

1. Pre-requisitos

1. Asegúrese de tener instalado Python en versión 3.7 o superior.
2. Para este taller usaremos Python y en particular la librería *pgmpy*. Instale *pgmpy* con el comando

```
pip install pgmpy
```


o si prefiere repositorios de anaconda use

```
conda install -c ankurankan pgmpy
```


También puede usar el administrador de paquetes de Anaconda.
3. La entrega de este taller consiste en un **reporte** y unos **archivos de soporte**. Cree el archivo de su **reporte** como un documento de texto en el que pueda fácilmente incorporar capturas de pantalla, textos y similares. Puede ser un archivo de word, libre office, markdown, entre otros.
4. En su **reporte** incluya las respuestas solicitadas en el taller, marcando la sección y el numeral al que se refieren.

2. Generando datos y estimando parámetros

1. El punto de inicio de esta sección es su modelo del taller 1 para el modelo de la alarma antirrobo.
2. Usando este modelo, use el módulo de muestreo de *pgmpy* para generar 100000 número aleatorios de acuerdo con la distribución definida por el grafo

```
from pgmpy.sampling import BayesianModelSampling
samples = BayesianModelSampling(modelo).forward_sample(size=int(1e5))
print(samples.head())
```
3. Cree un nuevo modelo con la misma **estructura** del inicial, pero **sin parámetros**.
4. Emplee el módulo de ajuste de *pgmpy* para ajustar algunas CPDs del nuevo modelo.

```
from pgmpy.estimators import MaximumLikelihoodEstimator
env = MaximumLikelihoodEstimator(model=mod_fit_mv, data=samples)

# Estimar para nodos sin padres
cpdem_r = env.estimate_cpd(node="Robo")
print(cpdem_r)
cpdem_s = env.estimate_cpd(node="Sismo")
print(cpdem_s)

# Estimar para nodo Alarma
cpdem_a = env.estimate_cpd(node="Alarma")
print(cpdem_a)
```

Incluya el resultado en su **reporte**.

5. Emplee el método fit para ajustar todas las CPDs al tiempo y explore el resultado

```
mod_fit_mv.fit(data=samples, estimator = MaximumLikelihoodEstimator)
for i in mod_fit_mv.nodes():
    print(mod_fit_mv.get_cpds(i))
```

Incluya el resultado en su **reporte**.

6. Defina ahora un estimador Bayesiano en pgmpy

```
from pgmpy.estimators import BayesianEstimator
eby = BayesianEstimator(model=mod_fit_by, data=samples)
```

7. Realice nuevamente la estimación del modelo usando un prior

```
cpdby_r = eby.estimate_cpd(node="Robo", prior_type="dirichlet",
    pseudo_counts=[[100000], [200000]])
print(cpdby_r)
```

Modifique los valores de pseudocounts e interprete los resultados. Incluya sus resultados y observaciones en su **reporte**.

y

```
html.Div(
    className="Columnas",
    children=[
        html.Ul(id='my-list', children=[html.Li(i) for i in df.columns])
    ],
)
```

3. Ahora con otra red y datos

Realice el proceso de estimación de la red en la gráfica, empleando los datos del archivo adjunto. Estudie los parámetros obtenidos en su reporte.

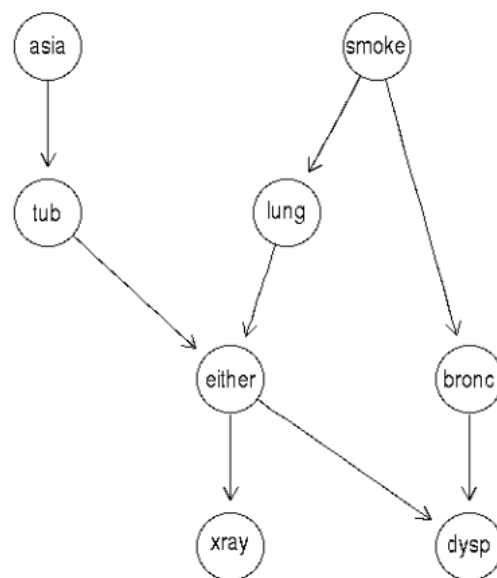


Figure 1: Grafo de una red Bayesiana para diagnosticar cáncer de pulmón