Machine Learning
MSE FTP MachLe
Christoph Würsch

OST
Ostschweizer
Fachhochschule

Inspector Clouseau: Extended Solution

- Author: Christoph Würsch
- Subject: MSE TSM_MachLe
- Bayesian Theorem, Probabilistic Thinking

Lets first import pandas and define the quantities that are given by the excercise.

The nomenclature is as follows:

conditional probabilities:

- $p(y|x) =$ `p1y_1x`
- $p(y|\bar{x}) =$ `p1y_0x`
- $p(\bar{y}|x) =$ `p0y_1x`
- $p(\bar{y}|\bar{x}) =$ `p0y_0x`

In [1]:
```
import pandas as pd

# prior probabilities
p1B = 0.6
p0B = 1-p1B
p1M = 0.2
p0M = 1-p1M
```

conditional probabilities

$$p(K|B,M) = 0.1$$

$$p(K|B,\bar{M}) = 0.6$$

$$p(K|\bar{B},M) = 0.2$$

$$p(K|\bar{B},\bar{M}) = 0.3$$

In [2]:
```
#conditional probabilities for the knife 1K means K=True
p1K_1B1M=0.1
p1K_1B0M=0.6
p1K_0B1M=0.2
p1K_0B0M=0.3

#conditional probabilities for not the knife 0K means K=False
p0K_1B1M=1-p1K_1B1M
p0K_1B0M=1-p1K_1B0M
p0K_0B1M=1-p1K_0B1M
p0K_0B0M=1-p1K_0B0M
```

Using **Bayes Theroem** we can invert the conditonal probabilities.

$$p(B|K)=\frac{p(K|B)\cdot p(B)}{p(K)}$$

To calculate the prior $p(K)$, we even have to marginalize over the states $b\in\mathrm{dom}(B)$ of the butler B.

$$p(B|K)=\frac{p(K|B)\cdot p(B)}{p(K)}=\frac{p(K|B)\cdot p(B)}{\sum_{b\in\mathrm{dom}(B)}p(K|B)\cdot p(B)}$$

Since the conditionals also depend on the state of the Maid M, we have to marginalize over the states $m\in\mathrm{dom}(M)$ of the Maid M. If we know the full joint probability density $p(k,b,m)$ for all states of K,B and M, we could just marignalize over the joint probability distribution:

$$p(K)=\sum_{b\in\mathrm{dom}(B)}\left\{\sum_{m\in\mathrm{dom}(M)}p(K,b,m)\right\}$$

But now that we are only given the *conditional probabilites*, we have to use **Bayes' therorem** to calculate the joint probability distribution:

$$p(k,b,m)=p(k|b,m)\cdot p(b)\cdot p(m)$$

So we have to sum up over the conditional multiplied by the priors $p(b)$ and $p(m)$:

$$p(K)=\sum_{b\in\mathrm{dom}(B)}\left\{\sum_{m\in\mathrm{dom}(M)}p(K|b,m)\cdot p(m)\right\}\cdot p(b)$$

Using this, we get:

$$p(B|K)=\frac{\left\{\sum_{m\in\mathrm{dom}(M)}p(K|B,m)\cdot p(m)\right\}\cdot p(B)}{\sum_{b\in\mathrm{dom}(B)}\left\{\sum_{m\in\mathrm{dom}(M)}p(K|b,m)\cdot p(m)\right\}\cdot p(b)}$$

In [3]:

```
# (a) calculate the probability that the butler ist the murderer
# given the fact that a knife was found as the corpus delicti
# p(B | K)

num=p1B*(p1K_1B1M*p1M+p1K_1B0M*p0M)
p1K=p1B*(p1K_1B1M*p1M+p1K_1B0M*p0M)+p0B*(p1K_0B1M*p1M+p1K_0B0M*p0M)
p0K=p1B*(p0K_1B1M*p1M+p0K_1B0M*p0M)+p0B*(p0K_0B1M*p1M+p0K_0B0M*p0M)

p1B_1K=num/p1K

print('The marginal for K=1 is p(K=1): ', p1K)
print('The marginal for K=0 is p(K=0): ', p0K)
print('test: p(K=0)+p(K=1): ', p0K+p1K)

print('p(M | K):', p1B_1K)
```

```
The marginal for K=1 is p(K=1):  0.41200000000000003
The marginal for K=0 is p(K=0):  0.5880000000000001
```

```
test: p(K=0)+p(K=1):  1.0
p(M | K): 0.7281553398058251
```

In [4]:

```
# calculation of the joint probability distribution
p1K1B1M=p1K_1B1M*p1B*p1M
p1K1B0M=p1K_1B0M*p1B*p0M
p1K0B1M=p1K_0B1M*p0B*p1M
p1K0B0M=p1K_0B0M*p0B*p0M

p0K1B1M=p0K_1B1M*p1B*p1M
p0K1B0M=p0K_1B0M*p1B*p0M
p0K0B1M=p0K_0B1M*p0B*p1M
p0K0B0M=p0K_0B0M*p0B*p0M
```

In [5]:

```
p= [p1K1B1M, p1K1B0M, p1K0B1M, p1K0B0M, p0K1B1M, p0K1B0M, p0K0B1M, p0K0B0M]
#create a dictionary and later a dataframe
d={'K': [1,1,1,1,0,0,0,0], 'B': [1,1,0,0,1,1,0,0], 'M': [1,0,1,0,1,0,1,0], 'p(K,B,M)':p


df=pd.DataFrame(data=d)

df.head(8)
```

Out[5]:

	K	B	M	p(K,B,M)
0	1	1	1	0.012
1	1	1	0	0.288
2	1	0	1	0.016
3	1	0	0	0.096
4	0	1	1	0.108
5	0	1	0	0.192
6	0	0	1	0.064
7	0	0	0	0.224

Let's test whether we did all calculations right: The probabilities must sum up to one.

In [6]:

```
df.iloc[:,3].sum()
```

Out[6]:

```
1.0
```

Now, we can easily calculate the marginals:

Marginal for the **Knife=True**:

$$p(K=\text{True}) = \sum_{b\in\text{dom}(B)} \left\{ \sum_{m\in\text{dom}(M)} p(K=\text{True}, b, m) \right\}$$

In [7]:

```
#marginal for the Knife K
print('p(K=True)=',df[df['K']==1].iloc[:,3].sum())
print('p(K=False)=',df[df['K']==0].iloc[:,3].sum())
```

```
p(K=True)= 0.41200000000000003
p(K=False)= 0.5880000000000001
```

Marginal for the **Butler=True**:

$$p(B = \text{True}) = \sum_{k \in \text{dom}(K)} \left\{ \sum_{m \in \text{dom}(M)} p(k, B = \text{True}, m) \right\}$$

In [8]:
```
#marginal for the Butler B
print('p(B=True)=',df[df['B']==1].iloc[:,3].sum())
print('p(B=False)=',df[df['B']==0].iloc[:,3].sum())
```

```
p(B=True)= 0.6000000000000001
p(B=False)= 0.4
```

Marginal for the **Maid=True**:

$$p(M = \text{True}) = \sum_{k \in \text{dom}(K)} \left\{ \sum_{b \in \text{dom}(B)} p(k, b, M = \text{True}) \right\}$$

In [9]:
```
#marginal for the MAid M
print('p(M=True)=',df[df['M']==1].iloc[:,3].sum())
print('p(M=False)=',df[df['M']==0].iloc[:,3].sum())
```

```
p(M=True)= 0.2
p(M=False)= 0.8
```

In [ ]: