AU 332 Artificial Intelligence: Principles and Techniques

By: Chi Zhang (517021910658)

HW#: 4

November 24, 2019

I. INTRODUCTION

A. Equipment

There is a minimal amount of equipment to be used in this lab. The few requirements are listed below:

- Python 3.7.0 (Anaconda)
- Win10

B. function profiling

In this section how to realize the functions in **BayesianNetworks.py** will be introduced.

1. joinFactors(factor1, factor2)

The function returns a factor table that is the join of factor1 and factor2. The *pd.merge* is used to merge two factortables into one which is the join of tow tables. A new column named *'bridge'* is established to combine factor1 and factor2 to avoid there is no same coloum in two tables. For the new table, the probability coloum in factor1 and factor2 will be represented as 'probs_x' and 'probs_y'. The joint distribution probability can be computed as probs_x multiply probs_y. Then the needless 'probs' and 'bridge' can be deleted.

```python
def joinFactors(factor1, factor2):
    # your code
    if ( factor1.empty) or ( factor2.empty):
        return ( factor1 if factor2.empty else factor2)
    intersection =list((factor1.columns).intersection((factor2.columns)))
    intersection.remove('probs')
    # print(intersection)
    copy_factor1 =pd.DataFrame.copy(factor1)
    copy_factor2 =pd.DataFrame.copy(factor2)
    copy_factor1['bridge']=1
    copy_factor2['bridge']=1
    intersection.append('bridge')
    Factor =pd.merge(copy_factor1, copy_factor2, how='outer', on=intersection)
    # print(Factor)
    Factor['probs_x'] *=Factor['probs_y']
    # print(Factor)
    Factor =Factor.rename(columns={'probs_x':'probs'}).drop(columns=['probs_y','bridge'])
    return Factor
```

2. marginalizeFactor(factorTable, hiddenVar)

The function returns a factor table that marginalizes marginal variable 'hiddenVar' out of the 'factorTable'. If the hidden variable is not in the columns of factorTable, return the factorTable directly. Else, delete the hidden variable column. If current table only has the column of probability, return the factorTable directly. Else, the *pd.groupby* is used to group the table after deleting the column of hidden variable and the probability.

```python
def marginalizeFactor(factorTable, hiddenVar):
    # your code
    copy_factorTable =pd.DataFrame.copy(factorTable)
    if hiddenVar not in list(copy_factorTable.columns):
        # print('return raw')
        # print(list(copy_factorTable.columns),hiddenVar)
```

```
        return factorTable
    if hiddenVar in list(copy_factorTable.columns):
        # print(list(copy_factorTable.columns),hiddenVar)
        copy_factorTable =copy_factorTable.drop(columns=hiddenVar) # delete
        val_list =list(copy_factorTable.columns)
        val_list.remove('probs')
        if not val_list:
            return factorTable

        copy_factorTable =copy_factorTable[copy_factorTable.columns].groupby(val_list, as_index=False).mean()

        return copy_factorTable
```

3. marginalizeNetworkVariables(bayesNet, hiddenVar)

The function returns a Bayesian network containing a list of factor tables that results when the list of variables in hiddenVar is marginalized out of bayesnet. As the function can marginalize more than one hidden variable in a bayesNet, two loops are used as one loop for the factortables of the net and another loop for the variables which need to be marginalized. The function ***marginalizeFactor(factorTable, hiddenVar)*** is used here.

```
def marginalizeNetworkVariables(bayesNet, hiddenVar):
    # your code
    marginalized_bayesNet =[]
    for factorTable in bayesNet:
        # print(factorTable)
        copy_factorTable =pd.DataFrame.copy(factorTable)
        for hiddenVar_x in hiddenVar:
            result =marginalizeFactor(copy_factorTable, hiddenVar_x)
        # print(factorTable)
            copy_factorTable =pd.DataFrame.copy(result)

        marginalized_bayesNet.append(result)
    return marginalized_bayesNet
```

4. evidenceUpdateNet(bayesNet, evidenceVars, evidenceVals)

The function sets the values of the evidence variables. Other values for the variables should be removed from the tables. And the normalization of factors is not required. For each evidence variable *variable* and the corresponding variable value *value*, if the *variable* is included in the column of factorTable, the row whose value of the variable is equal to the *value* will be saved in the result.

```
def evidenceUpdateNet(bayesNet, evidenceVars, evidenceVals):
    # your code
    '''
    no need to normalize the factors
    '''
    current_net =bayesNet.copy()
    for variable,value in zip(evidenceVars, evidenceVals):
        net_for_loop =current_net.copy()
        current_net =[]
        for factorTable in net_for_loop:
            if variable in factorTable.columns:
                factorTable =factorTable[factorTable[variable]==int(value)] # leave the corresponding variable
                                                                              with required value
                current_net.append(factorTable)
```

```
        else:
            current_net.append(factorTable)
    return current_net
```

5. inference(bayesNet, hiddenVar, evidenceVars, evidenceVals)

This function takes in a Bayesian network and returns a single joint probability table resulting from the given set of evidence variables and marginalizing a set of hidden variables. The table will be normalized to give valid probabilities.The final table will be a proper probability table (entries sum to 1). The hidden variables shown in hiddenVar will not be in the returned table.

To realize the function, firstly the bayesNet will be updated with the provided hiddenVar and evidenceVars via the function ***evidenceUpdateNet(bayesNet, evidenceVars, evidenceVals)***. After filtering out all the variables in the columns, for each variable *variable*, a loop is used to dispose *variable* in the factorTable *factorTable* of the net. If *variable* is in the column of the *factorTable*, the ***joinFactors(factor, factorTable)*** will be used to join two factors. After the loop for the net, if the *variable* is one of the hidden variable, the ***marginalizeFactor(factor, variable)*** will be used to marginalize it.

To realize the normalization, the *norm_scale* is computed to normalize the probability whose entries sum to 1.

```
def inference(bayesNet, hiddenVar, evidenceVars, evidenceVals):
    # your code

    # update net with evidenceUpdateNet
    updated_net =evidenceUpdateNet(bayesNet,evidenceVars,evidenceVals)

    # filter all the variables
    all_variables =set()
    for factorTable in bayesNet:
        all_variables.update(factorTable.columns)
    all_variables.remove('probs')

    # for each variable in the net
    for variable in all_variables:
        copy_net =updated_net.copy()
        updated_net =[]
        factor =pd.DataFrame(columns=['probs'])

        # for each table in the net
        for factorTable in copy_net:
            if variable in factorTable.columns:
                factor =joinFactors(factor, factorTable)
            else:
                updated_net.append(factorTable)
        if variable in hiddenVar:
            # if variable is in hiddenVar, whitch means it should be marginalized
            factor =marginalizeFactor(factor, variable)
        updated_net.append(factor)

    # normalization
    norm_scale =sum(list(factor['probs']))
    factor['probs'] /=norm_scale
    return factor
```

II. USE BAYESIANNETWORKS TO BEHAVIORAL RISK FACTOR SURVEILLANCE SYSTEM SURVEY

Q1 Answer:

The size of networks can be computed by:

$$8+8\times2+8\times2+8\times2\times4+8\times2\times2\times4+8\times2\times2\times2+4\times4+4\times4\times2\times2+4\times4\times2\times2+4\times4\times2\times2=504$$

The total number of probabilities needed to store the full joint distribution is

$$8\times2\times2\times4\times4\times2\times2\times2\times2\times4=2^{15}=32768$$

Q2 Answer:

The probability of the outcome if I have bad habits or good habits and the probability of the outcome if I have poor health or good health are shown in Table I. The output of the code for question 2 are shown in Figure 1.

health outcomes		bad habits	good habits	pool health	good health
diabetes	1	15.05%	12.71%	11.54%	5.771%
	2	0.8965%	0.8865%	0.7662%	9.543%
	3	82.24%	84.77%	86.09%	92.22%
	4	1.810%	1.632%	1.604%	1.055%
stroke	1	4.926%	3.611%	8.269%	1.446%
	2	95.07%	96.39%	91.73%	98.55%
heart attack	1	7.433%	5.280%	14.08%	1.616%
	2	92.57%	94.72%	85.92%	98.38%
angina	1	8.045%	5.476%	16.16%	1.333%
	2	91.96%	94.52%	83.84%	98.67%

TABLE I: Results for Question 2.

```
with bad habits:
diabetes:
   smoke  exercise  diabetes     probs
0      1         2         1  0.150516
1      1         2         2  0.008965
2      1         2         3  0.822423
3      1         2         4  0.018096
stroke:
   smoke  exercise  stroke     probs
0      1         2       1  0.049264
1      1         2       2  0.950736
attack:
   smoke  exercise  attack    probs
0      1         2       1  0.07433
1      1         2       2  0.92567
angina:
   smoke  exercise  angina     probs
0      1         2       1  0.080448
1      1         2       2  0.919552
```

```
with good habits:
diabetes:
   smoke  exercise  diabetes     probs
0      2         1         1  0.127119
1      2         1         2  0.008865
2      2         1         3  0.847693
3      2         1         4  0.016323
stroke:
   smoke  exercise  stroke    probs
0      2         1       1  0.03611
1      2         1       2  0.96389
attack:
   smoke  exercise  attack     probs
0      2         1       1  0.052798
1      2         1       2  0.947202
angina:
   smoke  exercise  angina     probs
0      2         1       1  0.054755
1      2         1       2  0.945245
```

```
with pool health:
diabetes:
   bmi  diabetes     probs
0    3         1  0.115423
1    3         2  0.007662
2    3         3  0.860873
3    3         4  0.016043
stroke:
   cholesterol  bmi  bp  stroke     probs
0            1    3   1       1  0.082686
1            1    3   1       2  0.917314
attack:
   cholesterol  bmi  bp  attack     probs
0            1    3   1       1  0.140784
1            1    3   1       2  0.859216
angina:
   cholesterol  bmi  bp  angina     probs
0            1    3   1       1  0.161608
1            1    3   1       2  0.838392
```

```
with good health:
diabetes:
   bmi  diabetes     probs
0    2         1  0.057710
1    2         2  0.009543
2    2         3  0.922194
3    2         4  0.010553
stroke:
   cholesterol  bmi  bp  stroke    probs
0            2    2   3       1  0.01446
1            2    2   3       2  0.98554
attack:
   cholesterol  bmi  bp  attack     probs
0            2    2   3       1  0.016161
1            2    2   3       2  0.983839
angina:
   cholesterol  bmi  bp  angina     probs
0            2    2   3       1  0.013326
1            2    2   3       2  0.986674
```

FIG. 1: Results for Question 2.

Q3 Answer:
The effect that a person's income has on their probability of having one of the four health outcomes are evaluated and the comparations are shown in Figure 2. We can conclude that the higher a person's income status is ,the less probabilities to have diseases problem. But the people who earn 10000-15000$ seem to be more likely to have diseases problem compared to the people who earn less than 10000$.

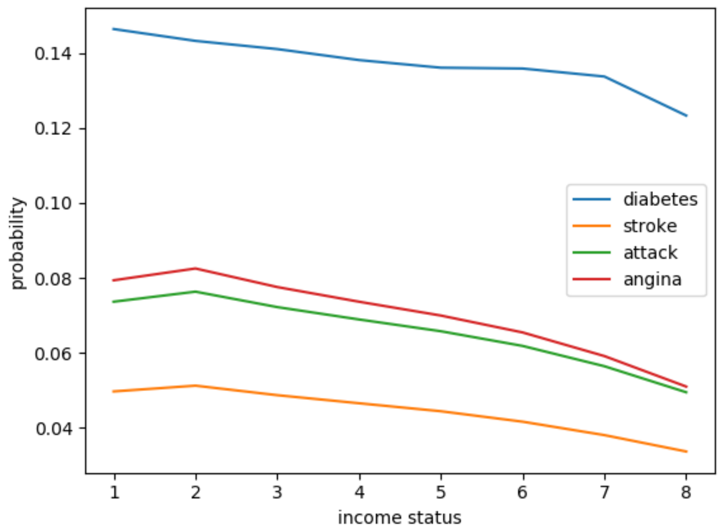

FIG. 2: Results for Question 3.

Q4 Answer:

As there are no links in the graph between the habits (smoking and exercise) and the outcomes, it can be infered that the effects of smoking and exercise on health problems are ignored, in other words, an assumption that smoking and exercise have no effects on health problems is made. Test for the validity of these assumptions is made by adding edges from smoking to each of the four outcomes and edges from exercise to each of the four outcomes.

The probability of the outcome if I have bad habits or good habits and the probability of the outcome if I have poor health or good health are shown in Table II. The output of the code for question 2 are shown in Figure 3.

health outcomes		bad habits	good habits	pool health	good health
diabetes	1	21.09%	9.855%	12.35%	5.417%
	2	6.915%	0.9884%	0.7460%	0.9731%
	3	76.07%	87.76%	85.24%	92.60%
	4	2.145%	1.399%	1.664%	1.014%
stroke	1	7.804%	2.421%	8.426%	1.400%
	2	92.20%	97.57%	91.57%	98.60%
heart attack	1	12.12%	3.102%	14.22%	1.547%
	2	87.88%	96.90%	85.78%	98.45%
angina	1	11.90%	3.68%	16.30%	1.294%
	2	88.10%	96.32%	83.70%	98.71%

TABLE II: Results for Question 4.

```
with bad habits:
diabetes:
   smoke  exercise  diabetes     probs
0      1         2         1  0.210939
1      1         2         2  0.006915
2      1         2         3  0.760698
3      1         2         4  0.021447
stroke:
   smoke  exercise  stroke     probs
0      1         2       1  0.078035
1      1         2       2  0.921965
attack:
   smoke  exercise  attack     probs
0      1         2       1  0.121166
1      1         2       2  0.878834
angina:
   smoke  exercise  angina     probs
0      1         2       1  0.119007
1      1         2       2  0.880993
```

```
with good habits:
diabetes:
   smoke  exercise  diabetes     probs
0      2         1         1  0.098552
1      2         1         2  0.009884
2      2         1         3  0.877576
3      2         1         4  0.013988
stroke:
   smoke  exercise  stroke     probs
0      2         1       1  0.024311
1      2         1       2  0.975689
attack:
   smoke  exercise  attack     probs
0      2         1       1  0.031015
1      2         1       2  0.968985
angina:
   smoke  exercise  angina   probs
0      2         1       1  0.0368
1      2         1       2  0.9632
```

```
with pool health:
diabetes:
   cholesterol  bp  bmi  diabetes     probs
0            1   1    3         1  0.123481
1            1   1    3         2  0.007460
2            1   1    3         3  0.852416
3            1   1    3         4  0.016643
stroke:
   cholesterol  bmi  bp  stroke     probs
0            1    3   1       1  0.084257
1            1    3   1       2  0.915743
attack:
   cholesterol  bmi  bp  attack     probs
0            1    3   1       1  0.142199
1            1    3   1       2  0.857801
angina:
   cholesterol  bmi  bp  angina     probs
0            1    3   1       1  0.162972
1            1    3   1       2  0.837028
```

```
with good health:
diabetes:
   cholesterol  bp  bmi  diabetes     probs
0            2   3    2         1  0.054173
1            2   3    2         2  0.009731
2            2   3    2         3  0.925952
3            2   3    2         4  0.010144
stroke:
   cholesterol  bmi  bp  stroke     probs
0            2    2   3       1  0.013997
1            2    2   3       2  0.986003
attack:
   cholesterol  bmi  bp  attack     probs
0            2    2   3       1  0.015469
1            2    2   3       2  0.984531
angina:
   cholesterol  bmi  bp  angina     probs
0            2    2   3       1  0.012944
1            2    2   3       2  0.987056
```

FIG. 3: Results for Question 4.

Compare Table II and Table I, we can find that habits have significant impacts on the health outcome, while health status do not. For habits and outcomes, we can find that with good habits the morbidity will decrease and with bad habits it will increase, which means that there are some dependences between habits and outcomes and the assumption of habits is not valid. But for the health status and outcomes, the assumption that health status and outcomes are independent is valid.

Q5 Answer:

As there are no edges between the four outcomes, it can be infered that the effects between four outcomes are ignored, in other words, an assumption that one outcome have no effects on others is made.

The output of the code for question 5 are shown in Figure 4.

```
T5
P(stroke = 1|diabetes = 1)
   stroke  diabetes     probs
0       1         1  0.044164
1       2         1  0.955836
P(stroke = 1|diabetes = 3)
   stroke  diabetes     probs
0       1         3  0.040478
1       2         3  0.959522

Add an edge from diabetes to stroke

P(stroke = 1|diabetes = 1)
   diabetes  stroke     probs
0         1       1  0.076198
1         1       2  0.923802
P(stroke = 1|diabetes = 3)
   diabetes  stroke     probs
0         3       1  0.035015
1         3       2  0.964985
```

FIG. 4: Results for Question 5.

From the results we can find that a person with diabetes problem is much more likely to face stroke problem, which means that the diabetes and stroke are correlative and the assumption that diabetes and stroke are independent is invalid.