561 HW 8

1a) - $I(0)=1$: if $P(X=x)=1$ we are certain of the outcome and there is no more info to learn.

- $I(\cdot)$ is monotonic decreasing: We begin with a finite amount of information that can be learned so as we learn/train the amount of remaining information must less than or equal to its past value, implying monotonic decreasing.
- $I(\cdot)\geq 0$: The concept of negative information does not make sense so we need the range of $I(\cdot)$ to be greater than or equal to 0.
- $I(P(X_1;X_2))=$
$I(P(X_1)+P(X_2))$: for indepen RVs $P(X_1,X_2)=P(X_1)P(X_2)$
$\log(ab)=\log(a)+\log(b)$ so $\log(P(X_1,X_2))=\log(P(X_1))+\log(P(X_2))$

1b) $I(1)=0$: $\log_2\left(\frac{1}{P(X=x)}\right)\rightarrow P(X=x)=1\rightarrow\log_2(1)=0$ ✓

$I(\cdot)$ monotonic decrease: as we approach $P(X=x)=1^+$ the value of $\log_2(1/P(X=x))$ approaches 0
That is

$$\log_2\left(\frac{1}{P(X=x)=0.1}\right)>\log_2\left(\frac{1}{P(X=x)=0.5}\right)>\log_2\left(\frac{1}{P(X=x)=1}\right)$$

which is equivalent to $I(0.1)>I(0.5)>I(1)$
so as more info is learned

$I(\cdot)\geq 0$: since $P(X=x)$ is on range $[0\ 1]$
$\log_2\left(\frac{1}{P(X=x)}\right)$ is on range $[0\ \infty)$ so always non negative

1b) For independent X_1, X_2

$$\log_2\left(\frac{1}{P(X_1=x_1, X_2=x_2)}\right) = \log_2\left(\frac{1}{P(X_1=x_1)P(X_2=x_2)}\right)$$

$$= \log_2\left(\frac{1}{P(X_1=x_1)}\right) + \log_2\left(\frac{1}{P(X_2=x_2)}\right) \rightarrow I(P(X_1)) + I(P(X_2))$$

2) $H(\hat{y}) + H(X|\hat{y}) = H(X) + H(\hat{y}|X)$

when y is a function completly dependent on X such that $y = g(x)$ we know $H(y|x) = 0$ as X explains all the entropy of y.

Here $\hat{y} = f(x)$ so $H(\hat{y}|X) = 0 \rightarrow H(\hat{y}) = H(X) - H(X|\hat{y})$

so if $H(X|\hat{y}) = 0$: $H(\hat{y}) = H(X)$

if $H(X|\hat{y}) \neq 0$: $H(\hat{y}) < H(X)$

3a) See attatched code

3b) $X_1 | X_2 = 0$ $\quad \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{bmatrix}$ $\quad \rho = \frac{3}{\sqrt{7}\sqrt{2}} = 0.802$

$$P(X_1|X_2=0) \sim N\left(\mu_1 + \frac{\sigma_1}{\sigma_2}\rho(0-\mu_2), (1-\rho^2)\sigma_1^2\right)$$

$$\boxed{P(X_1 | X_2=0) \sim N(4.5, 2.5)}$$

3c) $p(X_2) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} X$ $\quad A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

$$\boxed{p(X_2) \sim N(A\mu, A\Sigma A^T) = N(-1, 2)}$$

4a) $\text{MAP} = \dfrac{p(x|y)\cdot p(y)}{p(x)}$ $\qquad p(x) = \frac{1}{2}\sum_{i=0}^{1} p(x|y=i)$

$$p(x|y=0) = \frac{1}{2\pi|\Sigma|^{1/2}} e^{\left(-\frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)\right)}$$

$$p(x|y=1) = \frac{1}{2\pi|\Sigma|^{1/2}} e^{\left(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)}$$

$$p(y=0|x) = \frac{\frac{1}{2}N(\mu_0,\Sigma_0)}{\frac{1}{2}(N(\mu_0,\Sigma_0)+N(\mu_1,\Sigma_1))}$$

$$\log(p(y=0|x)) = \log(N(\mu_0,\Sigma_0)) - \log(N(\mu_0,\Sigma_0)+N(\mu_1,\Sigma_1))$$

$$\log(p(y=1|x)) = \log(N(\mu_1,\Sigma_1)) - \log(N(\mu_0,\Sigma_0)+N(\mu_1,\Sigma_1))$$

$$\frac{p(y=0|x)}{p(y=1|x)} \gtrless 1 \rightarrow \log\left(\frac{p(y=0|x)}{p(y=1|x)}\right) \lesseqgtr 0$$

so $\log(p(y=0|x)) - \log(p(y=1|x)) \geq 0$

$\log(p(y=0|x)) \lesseqgtr \log(p(y=1|x))$

which simplifies to

$$\frac{1}{2}\log|\Sigma_0| - \frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0) \gtrless \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)$$

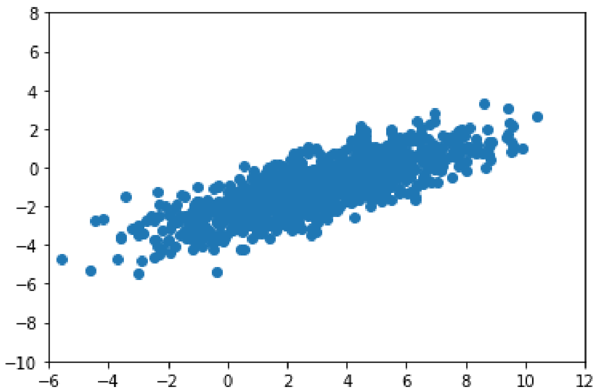
Question 3a

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
sig = np.array([[7,3],[3,2]])
L = np.linalg.cholesky(sig)
```

In [ ]:
```
x1 = []
x2 = []
for i in range(1000):
  x = np.random.randn(2,1)
  temp = L@x
  x1.append(temp[0]+3)
  x2.append(temp[1]-1)
```

In [ ]:
```
plt.scatter(x1,x2)
plt.ylim((-10,8));
plt.xlim((-6,12));
```



In [ ]:

Question 4

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
# Vector Means
mu0 = np.array([[1],[0]])
mu1 = np.array([[0],[2]])

# Vector Covariance Matrices
sig0 = np.array([[8,3],[3,2]])
sig1 = np.array([[1,0.1],[0.1,1]])

# Inverse Covariance Matrices
inv_sig0 = np.linalg.inv(sig0)
inv_sig1 = np.linalg.inv(sig1)

# Covariance Matrix Determinants
det0 = np.linalg.det(sig0)
det1 = np.linalg.det(sig1)

# Linear Transformations
A0 = np.linalg.cholesky(sig0)
A1 = np.linalg.cholesky(sig1)
```

In [3]:
```
def classify(x):
  dx0 = (x-mu0)
  y0 = (-0.5*np.log(det0))-(0.5*dx0.T@inv_sig0@dx0)
  dx1 = (x-mu1)
  y1 = (-0.5*np.log(det1))-(0.5*dx1.T@inv_sig1@dx1)
  if y0 >= y1:
    return True
  else:
    return False
```

In [4]:
```
xy0 = np.zeros((1000,2))
xy1 = np.zeros((1000,2))
misclass = 0
for i in range(1000):
  x = np.random.randn(2,1)
  temp = (A0@x)+mu0
  xy0[i] = [temp[0],temp[1]]
  if classify(temp):
    misclass+=1
  x = np.random.randn(2,1)
  temp = (A1@x)+mu1
  xy1[i] = [temp[0],temp[1]]
  if not classify(temp):
    misclass+=1
```

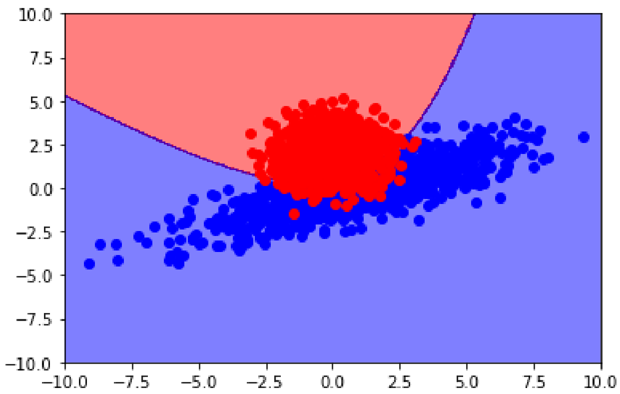
```
<ipython-input-4-ba9f797bb53b>:7: DeprecationWarning: setting an array element with a sequence. This was supporte
d in some cases where the elements are arrays with a single element. For example `np.array([1, np.array([2])], dt
ype=int)`. In the future this will raise the same ValueError as `np.array([1, [2]], dtype=int)`.
  xy0[i] = [temp[0],temp[1]]
<ipython-input-4-ba9f797bb53b>:12: DeprecationWarning: setting an array element with a sequence. This was support
ed in some cases where the elements are arrays with a single element. For example `np.array([1, np.array([2])], d
type=int)`. In the future this will raise the same ValueError as `np.array([1, [2]], dtype=int)`.
  xy1[i] = [temp[0],temp[1]]
```

In [5]:
```
nn = 400
x1g = np.linspace(-10, 10, nn)
x2g = np.linspace(-10, 10, nn)
decisions = -1*np.ones((nn,nn))

for i, x1 in enumerate(x1g):
  for j, x2 in enumerate(x2g):
    x = np.array([[x1],[x2]])
    if (classify(x)):
      decisions[j,i] = 1
```

In [6]:
```
plt.figure()
plt.contourf(x1g, x2g, decisions,colors=['red','blue'],alpha=0.5)
'blue')
plt.scatter(xy1[:,0],xy1[:,1],color='red')
```

Out[6]: <matplotlib.collections.PathCollection at 0x238f19fd700>



In [7]:
```
print("Misclassification Rate: %f" %(1-(misclass/2000)))
```

```
Misclassification Rate: 0.084500
```

In [ ]: