=0.75$
$n=5$	$p=0.325$ //

DATE

(3) Given feature → height (x)

class → male, female
$\quad\quad\quad\quad (0) \quad (1)$

$Pr\{x/class=0\} = f_0(x) \rightarrow N(5.8, 1)$
$Pr\{x/class=1\} = f_1(x) \rightarrow N(5, 1)$ } class conditional densities

$\mu_0 = 5.8, \; \mu_1 = 5$
$\sigma_0 = \sigma_1 = \sigma = 1$

(i) case a:

↳ $p_0 = Pr\{class=0\} = Pr\{male\} = 0.5$
$p_1 = Pr\{class=1\} = Pr\{female\} = 0.5$ } prior probabilities

↳ $q_0(x) = Pr\{class=0/x\} = p_0 f_0(x) / f_X(x)$
$q_1(x) = Pr\{class=1/x\} = p_1 f_1(x) / f_X(x)$ } posterior probability densities

$f_X(x) = P_0 f_0(x) + P_1 f_1(x)$

↳ Bayes classifier $h_B(x) = 0$ if $q_0(x) \geq q_1(x)$
$\quad\quad\quad\quad\quad 1 \quad q_0(x) < q_1(x)$

↳ decision boundary

we have, if $q_0(x) \geq q_1(x)$ → class 0
$q_0(x) < q_1(x)$ → class 1

~~decision boundary is given by x where~~ ~~$q_0(x) = q_1(x)$~~

~~ie~~

Consider $q_0(x) \geq q_1(x)$

$\Rightarrow \dfrac{\cancel{p_0} f_0(x)}{\cancel{f_X(x)}} \geq \dfrac{\cancel{p_1} f_1(x)}{\cancel{f_X(x)}}$ here $p_0 = p_1 = 0.5$

$\Rightarrow f_0(x) \geq f_1(x)$

$\Rightarrow \dfrac{1}{\cancel{\sqrt{2\pi}\sigma}} \exp\left[-\dfrac{(x-\mu_0)^2}{2\sigma^2}\right] \geq \dfrac{1}{\cancel{\sqrt{2\pi}\sigma}} \exp\left[-\dfrac{(x-\mu_1)^2}{2\sigma^2}\right]$

↓ take '$\ln$' on both sides

$\Rightarrow -\dfrac{(x-\mu_0)^2}{\cancel{2\sigma^2}} \geq -\dfrac{(x-\mu_1)^2}{\cancel{2\sigma^2}}$

DATE

$$\Rightarrow (x-\mu_0)^2 \le (x-\mu_1)^2$$

$$\cancel{x^2} - 2\mu_0 x + \mu_0^2 \le \cancel{x^2} - 2\mu_1 x + \mu_1^2$$

$$+2x(\mu_1 - \mu_0) \le (\mu_1^2 - \mu_0^2)$$

$$x \le \frac{(\mu_1^2 - \mu_0^2)}{2(\mu_1 - \mu_0)}$$

$$\boxed{x \le \frac{\mu_1 + \mu_2}{2}} \Rightarrow \text{class } 0$$

$$\boxed{x > \frac{\mu_1 + \mu_0}{2}} \rightarrow \text{class } 1$$

$\Rightarrow$ decision boundary $= \frac{\mu_1 + \mu_0}{2} = \frac{5.8 + 5}{2} = 5.4 /\!/ = D$

↳ <u>misclassification error:</u>

$$\text{error} = p_1 \int_{\text{decision boundary} = D}^{\infty} f_1(x)\,dx + p_0 \int_{-\infty}^{\text{decision boundary} = D} f_0(x)\,dx$$

$$\text{error} = p_1 [1 - \underbrace{F_1(D)}_{\text{CDF}}] + p_0 \underbrace{F_0[D]}_{\text{CDF}} = 0.5 [\underbrace{1 - F_1(5.4)}_{Pr\{x > 5.4 / \text{class} = 1\}}] + 0.5 \underbrace{F_0(0.5)}_{Pr\{x < 5.4 / \text{class} = 0\}} /\!/$$

⊙ <u>case b:</u>

↳ $p_0 = 0.1$, $p_1 = 0.9$ } prior probabilities

↳ $q_0(x) = p_0 f_0(x) / f_X(x)$, $q_1(x) = p_1 f_1(x) / f_X(x)$ } posterior probability densities

$$f_X(x) = p_0 f_0(x) + p_1 f_1(x)$$

↳ <u>Bayes classifier:</u> $h_B(x) = 0$ if $q_0(x) \ge q_1(x)$

$\qquad\qquad 1 \quad q_0(x) < q_1(x)$

DATE

↳ decision boundary

we have, if $q_0(x) \geq q_1(x) \rightarrow$ class 0

$q_0(x) < q_1(x) \rightarrow$ class 1

consider $q_1(x) > q_0(x)$

$$\Rightarrow \frac{p_1 f_1(x)}{\cancel{f_X(x)}} > \frac{p_0 f_0(x)}{\cancel{f_X(x)}}$$

$$\Rightarrow p_1 \frac{1}{\cancel{\sqrt{2\pi}\sigma}} \exp\left[-\frac{(x-\mu_1)^2}{2\sigma^2}\right] > p_0 \frac{1}{\cancel{\sqrt{2\pi}\sigma}} \exp\left[-\frac{(x-\mu_0)^2}{2\sigma^2}\right]$$

here $p_0 \neq p_1$

$$\Rightarrow \exp\left[-\frac{(x-\mu_1)^2}{2\sigma^2}\right] > \frac{p_0}{p_1} \exp\left[-\frac{(x-\mu_0)^2}{2\sigma^2}\right]$$

↓ take 'ln' on both sides

$$\Rightarrow -\frac{(x-\mu_1)^2}{2\sigma^2} > \ln\left(\frac{p_0}{p_1}\right) - \frac{(x-\mu_0)^2}{2\sigma^2}$$

$$\Rightarrow -\frac{(x-\mu_1)^2}{2\sigma^2} > -\ln\left(\frac{p_1}{p_0}\right) - \frac{(x-\mu_0)^2}{2\sigma^2}$$

$$\Rightarrow \frac{(x-\mu_1)^2}{2\sigma^2} < \ln\left(\frac{p_1}{p_0}\right) + \frac{(x-\mu_0)^2}{2\sigma^2}$$

$$\Rightarrow (x-\mu_1)^2 < 2\sigma^2 \ln\left(\frac{p_1}{p_0}\right) + (x-\mu_0)^2$$

$$\Rightarrow \cancel{x^2} - 2x\mu_1 + \mu_1^2 < 2\sigma^2 \ln\left(\frac{p_1}{p_0}\right) + \cancel{x^2} - 2x\mu_0 + \mu_0^2$$

$$\Rightarrow 2\sigma^2 \ln\left(\frac{p_1}{p_0}\right) - 2x\mu_0 + \mu_0^2 > -2x\mu_1 + \mu_1^2$$

$$\Rightarrow 2\sigma^2 \ln\left(\frac{p_1}{p_0}\right) + 2x(\mu_1 - \mu_0) > (\mu_1^2 - \mu_0^2)$$

$$\Rightarrow \sigma^2 \ln\left(\frac{p_1}{p_0}\right) + x(\mu_1 - \mu_0) > \frac{(\mu_1^2 - \mu_0^2)}{2}$$

$$\Rightarrow x > \frac{(\mu_1^2-\mu_0^2)}{2(\mu_1-\mu_0)} - \frac{\sigma^2}{(\mu_1-\mu_0)} \ln\left(\frac{p_1}{p_0}\right)$$

$$\Rightarrow \boxed{x > \frac{(\mu_1+\mu_0)}{2} - \frac{\sigma^2}{(\mu_1-\mu_0)} \ln\left(\frac{p_1}{p_0}\right)} \Rightarrow \text{class } 1$$

$$\Rightarrow \boxed{x \leq \frac{(\mu_1+\mu_0)}{2} - \frac{\sigma^2}{(\mu_1-\mu_0)} \ln\left(\frac{p_1}{p_0}\right)} \Rightarrow \text{class } \theta$$

$$\Rightarrow \text{decision boundary} = \frac{\mu_1+\mu_0}{2} - \frac{\sigma^2}{(\mu_1-\mu_0)} \ln\left(\frac{p_1}{p_0}\right) = D$$

here $\mu_1 = 5.$ $\quad p_1 = 0.9 \quad \sigma = 1$

$\mu_0 = 5.8 \quad p_0 = 0.1$

$$\Rightarrow D = \frac{5+5.8}{2} - \frac{1}{(5-5.8)} \ln\left(\frac{0.9}{0.1}\right) = 8.1465$$

↳ <u>misclassification error:</u>

$$\text{error} = p_1 \int_D^\infty f_\theta(x)\,dx + p_0 \int_{-\infty}^D f_0(x)\,dx$$

$$\text{error} = 0.9\,[1 - F_1(8.1465)] + 0.1\,[F_\theta(8.1465)]$$

$Pr\{x > 8.1465 / \text{class}=1\}$ $\quad$ $Pr\{X < 8.1465 / \text{class}=\theta\}$