Bayesian network - Exact inference by variable elimination

- Bayesina networks help solving problems that provide limited information and resources
- Bayesian Networks are used to model uncertainties by using graphs
- These graphs also known as a "Bayes net"

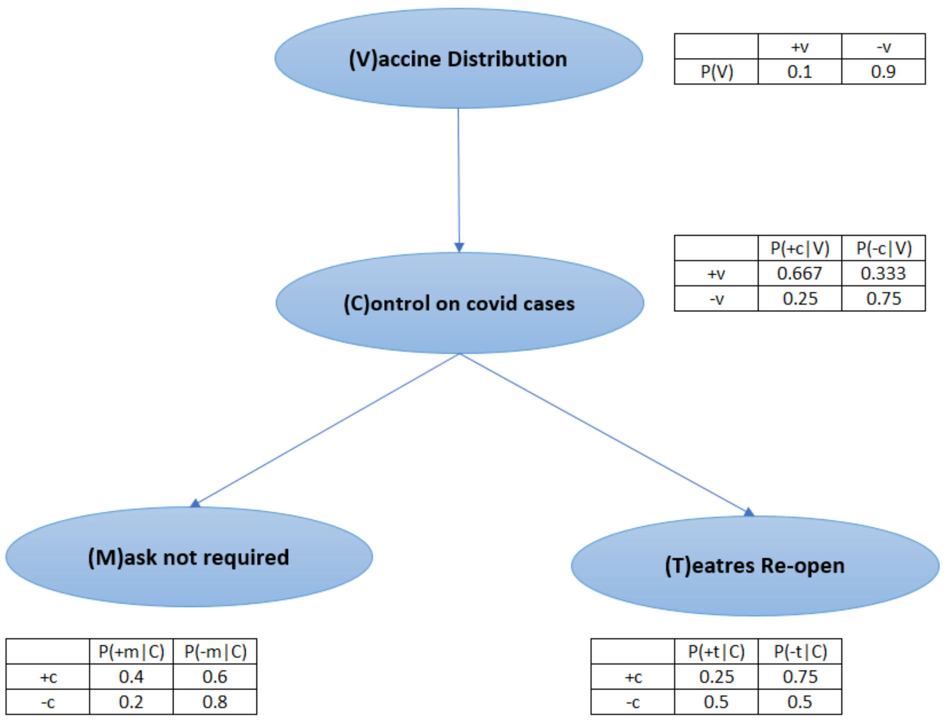
TASK1

Joint probability distribution calculation task - 3 marks

Whether vaccine distribution(V) will have an effect on controlling covid cases(C), which then influence the re-opening of thatres(T) and to get rid of masks(M). With the knowledge of probability, let's try to analyse the situation with the Bayes net below.

The full joint distribution is as mentioend below. Fill out the missing values.

- Remove `"YOUR ANSWER"` and write your answer there.



	+v	-v
P(V)	0.1	0.9

	P(+c|V)	P(-c|V)
+v	0.667	0.333
-v	0.25	0.75

	P(+m|C)	P(-m|C)
+c	0.4	0.6
-c	0.2	0.8

	P(+t|C)	P(-t|C)
+c	0.25	0.75
-c	0.5	0.5

V	C	T	M	P(V,C,T,M)

+	+	+	+	1/150
+	+	+	-	(1/10)(2/3)(4/10)(3/4) = 24/1200 = 1/50
+	+	-	+	1/100
+	+	-	-	(1/10)(2/3)(6/10)(3/4) = 36/1200 = 3/100
+	-	+	+	1/300
+	-	+	-	1/300
+	-	-	+	(1/10)(1/5)(8/10)(1/2) = 8/1000 = 2/250
+	-	-	-	1/75
-	+	+	+	**(9/10)(1/4)(1/4)(4/10) = 9/400**
-	+	+	-	27/400
-	+	-	+	**(9/10)(1/4)(3/4)(4/10) = 27/400**
-	+	-	-	81/800
-	-	+	+	27/400
-	-	+	-	27/400
-	-	-	+	**(9/10)(3/4)(1/2)(2/10) =27/400**
-	-	-	-	27/100

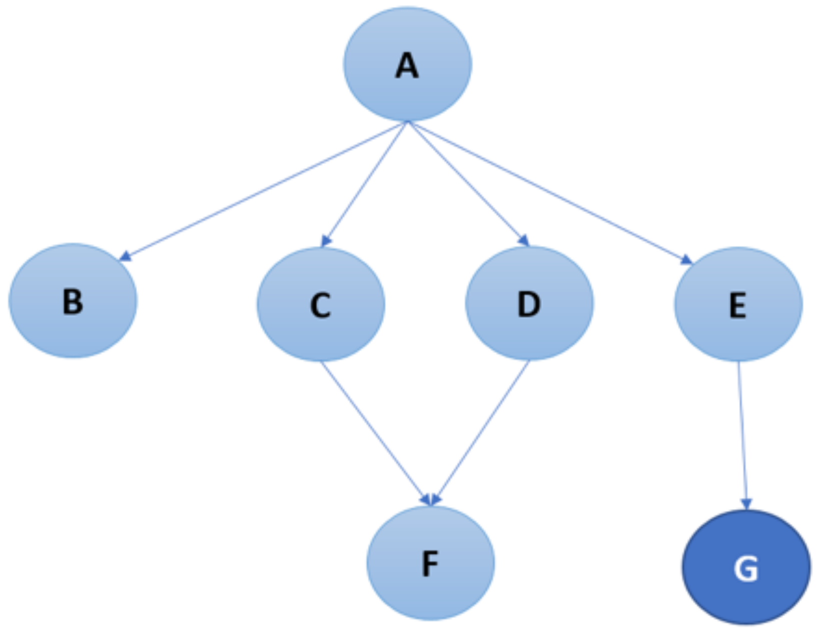
Vaiable Elimination

- Variable elimination (VE) is a simple and general exact inference algorithm in Bayesian networks.
- we loop over the variables in order and eliminate the hidden variables in provided ordering
- Variable elimination can be performed in different ordering
- The basic concept of variable elimination is that it avoids computing the Joint Distribution by doing marginalization over much smaller factors. So basically if we want to eliminate X from our distribution, then we compute the product of all the factors involving X and marginalize over them, thus allowing us to work on much smaller factors

Task2

For the given Bayes Net, calculate P(F/+g). All variables have binary domains. Assume we run variable elimination to compute the answer to this query with the the `"E,A,B,C,D"` variable elimination ordering. Cmpute the factors generated in this process. Start with the following factor.

`P(A),P(B/A),P(C/A),P(D/A),P(E/A),P(F/C,D),P(+g/E)`



- STEP 1: When eliminating E, new factor f1 will be generated as follows:
  - Filter probabilities where E is involved - P(E/A) and P(+g/E)
  - Now, remove E and generate factor f1
  - `f1(A,+g) = Σ P(E/A)P(+g/E)`
  - Now, rewrite the probability distribution with updated factor f1
  - New distribution = `P(A)P(B/A)P(C/A)P(D/A)P(F/C,D)f1(A,+g)`
- STEP 2: When eliminating A, new factor f1 will be generated as follows:
  - Filter probabilities where A is involved - P(A)P(B/A)P(C/A)P(D/A)f1(A,+g)
  - Now, remove A and generate factor f2
  - f2(B,C,D,+g) = `Σ P(A)P(B/A)P(C/A)P(D/A)f1(A,+g)`
  - Now, rewrite the probability distribution with updated factor f2
  - New distribution = `P(F/C,D)f2(B,C,D,+g)`

Perform same tasks for B,C & D - write your answer below (2 Marks)

- STEP 3: When eliminating B, new factor f1 will be generated as follows:

   - Filter probabilities where B is involved - `P(B/A)`
   - Now, remove B and generate factor f3
   - `f3(C,D,+g) = Σ f2(B,C,D,+g)`
   - Now, rewrite the probability distribution with updated factor f3
   - New distribution = `P(F/C,D)f3(C,D,+g)`
- STEP 4: When eliminating C, new factor f1 will be generated as follows:
   - Filter probabilities where C is involved - `P(F/C,D)f3(C,D,+g)`
   - Now, remove C and generate factor f4
   - `f4(F,D,+g) = Σ P(F/C,D)f3(C,D,+g)`
   - Now, rewrite the probability distribution with updated factor f4
   - New distribution = `P(F,D)f4(D,+g)`
- STEP 5: When eliminating D, new factor f1 will be generated as follows:
   - Filter probabilities where D is involved - `P(F,D)f4(D,+g)`
   - Now, remove D and generate factor f5
   - `f5(F,+g) = Σ P(F,D)f4(D,+g)`
   - Now, rewrite the probability distribution with updated factor f5
   - New distribution = `f5(F,+g)`

Try validating your answer on this interactive web tool:
http://pgmlearning.herokuapp.com/vElimApp

Task3: Programming of variable elimination

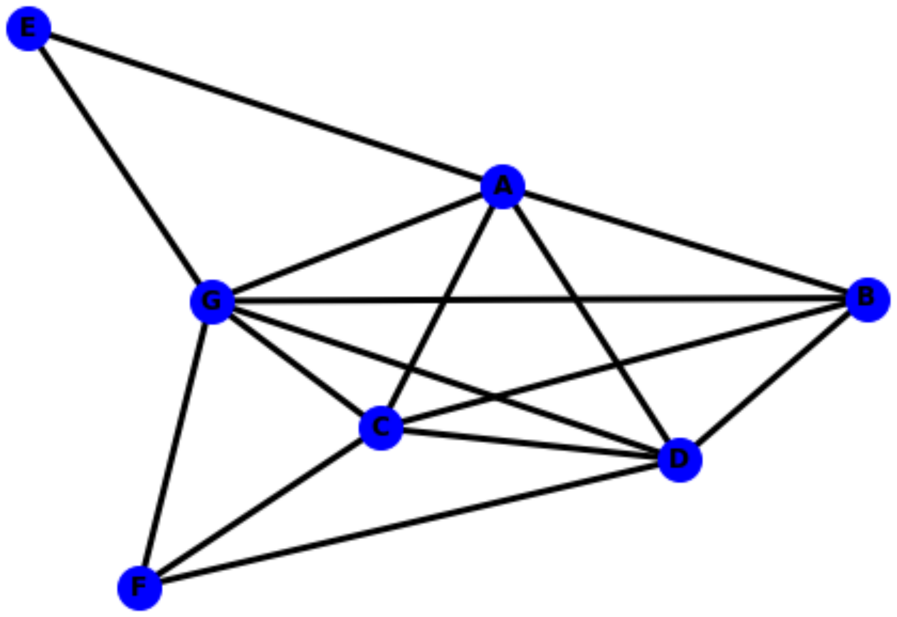
Python libraries for bayesian theorm, Exact inference by enumeration and variable elimination problems.

- https://github.com/petermlm/ProbPy
- https://pgmpy.org/ (For variable elimination refer - https://pgmpy.org/inference.html#variable-elimination)
- https://pyagrum.readthedocs.io/en/0.18.0/ (visualization support is very good in this library)
   - Example: .ipynb file with visualizations)

All these libraries have a documentatin maintained so check this sites and try to deep drive more into the world of bayesian networks!

In [6]:
```
import numpy as np
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
from pgmpy.models import BayesianModel
from pgmpy.inference import VariableElimination


values = pd.DataFrame(np.random.randint(low=0, high=2, size=(1000, 7)),
                      columns=['A', 'B', 'C', 'D', 'E','F','G'])

#Define the bayesia network structure
node_edges = [('A','B'),('A','C'),('A','D'),('A','E'),('C','F'),('D','F'),('E
model = BayesianModel(node_edges)
model.fit(values)
inference = VariableElimination(model)
elimination_order = ['E','A','B','C','D','F','G']

# width is the defined as the number of nodes in the largest clique in the gr
print(inference.induced_width(elimination_order))  # This is the largest fact

# To know more about graph returned, please check official documentation at h
graph = inference.induced_graph(elimination_order)
print("Nodes of graph = ", list(graph.nodes))
print("Edges of graph = ", list(graph.edges))
print("Neighbours of A = ",list(graph.neighbors('A')))
print("Connected components of graph =", list(nx.connected_components(graph))


#ploting a graph
options = {
    'node_color': 'blue',
    'node_size': 400,
    'width': 3,
}
nx.draw(graph, with_labels=True, font_weight='bold', **options)
plt.show()
```

```
Finding Elimination Order: : 100%|██████████| 2/2 [27:05<00:00, 812.82s/it]
4
Nodes of graph =  ['A', 'G', 'D', 'F', 'B', 'C', 'E']
Edges of graph =  [('A', 'G'), ('A', 'D'), ('A', 'B'), ('A', 'C'), ('A', 'E'),
('G', 'F'), ('G', 'D'), ('G', 'C'), ('G', 'B'), ('G', 'E'), ('D', 'F'), ('D',
'C'), ('D', 'B'), ('F', 'C'), ('B', 'C')]
Neighbours of A =  ['G', 'D', 'B', 'C', 'E']
Connected components of graph = [{'C', 'E', 'A', 'F', 'G', 'D', 'B'}]
```



In [3]:

```
from pgmpy.factors.discrete import TabularCPD

vaccine_model = BayesianModel([('vaccine', 'covid_cases'),
                              ('covid_cases', 'mask'),
                              ('covid_cases', 'theatre')])

cpd_vaccine = TabularCPD(variable='vaccine', variable_card=2,
                         values=[[0.1], [0.9]])
cpd_covid_cases = TabularCPD(variable='covid_cases', variable_card=2,
                             values=[[0.667 ,0.25],[0.333,0.75]],
                             evidence=['vaccine'],
                             evidence_card=[2])
cpd_masks = TabularCPD(variable='mask', variable_card=2,
                       values=[[0.4 ,0.2],[0.6,0.8]],
                       evidence=['covid_cases'],
                       evidence_card=[2])
cpd_theatre = TabularCPD(variable='theatre', variable_card=2,
                         values=[[0.25 ,0.5],[0.75,0.5]],
                         evidence=['covid_cases'],
                         evidence_card=[2])

vaccine_model.add_cpds(cpd_vaccine, cpd_covid_cases, cpd_masks, cpd_theatre)
vaccine_model.check_model()
print(vaccine_model.is_active_trail('covid_cases', 'vaccine'))

vaccine_model.active_trail_nodes('mask')  # get all d-connected nodes
vaccine_model.local_independencies('theatre')  # Local independences for a no
vaccine_model.get_independencies()
vaccine_model.nodes()


vaccine_infer = VariableElimination(vaccine_model)

q = vaccine_infer.query(variables=['vaccine'], evidence={'mask': 0})
print(q)
```

```
Finding Elimination Order: :   0%|          | 0/2 [00:00<?, ?it/s]
  0%|          | 0/2 [00:00<?, ?it/s]
Eliminating: theatre:   0%|          | 0/2 [00:00<?, ?it/s]
Eliminating: covid_cases: 100%|██████████| 2/2 [00:00<00:00, 418.36it/s]
True
+------------+----------------+
| vaccine    |   phi(vaccine) |
+============+================+
| vaccine(0) |         0.1291 |
+------------+----------------+
| vaccine(1) |         0.8709 |
+------------+----------------+
```

Task3 - Programming variable elimination (5 marks)

- Inspect the starter code given below and check the characteristics of data and model
- Apply variable elimination to heart disease model and get infer
- Apply query to the geenerated infer and get probability of having heart disease when when "Age"=30 and "sex"=0

In [6]:
```
pip install pgmpy
```

```
Collecting pgmpy
  Downloading pgmpy-0.1.13-py3-none-any.whl (324 kB)
     |████████████████████████████████| 324 kB 4.0 MB/s eta 0:00:01
Collecting torch
  Downloading torch-1.8.0-cp39-none-macosx_10_9_x86_64.whl (120.6 MB)
     |████████████████████████████████| 120.6 MB 94.2 MB/s eta 0:00:01
Requirement already satisfied: pandas in /Library/Frameworks/Python.framework/
Versions/3.9/lib/python3.9/site-packages (from pgmpy) (1.2.3)
Collecting statsmodels
  Downloading statsmodels-0.12.2-cp39-cp39-macosx_10_15_x86_64.whl (9.6 MB)
     |████████████████████████████████| 9.6 MB 60.1 MB/s eta 0:00:01
Collecting tqdm
  Using cached tqdm-4.59.0-py2.py3-none-any.whl (74 kB)
Collecting joblib
  Using cached joblib-1.0.1-py3-none-any.whl (303 kB)
Requirement already satisfied: numpy in /Library/Frameworks/Python.framework/V
ersions/3.9/lib/python3.9/site-packages (from pgmpy) (1.20.1)
Collecting scikit-learn
  Downloading scikit_learn-0.24.1-cp39-cp39-macosx_10_13_x86_64.whl (7.3 MB)
     |████████████████████████████████| 7.3 MB 18.5 MB/s eta 0:00:01
Collecting scipy
  Downloading scipy-1.6.1-cp39-cp39-macosx_10_9_x86_64.whl (30.9 MB)
     |████████████████████████████████| 30.9 MB 20.7 MB/s eta 0:00:01
Collecting networkx
  Downloading networkx-2.5-py3-none-any.whl (1.6 MB)
     |████████████████████████████████| 1.6 MB 37.8 MB/s eta 0:00:01
Requirement already satisfied: pyparsing in /Library/Frameworks/Python.framewo
rk/Versions/3.9/lib/python3.9/site-packages (from pgmpy) (2.4.7)
Requirement already satisfied: decorator>=4.3.0 in /Library/Frameworks/Python.
framework/Versions/3.9/lib/python3.9/site-packages (from networkx->pgmpy) (4.4
.2)
Requirement already satisfied: python-dateutil>=2.7.3 in /Library/Frameworks/P
ython.framework/Versions/3.9/lib/python3.9/site-packages (from pandas->pgmpy)
(2.8.1)
Requirement already satisfied: pytz>=2017.3 in /Library/Frameworks/Python.fram
ework/Versions/3.9/lib/python3.9/site-packages (from pandas->pgmpy) (2020.5)
Requirement already satisfied: six>=1.5 in /Users/kavanchandra/Library/Python/
3.9/lib/python/site-packages (from python-dateutil>=2.7.3->pandas->pgmpy) (1.1
5.0)
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-2.1.0-py3-none-any.whl (12 kB)
Collecting patsy>=0.5
  Downloading patsy-0.5.1-py2.py3-none-any.whl (231 kB)
     |████████████████████████████████| 231 kB 46.1 MB/s eta 0:00:01
Collecting typing-extensions
  Downloading typing_extensions-3.7.4.3-py3-none-any.whl (22 kB)
Installing collected packages: typing-extensions, threadpoolctl, scipy, patsy,
joblib, tqdm, torch, statsmodels, scikit-learn, networkx, pgmpy
Successfully installed joblib-1.0.1 networkx-2.5 patsy-0.5.1 pgmpy-0.1.13 scik
it-learn-0.24.1 scipy-1.6.1 statsmodels-0.12.2 threadpoolctl-2.1.0 torch-1.8.0
tqdm-4.59.0 typing-extensions-3.7.4.3
Note: you may need to restart the kernel to use updated packages.
```

In [7]:
```
import numpy as np
import pandas as pd
from pgmpy.models import BayesianModel
from pgmpy.inference import VariableElimination

names = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
'thal', 'heartdisease']
heartDisease = pd.read_csv('heart.csv', names = names)
heartDisease = heartDisease.replace('?', np.nan)

model = BayesianModel([('age', 'trestbps'), ('age', 'fbs'), ('sex', 'trestbps
model.fit(heartDisease)

# Apply variable eliminatin to heart disease model and get infer
# Apply query to the geenerated infer and get probability of having heart dis
# Write your code here
infer = VariableElimination(model)
query = infer.query(variables=['heartdisease'], evidence={'age': 30, 'sex': 0
print(query)
```

```
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/
pgmpy/factors/discrete/DiscreteFactor.py:518: UserWarning: Found unknown state
name. Trying to switch to using all state names as state numbers
  warn(
Finding Elimination Order: :   0%|          | 0/6 [00:00<?, ?it/s]
  0%|          | 0/6 [00:00<?, ?it/s]
Finding Elimination Order: : 100%|██████████| 6/6 [00:00<00:00, 728.24it/s]

Eliminating: chol:   0%|          | 0/6 [00:00<?, ?it/s]
Eliminating: trestbps:   0%|          | 0/6 [00:00<?, ?it/s]
Eliminating: exang:   0%|          | 0/6 [00:00<?, ?it/s]
Eliminating: restecg:   0%|          | 0/6 [00:00<?, ?it/s]
Eliminating: thalach: 100%|██████████| 6/6 [00:00<00:00, 298.76it/s]
+-----------------------------+---------------------+
| heartdisease                |   phi(heartdisease) |
+=============================+=====================+
| heartdisease(0)             |              0.5728 |
+-----------------------------+---------------------+
| heartdisease(1)             |              0.1149 |
+-----------------------------+---------------------+
| heartdisease(2)             |              0.2545 |
+-----------------------------+---------------------+
| heartdisease(3)             |              0.0463 |
+-----------------------------+---------------------+
| heartdisease(4)             |              0.0114 |
+-----------------------------+---------------------+
| heartdisease(heartdisease)  |              0.0001 |
+-----------------------------+---------------------+
```

In [ ]: