Experiment 7 - Modelling Medical Data with a Bayesian Network

April 30, 2023

1 Experiment Details

1.1 Submitted By

Desh Iyer, 500081889, Year III, AI/ML(H), B5

1.2 Problem Statement

Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard *Heart Disease Dataset*.

1.3 Theory

Bayesian networks are probabilistic graphical models that use Bayesian inference to model probabilistic relationships between a set of variables. In medical diagnosis, Bayesian networks are commonly used to represent the probabilistic dependencies between the symptoms and the underlying diseases. The network consists of a set of nodes representing variables and edges representing the probabilistic dependencies between them.

The Heart Disease Data Set is a standard data set that contains information about patients who have heart disease or not. The data set consists of 14 attributes including age, sex, chest pain type, blood pressure, serum cholesterol level, fasting blood sugar, electrocardiographic results, maximum heart rate achieved, exercise-induced angina, ST depression induced by exercise, slope of the peak exercise ST segment, number of major vessels colored by fluoroscopy, thal, and the target variable that indicates whether or not the patient has heart disease.

The goal of the Bayesian network is to use the probabilistic dependencies between the attributes to predict the probability of heart disease given the observed symptoms.

1.4 Steps to construct a Bayesian network for medical data

1. Identify the set of relevant variables: In this case, the relevant variables are the attributes in the Heart Disease Data Set.
2. Define the structure of the network: The structure of the network can be determined based on domain knowledge or by using a learning algorithm such as the K2 algorithm or the hill climbing algorithm.
3. Assign probabilities to the nodes: Once the structure of the network is determined, probabilities need to be assigned to the nodes based on the data. This can be done using maximum likelihood estimation or Bayesian estimation.

4. Inference: After constructing the network and assigning probabilities, the network can be used for inference to predict the probability of heart disease given the observed symptoms.

2 Import libraries

[ ]:
```
import pandas as pd
import bnlearn as bn
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings("ignore")
```

3 Read Data from .csv File

[ ]:
```
data = pd.read_csv(r'./data.csv')
data
```

4 Exploring the Data

[ ]:
```
data.columns
```

[ ]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'num'],
      dtype='object')
```

[ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    object
 12  thal      303 non-null    object
```

```
 13  num       303 non-null    int64
dtypes: float64(1), int64(11), object(2)
memory usage: 33.3+ KB
```

[ ]:
```
data.describe()
```

[ ]:
```
              age         sex          cp    trestbps        chol         fbs  \
count  303.000000  303.000000  303.000000  303.000000  303.000000  303.000000
mean    54.438944    0.679868    3.158416  131.689769  246.693069    0.148515
std      9.038662    0.467299    0.960126   17.599748   51.776918    0.356198
min     29.000000    0.000000    1.000000   94.000000  126.000000    0.000000
25%     48.000000    0.000000    3.000000  120.000000  211.000000    0.000000
50%     56.000000    1.000000    3.000000  130.000000  241.000000    0.000000
75%     61.000000    1.000000    4.000000  140.000000  275.000000    0.000000
max     77.000000    1.000000    4.000000  200.000000  564.000000    1.000000

          restecg     thalach       exang     oldpeak       slope         num
count  303.000000  303.000000  303.000000  303.000000  303.000000  303.000000
mean     0.990099  149.607261    0.326733    1.039604    1.600660    0.937294
std      0.994971   22.875003    0.469794    1.161075    0.616226    1.228536
min      0.000000   71.000000    0.000000    0.000000    1.000000    0.000000
25%      0.000000  133.500000    0.000000    0.000000    1.000000    0.000000
50%      1.000000  153.000000    0.000000    0.800000    2.000000    0.000000
75%      2.000000  166.000000    1.000000    1.600000    2.000000    2.000000
max      2.000000  202.000000    1.000000    6.200000    3.000000    4.000000
```

5 Extracting X and y

[ ]:
```
X = data.iloc[:, :-1]
```

[ ]:
```
y = data['num']
```

6 Train-test Split

[ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, shuffle=True,␣
↪random_state=42)
```

[ ]:
```
training = pd.concat([X_train, y_train], axis='columns')
testing = pd.concat([X_test, y_test], axis='columns')
```

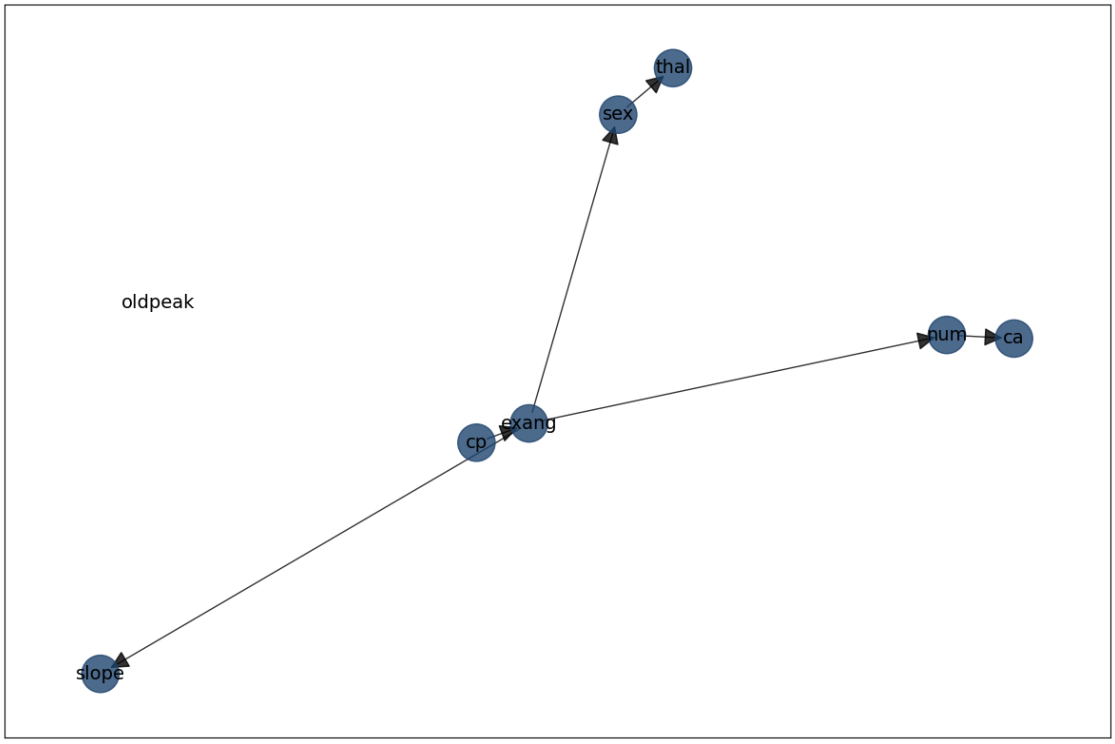
7 Plotting Bayesian Network

[ ]:
```
DAG = bn.structure_learning.fit(training, methodtype='hc', root_node='sex',␣
 ↪bw_list_method='nodes', verbose=3)

# Plot
G = bn.plot(DAG)

# Parameter learning
model = bn.parameter_learning.fit(DAG, training, verbose=3)
```

```
[bnlearn] >Warning: Computing DAG with 14 nodes can take a very long time!
[bnlearn] >Computing best DAG using [hc]
[bnlearn] >Set scoring type at [bic]
[bnlearn] >Compute structure scores ['k2', 'bds', 'bic', 'bdeu'] for model
comparison (higher is better).
[bnlearn] >Set node properties.
[bnlearn] >Set edge properties.
[bnlearn] >Plot based on Bayesian model
```




```
[bnlearn] >Parameter learning> Computing parameters using [bayes]
[bnlearn] >Converting [<class 'pgmpy.base.DAG.DAG'>] to BayesianNetwork model.
[bnlearn] >Converting adjmat to BayesianNetwork.
[bnlearn] >CPD of sex:
```