Introduction to Intelligent Computing

Final Project Report

MCMC implementation

Group Members:
林文智
亞琳娜
蔡潔詩

Target problem domain

How does weather affect the emotions of people? How does the emotion of people affect the activities they participate in? And how does gender and age affect the types of activities?

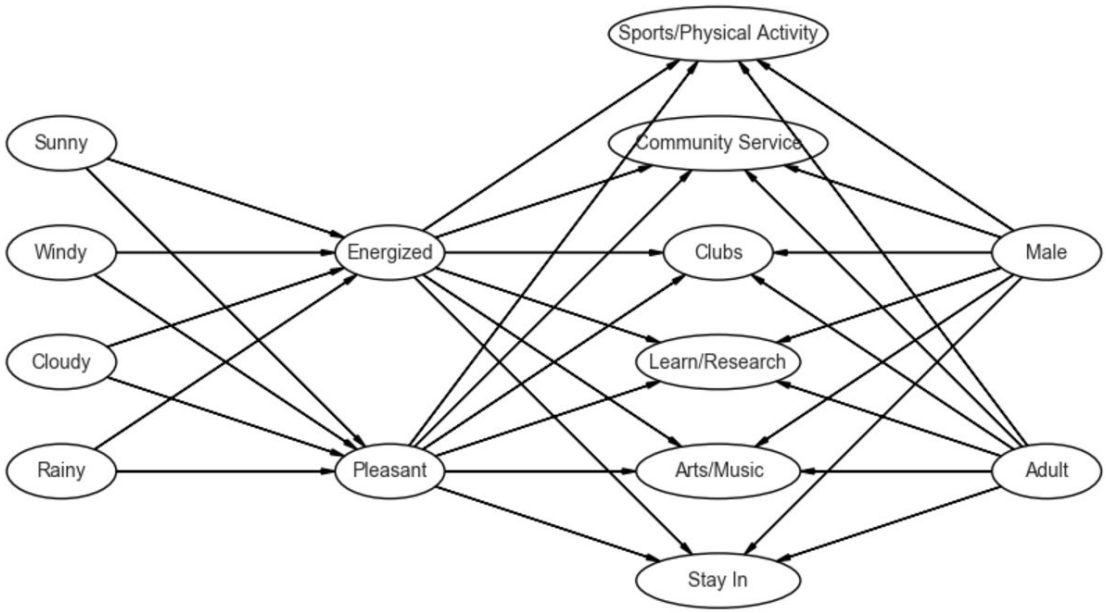


Figure : Belief network of our target problem domain with their causality links

We categorised the types of emotion into four, based on the PANAS model by Watson and Tellegen (1985), with examples of moods (in the circle) from Russell (1980) and Barrett & Russell (1999).

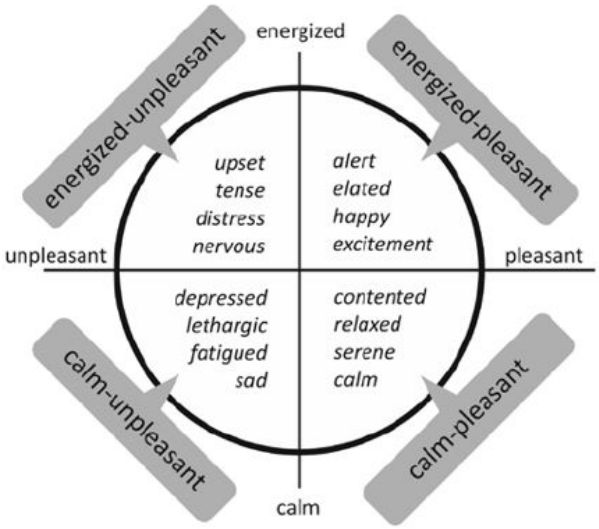


Figure : Categories for types of emotion

Data collection

Weather

For the weather CPT, we collected data from the Weather Channel, for the period of 2016 in Taipei. We decided to consider the whole year to avoid seasons to affect our probabilities.

The weather is defined as Rainy if the weather channel define it as rainy.
Windy is defined as reported winds of at least 24 km/h.
Weather channel define cloudy on a scale of 0 to 5. For our probabilities we consider cloudy anything equal or over 3, and sunny anything equal or less than 3.

	CST	Sunny	Windy	Rainy	Cloudy
0	1/1/2016	1	1	0	1
1	2/1/2016	0	0	1	1
2	3/1/2016	0	0	1	1
3	4/1/2016	0	0	1	1
4	5/1/2016	0	0	1	1
5	6/1/2016	0	0	1	1
6	7/1/2016	0	0	1	1
7	8/1/2016	0	0	1	1
8	9/1/2016	1	0	0	1
9	10/1/2016	0	0	0	1
10	11/1/2016	0	0	1	1
11	12/1/2016	0	0	1	1
12	13/1/2016	0	0	1	1
13	14/1/2016	0	0	1	1
14	15/1/2016	0	0	1	1
15	16/1/2016	0	0	1	1
16	17/1/2016	0	0	1	1
17	18/1/2016	0	0	1	1
18	19/1/2016	0	0	1	1
19	20/1/2016	0	1	1	1
20	21/1/2016	0	0	1	1
21	22/1/2016	0	0	1	1
22	23/1/2016	0	0	1	1
23	24/1/2016	0	0	1	1
24	25/1/2016	0	0	0	1
25	26/1/2016	1	0	0	1
26	27/1/2016	0	0	1	1
27	28/1/2016	0	0	1	1
28	29/1/2016	0	0	1	1

Figure : Part of the parsed weather data

We obtained the conditional probability table through the parsed weather data. For example, to get P(Sunny), we calculated the total number of “1”, divided by total number of days and P(¬Sunny) = 1 - P(Sunny).

Node:	Sunny
P(S)	P(~S)
0.2951807	0.7048193
Node:	Windy
P(W)	P(~W)
0.0813253	0.9186747
Node:	Cloudy
P(C)	P(~C)
0.8915663	0.1084337
Node:	Rainy
P(R)	P(~R)
0.6385542	0.3614458

Figure : CPT for parsed weather data

Age and Gender

We had collected our data by using Google survey and had a total of 252 respondents. We categorised the respondents age above 25 to be adults.

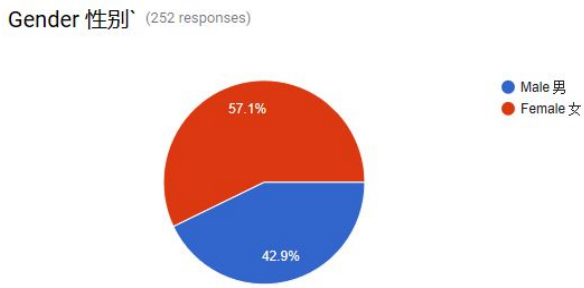


Figure : Pie chart of respondents gender distribution

Node:	Male
P(M)	P(~M)
0.571	0.429

Figure : CPT for respondents gender

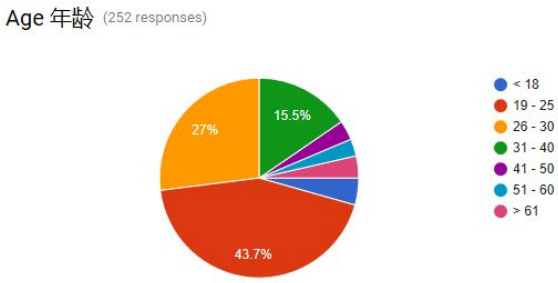


Figure : Pie chart of respondents age distribution

Node:	Adult
P(A)	P(~A)
0.519	0.481

Figure : CPT for respondents age

Emotion

We obtained data for the types of emotion based on weather by asking the questions “Which of the following best describe how you usually feel on a … day?” in our survey.

Figure : Legend for pie chart below

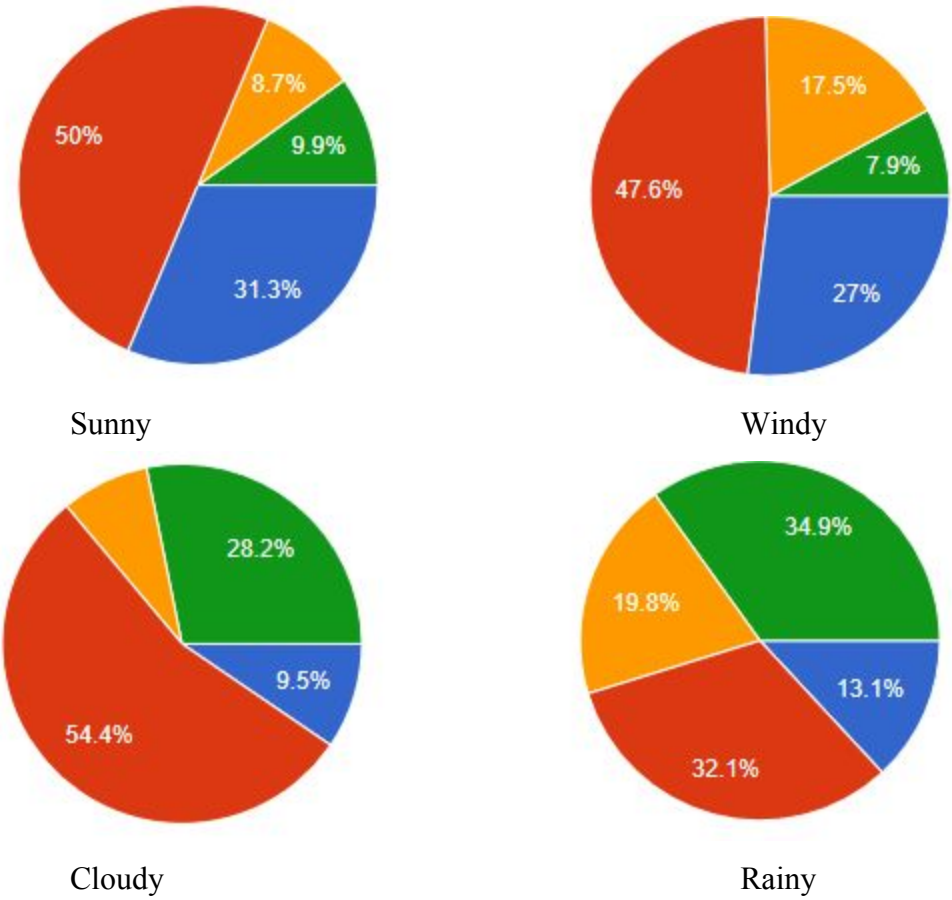


Figure : Pie chart for types of emotion based on weather

To obtain the conditional probability table for P(Energized) when P(Sunny)=True, P(Windy)=True, P(Cloudy)=True and P(Rainy)=True, we filter data by text contained “Alert / elated / happy / excitement” or “Upset / tense / distress / nervous” which fall under the category “Energized”.

Node:	Energized				
S	W	C	R	P(E)	P(~E)
T	T	T	T	0.007936507937	0.9920634921
T	T	T	F	0.0119047619	0.9880952381
T	T	F	T	0.02380952381	0.9761904762
T	T	F	F	0.09523809524	0.9047619048
T	F	T	T	0.0119047619	0.9880952381
T	F	T	F	0.03571428571	0.9642857143
T	F	F	T	0.05555555556	0.9444444444
T	F	F	F	0.1587301587	0.8412698413
F	T	T	T	0.02777777778	0.9722222222
F	T	T	F	0.02380952381	0.9761904762
F	T	F	T	0.09920634921	0.9007936508
F	T	F	F	0.1547619048	0.8452380952
F	F	T	T	0.03571428571	0.9642857143
F	F	T	F	0.01984126984	0.9801587302
F	F	F	T	0.06746031746	0.9325396825
F	F	F	F	0.1706349206	0.8293650794

Figure : CPT for Energized

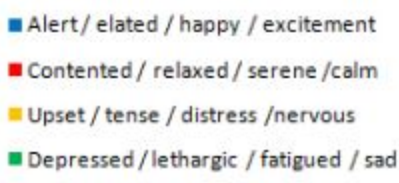
To obtain the conditional probability table for P(Pleasant) when P(Sunny)=True, P(Windy)=True, P(Cloudy)=True and P(Rainy)=True, we filter data by text contained “Alert / elated / happy / excitement” or “Contented / relaxed / serene / calm” which fall under the category “Pleasant”.

Node:	Pleasant				
S	W	C	R	P(P)	P(~P)
T	T	T	T	0.2301587302	0.7698412698
T	T	T	F	0.1428571429	0.8571428571
T	T	F	T	0.02777777778	0.9722222222
T	T	F	F	0.1865079365	0.8134920635
T	F	T	T	0.05158730159	0.9484126984
T	F	T	F	0.05952380952	0.9404761905
T	F	F	T	0.04761904762	0.9523809524
T	F	F	F	0.06746031746	0.9325396825
F	T	T	T	0.08333333333	0.9166666667
F	T	T	F	0.06349206349	0.9365079365
F	T	F	T	0.003968253968	0.996031746
F	T	F	F	0.007936507937	0.9920634921
F	F	T	T	0.007936507937	0.9920634921
F	F	T	F	0	1
F	F	F	T	0	1
F	F	F	F	0.01984126984	0.9801587302

Figure : CPT for Pleasant

Activities

We obtained data for the types of activities based on emotion by asking the questions “Which of the following activities would you prefer to participate when you are feeling … ?” in our survey.



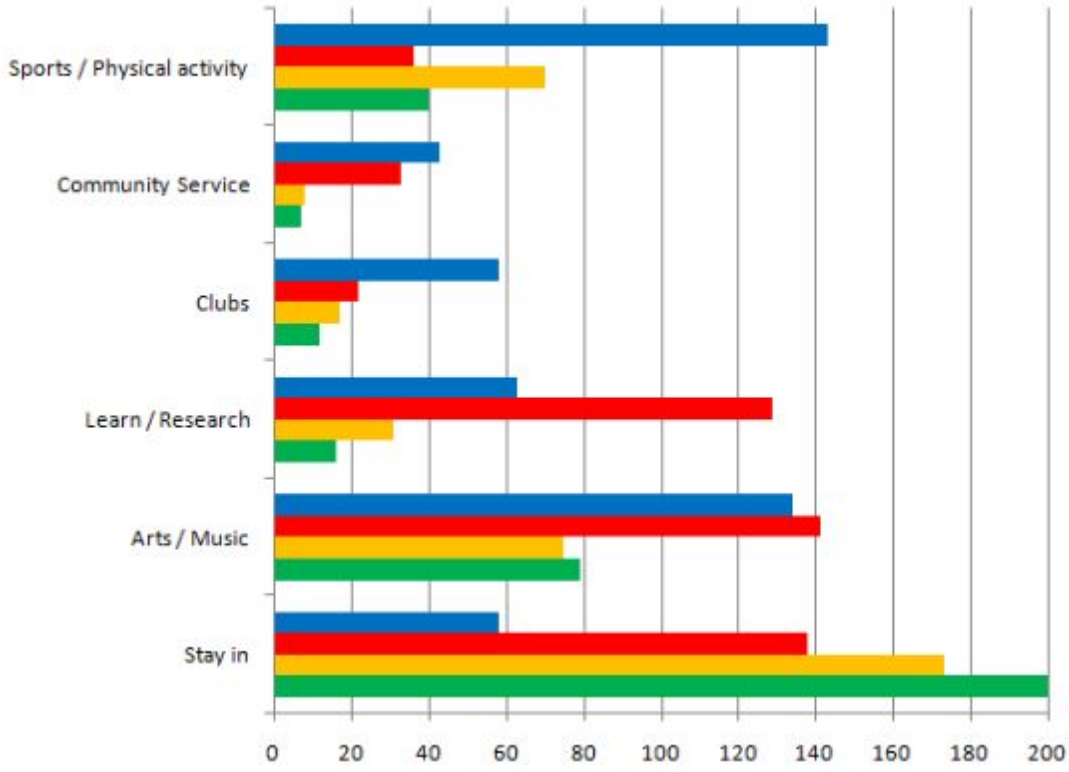


Figure : Bar chart for types of activities based on emotion

We calculated the probability for the types of activities based on emotion, age and gender of the respondents. For example, to obtain P(Clubs) when P(Male)=True, P(Adult)=True, P(Energized)=True and P(Pleasant)=True, we filter the respondents gender to be make, age range above 25, feeling “Alert / elated / happy / excitement”.

Node:	Sports				
M	A	E	P	P(SP)	P(~SP)
T	T	T	T	0.1443661972	0.8556338028
T	T	T	F	0.07394366197	0.926056338
T	T	F	T	0.04929577465	0.9507042254
T	T	F	F	0.04577464789	0.9542253521
T	F	T	T	0.09507042254	0.9049295775
T	F	T	F	0.05281690141	0.9471830986
T	F	F	T	0.03169014085	0.9683098592
T	F	F	F	0.03873239437	0.9612676056
F	T	T	T	0.1056338028	0.8943661972
F	T	T	F	0.04225352113	0.9577464789
F	T	F	T	0.0176056338	0.9823943662
F	T	F	F	0.02464788732	0.9753521127
F	F	T	T	0.1443661972	0.8556338028
F	F	T	F	0.07394366197	0.926056338
F	F	F	T	0.02816901408	0.9718309859
F	F	F	F	0.03169014085	0.9683098592

Figure : CPT for sports / physical activities

Node:	community				
M	A	E	P	P(CO)	P(~CO)
T	T	T	T	0.1063829787	0.8936170213
T	T	T	F	0.01063829787	0.9893617021
T	T	F	T	0.07446808511	0.9255319149
T	T	F	F	0.02127659574	0.9787234043
T	F	T	T	0.06382978723	0.9361702128
T	F	T	F	0.01063829787	0.9893617021
T	F	F	T	0.08510638298	0.914893617
T	F	F	F	0.01063829787	0.9893617021
F	T	T	T	0.1063829787	0.8936170213
F	T	T	F	0.04255319149	0.9574468085
F	T	F	T	0.07446808511	0.9255319149
F	T	F	F	0.01063829787	0.9893617021
F	F	T	T	0.2021276596	0.7978723404
F	F	T	F	0.02127659574	0.9787234043
F	F	F	T	0.1276595745	0.8723404255
F	F	F	F	0.03191489362	0.9680851064

Figure : CPT for community service

Node:	clubs				
M	A	E	P	P(CL)	P(~CL)
T	T	T	T	0.1214953271	0.8785046729
T	T	T	F	0.03738317757	0.9626168224
T	T	F	T	0.03738317757	0.9626168224
T	T	F	F	0.03738317757	0.9626168224
T	F	T	T	0.07476635514	0.9252336449
T	F	T	F	0.01869158879	0.9813084112
T	F	F	T	0.03738317757	0.9626168224
T	F	F	F	0.01869158879	0.9813084112
F	T	T	T	0.1121495327	0.8878504673
F	T	T	F	0.05607476636	0.9439252336
F	T	F	T	0.04672897196	0.953271028
F	T	F	F	0.01869158879	0.9813084112
F	F	T	T	0.214953271	0.785046729
F	F	T	F	0.04672897196	0.953271028
F	F	F	T	0.08411214953	0.9158878505
F	F	F	F	0.03738317757	0.9626168224

Figure : CPT for clubs

Node:	learn				
M	A	E	P	P(LE)	P(~LE)
T	T	T	T	0.09243697479	0.9075630252
T	T	T	F	0.03361344538	0.9663865546
T	T	F	T	0.1176470588	0.8823529412
T	T	F	F	0.02941176471	0.9705882353
T	F	T	T	0.05042016807	0.9495798319
T	F	T	F	0.02941176471	0.9705882353
T	F	F	T	0.1008403361	0.8991596639
T	F	F	F	0.01680672269	0.9831932773
F	T	T	T	0.07983193277	0.9201680672
F	T	T	F	0.03781512605	0.9621848739
F	T	F	T	0.1638655462	0.8361344538
F	T	F	F	0.01260504202	0.987394958
F	F	T	T	0.04201680672	0.9579831933
F	F	T	F	0.02941176471	0.9705882353
F	F	F	T	0.1554621849	0.8445378151
F	F	F	F	0.008403361345	0.9915966387

Figure : CPT for learn / research

Node:	Arts				
M	A	E	P	P(AR)	P(~AR)
T	T	T	T	0.06074766355	0.9392523364
T	T	T	F	0.04906542056	0.9509345794
T	T	F	T	0.08177570093	0.9182242991
T	T	F	F	0.03504672897	0.964953271
T	F	T	T	0.06074766355	0.9392523364
T	F	T	F	0.03504672897	0.964953271
T	F	F	T	0.06074766355	0.9392523364
T	F	F	F	0.03037383178	0.9696261682
F	T	T	T	0.1051401869	0.8948598131
F	T	T	F	0.04205607477	0.9579439252
F	T	F	T	0.09579439252	0.9042056075
F	T	F	F	0.04906542056	0.9509345794
F	F	T	T	0.08411214953	0.9158878505
F	F	T	F	0.04906542056	0.9509345794
F	F	F	T	0.09112149533	0.9088785047
F	F	F	F	0.07009345794	0.9299065421

Figure : CPT for arts / musics

Node:	Stay in				
M	A	E	P	P(ST)	P(~ST)
T	T	T	T	0.03339191564	0.9666080844
T	T	T	F	0.05799648506	0.9420035149
T	T	F	T	0.05799648506	0.9420035149
T	T	F	F	0.07732864675	0.9226713533
T	F	T	T	0.02460456942	0.9753954306
T	F	T	F	0.06678383128	0.9332161687
T	F	F	T	0.04920913884	0.9507908612
T	F	F	F	0.06854130053	0.9314586995
F	T	T	T	0.0158172232	0.9841827768
F	T	T	F	0.08787346221	0.9121265378
F	T	F	T	0.07029876977	0.9297012302
F	T	F	F	0.1054481547	0.8945518453
F	F	T	T	0.02811950791	0.9718804921
F	F	T	F	0.0913884007	0.9086115993
F	F	F	T	0.06502636204	0.934973638
F	F	F	F	0.1001757469	0.8998242531

Figure : CPT for stay in

Implementation

For implementation we decided to use the Gibbs algorithm based on the following pseudocode:

function GIBBS-ASK(X, **e**, bn, N) **returns** an estimate of $\mathbf{P}(X|\mathbf{e})$
  local variables: $\mathbf{N}$, a vector of counts for each value of X, initially zero
    $\mathbf{Z}$, the nonevidence variables in bn
    $\mathbf{z}$, the current state of variables $\mathbf{Z}$, initially random
  for $i = 1$ to N **do**
    choose Z_j in $\mathbf{Z}$ uniformly at random
    set the value of Z_j in $\mathbf{z}$ by sampling from $\mathbf{P}(Z_j|mb(Z_j))$
    $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where x is the value of X in $\mathbf{z}$
  return NORMALIZE($\mathbf{N}$)

As well as the following formula for the Markov Blanket calculation:

$$P(x'_j|mb(X_j)) = P(x'_j|parents(X_j))\Pi_{Z_\ell \in Children(X_j)}P(z_\ell|parents(Z_\ell))$$

As a programming language we used python and the complete code can be found in the following github repository.Here are some snippets of the implemented code:

```
def gibbs_ask(self,x,e,bn,N):
    #returns an estimate of P(x|e)
    #prepare graph
    result = [0,0]
    bnNew = copy.deepcopy(bn)
    mNodes = bnNew.fill_nodes(e)
    myX = bnNew.get_x_node(x)
    for i in range(N):
        shufList = [i for i in range(len(mNodes))]
        shuffle(shufList)
        for z in shufList:
            ##Set value of Zj in z by sampling from P(Zj|mb(Zj))
            zState = self.mb(mNodes[z],bnNew)
            bnNew.set_state(zState,mNodes[z])
            xState = self.mb(myX,bnNew)
            if xState:
                result[1] = result[1] + 1
            else:
                result[0] = result[0] + 1

    total = result[0] + result[1]
    normalizedFalse = result[0]/total
    normalizedTrue = result[1]/total
    return {True: normalizedTrue, False: normalizedFalse}
```

```
def mb(self,z,bn):
    #returns probability of Z in the markov blanket space in BN
    #P(x given mb(X)) = P(x given parents(X)) multiplied by the multiplication
    # of all children Z with formula P(Z given parents(Z))
    probParent = bn.get_prob(z)
    probChild =  bn.get_prob_children(z)
    return bn.def_new_state(probParent*probChild, z)
```

Results Analysis

We can see from the first tests realized on the Gibbs implementation, when increasing the number of samples *N* the precision of the result becomes more accurate.

```
print(GibbsImplementation.get_result('e',{'s':True,'w':False,'c':False,'r':False},g,100))
print(GibbsImplementation.get_result('e',{'s':True,'w':False,'c':False,'r':False},g,1000))
print(GibbsImplementation.get_result('e',{'s':True,'w':False,'c':False,'r':False},g,2000))
```

*probability of '**energized**' given **rainy**=False, **windy**=False, **cloudy**=False, **sunny**=True, is: {False: 0.867, True: 0.133}*
*probability of '**energized**' given **rainy**=False, **windy**=False, **cloudy**=False, **sunny**=True, is: {False: 0.8355, True: 0.1645}*
*probability of '**energized**' given **rainy**=False, **windy**=False, **cloudy**=False, **sunny**=True, is: {False: 0.84885, True: 0.15115}.*

Additionally, since the implementation is randomized algorithm the probabilities can slightly change when the algorithm is executed, we can see that the results change but the variation is not significant.

*probability of '**energized**' given **sunny**=True, **windy**=False, **rainy**=False, **cloudy**=False, is: {False: 0.89, True: 0.11}*
*probability of '**energized**' given **sunny**=True, **windy**=False, **rainy**=False, **cloudy**=False, is: {False: 0.9054, True: 0.0946}*
*probability of '**energized**' given **sunny**=True, **windy**=False, **rainy**=False, **cloudy**=False, is: {False: 0.8424, True: 0.1576}*

We can also see that adding more evidence gives us different probabilities and also more accurate.

```
print(GibbsImplementation.get_result('e',{'s':True,'w':False,'c':False,'r':False},g,2000))
print(GibbsImplementation.get_result('e',{'s':True,'w':False,'c':False,'r':False,'p':False},g,2000))
print(GibbsImplementation.get_result('e',{'s':True,'w':False,'c':False,'r':False,'p':True},g,2000))
```

probability of **'energized'** given **windy**=False, **sunny**=True, **rainy**=False, **cloudy**=False
is: {False: 0.9007, True: 0.0993}
probability of **'energized'** given **windy**=False, **sunny**=True, **pleasant**=False, **rainy**=False, **cloudy**=False
is: {False: 0.7899444444444444, True: 0.21005555555555555}
probability of **'energized'** given **windy**=False, **sunny**=True, **pleasant**=True, **rainy**=False, **cloudy**=False
is: {False: 0.9998333333333334, True: 0.00016666666666666666}

Also, by performing tests with different evidence we can find different results that go more accordingly with the Probability Tables obtained from the Survey.

```
print(GibbsImplementation.get_result('e',{'s':False,'w':False,'c':False,'r':True},g,3000))
print(GibbsImplementation.get_result('e',{'s':False,'w':True,'c':False,'r':False},g,3000))
print(GibbsImplementation.get_result('e',{'s':True,'w':False,'c':True,'r':False},g,3000))
print(GibbsImplementation.get_result('e',{'s':True,'w':True,'c':False,'r':False},g,3000))
```

probability of **'energized'** given **rainy**=True, **windy**=False, **sunny**=False, **cloudy**=False, is:
{False: 0.8802333333333333, True: 0.11976666666666666}
probability of **'energized'** given **rainy**=False, **windy**=True, **sunny**=False, **cloudy**=False, is:
{False: 0.8405, True: 0.1595}
probability of **'energized'** given **rainy**=False, **windy**=False, **sunny**=True, **cloudy**=True, is:
{False: 0.6992333333333334, True: 0.3007666666666667}
probability of **'energized'** given **rainy**=False, **windy**=True, **sunny**=True, **cloudy**=False, is:
{False: 0.9042333333333333, True: 0.09576666666666667}

We have concluded that since the Belief Bayesian Network's nodes are very dependent and interconnected more information is needed to obtain a better sampling on the data.