In [44]:
```
import numpy as np
```

UFRGS - PPGEE - ENG405

Inferência Bayesiana e Teoria Evidências de Depster-Shafer

Túlio Dapper e Silva (194878)

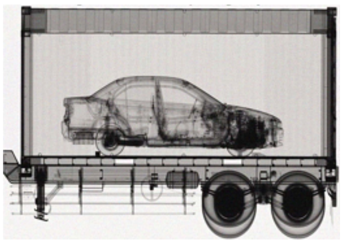
1 - Bayesian Inference

Multi-sensor data fusion application for Cargo Screening

A Bayesian approach

2010 2nd International Conference on Computer Technology and Development (ICCTD 2010)

Akiwowo Ayodeji O. and Efekhari Mahroo M.



Cargo Screening

x_1: state of the target at 'time' 1.

x = s_1, s_2 e s_3 (substances)

Sensor X (X = A, B)

y_1^X: observation made of target at 'time' 1 by sensor X.

Y_0^X: set of old data collected by sensor X.

Y_1^X: set of all observations made of the target by sensor X up to present time. $Y_1^X = Y_0^X + y_1^X$

Bayesian Inference

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$P(x_1|Y_1^X) = \frac{P(y_1^X|x_1)P(x_1|Y_0^X)}{P(y_1|Y_0^X)}$$

$P(y_1^X|x_1)$: likelihood

$P(x_1|Y_0^X)$: prior probability

Data Fusion

$$P(x=s_k|Y_1^A Y_1^B) = \frac{P(x=s_k|Y_1^A)P(x=s_k|Y_1^B)P(x=s_k|Y_0^A Y_0^B)}{P(x=s_k|Y_0^A)P(x=s_k|Y_0^B)}$$

$k = 1, 2, 3$

In [183]:
```
fuseB  = lambda p_y_1_, p_y_0_, p_yf_0_: p_y_1_[0]*p_y_1_[1]*p_yf_0_/(p_y_0_[0]*p_y_0_[1])
```

Experimental

Sensor A

$P(x=s_1|Y_0^A) = 0.4$

$P(x=s_2|Y_0^A) = 0.3$

$P(x=s_3|Y_0^A) = 0.3$

Sensor B

$P(x=s_1|Y_0^B) = 0.5$

$P(x=s_2|Y_0^B) = 0.3$

$P(x=s_3|Y_0^B) = 0.2$

Sensor A

$P(x=s_1|Y_1^A) = 0.64$

$P(x=s_2|Y_1^A) = 0.22$

$P(x=s_3|Y_1^A) = 0.14$

Sensor B

$P(x=s_1|Y_1^B) = 0.76$

$P(x=s_2|Y_1^B) = 0.21$

$P(x=s_3|Y_1^B) = 0.03$

Initial Values

$P(x=s_1|Y_0^A Y_0^B) = 0.5$

$P(x=s_2|Y_0^A Y_0^B) = 0.3$

$P(x=s_3|Y_0^A Y_0^B) = 0.2$

In [41]:
```
p_yf = [.5, .3, .2]
p_yA = np.array([[.4, .3, .3],[.64, .22, .14]])
p_yB = np.array([[.5, .3, .2],[.76, .21, .03]])
```

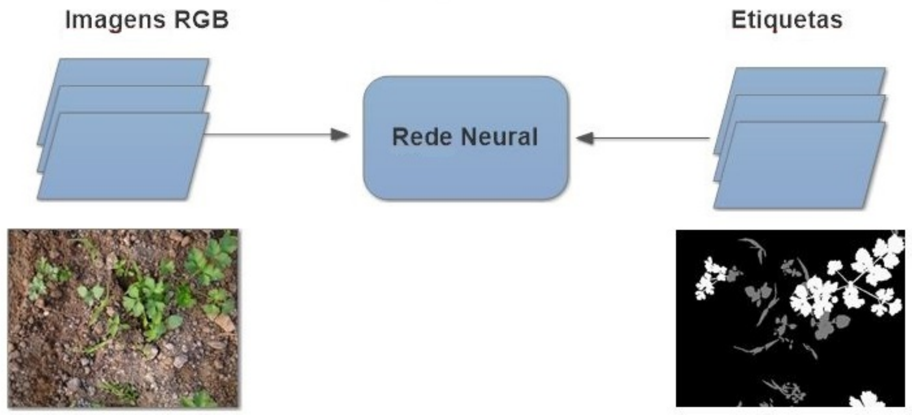
In [184]:
```
for s in range(0,3):
    p_yf[s]  = fuseB([p_yA[1,s], p_yB[1,s]], [p_yA[0,s], p_yB[0,s]], p_yf[s])
p_yf=p_yf/np.sum(p_yf)

print("Probabilities of being s1, s2 and s3")
print(p_yf*100)
```

```
Probabilities of being s1, s2 and s3
[9.73650236e+01 2.60271140e+00 3.22650174e-02]
```

Extra

Detecting Weed in Agriculture



Bayesian Inference

$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

Application

A : It is weed.

B_1: Presence or high probability of having weed within a certain distance.

B_2: High humidity.

$P(B_1B_2|A) = 90\%$

$P(B_1B_2|\bar{A}) = 5\%$

Let's suppose at a moment...

$P(A) = 70\%$

$P(B_1B_2) = P(A)P(B_1B_2|A) + P(\bar{A})P(B_1B_2|\bar{A}) = 0.7 * 0.9 + 0.3 * 0.05 = 0.645$

$P(A|B1B2) = 0.7 * 0.9/0.645 = 0.976 = 97.6\%$

2 - Teoria Evidências de Depster-Shafer

An Introduction to Bayesian and Dempster-Shafer Data Fusion

Don Koks and Subhash Challa

Aircraft Identification

Two sensors (1 and 2) detecting either a F-111, a F/A-18 or a P-3C Orion.
aviao.jpg

Dempster-Shafer

Elements

F-111, F/A-18, P-3C Orion

Subsets

F-111 = {F-111}

F/A-18 = {F/A-18}

P-3C = {P-3C}

Fast = {F-111, F/A-18}

Unknown = {F-111, F/A-18, P-3C}

Mc: Content Matrix

(the elements of the left edge subsets belong to top edge subsets)

$$M_c = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ms: Support Matrix

(top edge subsets give support to left edge subsets)

$$M_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Mp: Plausability Matrix

(top edge subsets do not contradict the left edge subsets)

$$M_p = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Fusion Data

$$m^{1,2}(C) = k \sum_{A \cap B = C} m^1(A) m^2(B)$$

$$m^{1,2} = r(M_1(M_c M_2))^T + r(M_2(M_c M_1))^T - r(M_1 M_2)$$

where:

$r = ones(size(m(A)))$

$$M_1 = \begin{bmatrix} m(A,1) & 0 & 0 & 0 \\ 0 & m(A,2) & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & m(A, size(m(A))) \end{bmatrix}$$

$$M_2 = \begin{bmatrix} m(B,1) & 0 & 0 & 0 \\ 0 & m(B,2) & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & m(B, size(m(B))) \end{bmatrix}$$

Sensor Samples

1st Sample

Target type	Sensor 1 (mass m^1)	Sensor 2 (mass m^2)
F-111	30%	40%
F/A-18	15%	10%
P-3C	3%	2%
Fast	42%	45%
Unknown	10%	3%
Total mass	100%	100%

2nd Sample

Target type	Sensor 1 (mass m^1)	Sensor 2 (mass m^2)
F-111	30%	50%
F/A-18	15%	30%
P-3C	3%	17%
Fast	42%	
Unknown	10%	3%

In [180]:

```
def fuseD(mA, mB, M):
    N = np.size(mA)
    MA = np.zeros([N,N])
    MB = np.zeros([N,N])
    r = np.ones([1,N])
    for i in range(0, N):
        MA[i,i]=mA[i]
        MB[i,i]=mB[i]
    x = np.dot(r,np.transpose(np.dot(MA,np.dot(M,MB))))+np.dot(r,np.transpose(np.dot(MB,np.dot(M,MA))))-np.dot(r,np.dot(MA,MB
))
    return x[0,:]/sum(x[0,:])
suppla = lambda m, M: np.dot(m, np.transpose(M))
```

In [182]:

```
M = np.array([[1, 0, 0, 1, 1], [0, 1, 0, 1, 1], [0, 0, 1, 0, 1], [0, 0, 0, 1, 1], [0, 0, 0, 0, 1]])
Ms = np.array([[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [0, 0, 1, 0, 0], [1, 1, 0, 1, 0], [1, 1, 1, 1, 1]])
Mp = np.array([[1, 0, 0, 1, 1], [0, 1, 0, 1, 1], [0, 0, 1, 0, 1], [1, 1, 0, 1, 1], [1, 1, 1, 1, 1]])

mA = np.array([[.3, .15, .03, .42, .1],[.3, .15, .03, .42, .1]])
mB = np.array([[.4, .1, .02, .45, .03],[.5, .3, .17, 0, .03]])

print("Legend")
print("   F-111   |   F/A-18   |    P-3C    |   Fast   |   Unknown")
print("")

mF1 = fuseD(mA[0,:], mB[0,:], Mc);
print("1st Sample - Fused Masses (%)")
print(mF1*100)
print("")

SA1 = suppla(mF1, Ms)
print("1st Sample - Support (%)")
print(SA1*100)
print("")

PA1 = suppla(mF1, Mp)
print("1st Sample - Plausability (%)")
print(PA1*100)
print("")

mF2 = fuseD(mA[1,:], mB[1,:], Mc);
print("2nd Sample - Fused Masses (%)")
print(mF2*100)
print("")

SA2 = suppla(mF2, Ms)
print("2nd Sample - Support (%)")
print(SA2*100)
print("")

PA2 = suppla(mF2, Mp)
print("2nd Sample - Plausability (%)")
print(PA2*100)
print("")
```

```
Legend
   F-111   |   F/A-18   |    P-3C    |   Fast   |   Unknown

1st Sample - Fused Masses (%)
[54.62330749 16.08610115  0.40504571 28.53836362  0.34718204]

1st Sample - Support (%)
[ 54.62330749  16.08610115   0.40504571  99.24777225 100.        ]

1st Sample - Plausability (%)
[ 83.50885314  44.9716468    0.75222775  99.59495429 100.        ]

2nd Sample - Fused Masses (%)
[63.1880561  30.99080078  3.46855678  1.90016589  0.45242045]

2nd Sample - Support (%)
[ 63.1880561   30.99080078   3.46855678  96.07902277 100.        ]

2nd Sample - Plausability (%)
[ 65.54064244  33.34338712   3.92097723  96.53144322 100.        ]
```