Mental Health in Tech Survey

Anmisha Reddy Ramidi
Department of Computer Science
University at Buffalo
anmishar@buffalo.edu

Sindhura Uppu
Department of Computer Science
University at Buffalo
suppu@buffalo.edu

Abstract

This project aims to measure and model the attitude towards mental health and frequency of mental health disorders in the tech workplace. The goal is to achieve this by generating a probabilistic graphical model or Bayesian Network to model the data answer queries. The dataset is obtained from here. Exact and Approximate inference algorithms have been applied on this network using which the queries have been answered.

1 Problem Domain

The dataset that is being modeled in this project belongs to the Kaggle dataset. The data is actually collected from a survey that is conducted on people working in the Tech workplace. The domain Mental Health is a very important topic which is being taken seriously by many corporations. This dataset is from a 2014 survey that measures attitudes towards mental health and frequency of mental health disorders in the tech workplace. It is said that better data leads to better health services. Surveys such as these help in determining how issues related to mental health in technical corporations effect the majority of population. The major goal of this project is to analyze the attitude of people working at the Tech places and help in improving the conditions.

1.1 Data Set

The data set is the collection of the answers the workers have answered during the Survey. The data set contains around 1260 rows and has 25 variables. The variables help in describing the attitude and features of how mental illness is treated at the tech workplace. The data set is used to determine a Bayesian network, where links are generated based on the relationship of the variables with each other. The data set was cleaned for better understanding and for getting better results when modeled. The following are the descriptions of each variables and what they represent.

Timestamp: Timestamp at which the survey was taken.

Age: Age of the person taking the survey.

Gender: Gender of the person taking the survey.

Country: Country to which the person belongs.

State: If you live in the United States, which state or territory do you live in?

Self_employed: Are you self-employed?

Family_history: Do you have a family history of mental illness?

Treatment: Have you sought treatment for a mental health condition?

Work_interfere: If you have a mental health condition, do you feel that it interferes with your work?

No_employees: How many employees does your company or organization have?

Remote_work: Do you work remotely (outside of an office) at least 50% of the time?

Tech_company: Is your employer primarily a tech company/organization?

Benefits: Does your employer provide mental health benefits?

Care_options: Do you know the options for mental health care your employer provides?

Wellness_program: Has your employer ever discussed mental health as part of an employee wellness program?

Seek_help: Does your employer provide resources to learn more about mental health issues and how to seek help?

Anonymity: Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources?

Leave: How easy is it for you to take medical leave for a mental health condition?

Mental_health_consequence: Do you think that discussing a mental health issue with your employer would have negative consequences?

Phys_health_consequence: Do you think that discussing a physical health issue with your employer would have negative sonsequences?

Coworkers: Would you be willing to discuss a mental health issue with your coworkers?

Supervisor: Would you be willing to discuss a mental health issue with your supervisor(s)?

Mental_health_interview: Would you bring up a mental health issue with a potential employer in an interview?

Phys_health_interview: Would you bring up a physical health issue with a potential employer in an interview?

Mental_vs_physical: Do you feel that your employer takes mental health as seriously as physical health?

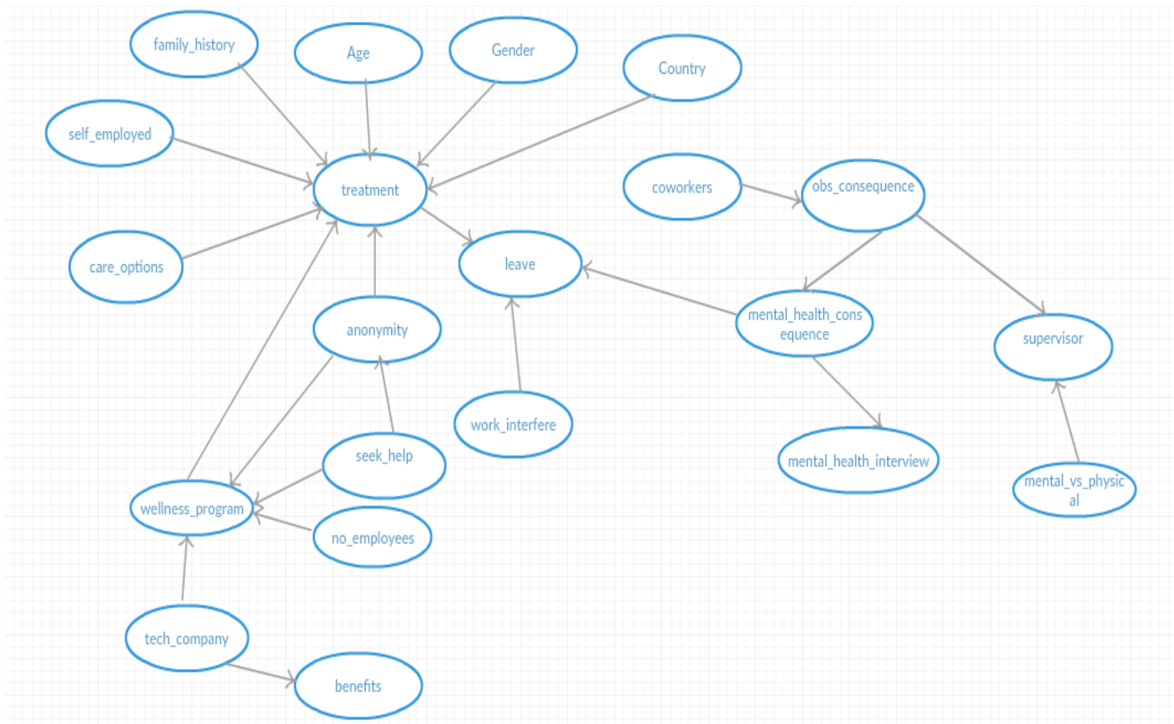
Obs_consequence: Have you heard of or observed negative consequences for coworkers with mental health conditions in your workplace?

Comments: Any additional notes or comments

1.2 Bayesian Network Model

A Bayesian Network represents the causal probabilistic relationship among a set of random variables, their conditional dependencies and it provides a compact representation of a joint probability distribution[1]. Edges represent conditional dependencies; nodes that are not connected represent variables that are conditionally independent of each other. Each node is associated with a probability function that take a particular set of values for the node's parent variables, and gives the probability of the variable represented by the node. They handle uncertainty through the established theory of probability.

The variables of the data set were assessed and relations were drawn. The Bayesian Network consists of 21 variables which contain over 22 links among them. The links represent the causal relationship between these 21 variables. The Bayesian Network that we represented was constructed manually based on intuition.



The Bayesian Network can be generated in Python using pgmpy library as follows:

```
Mental_health_model=
BayesianModel([('Age','treatment'),('Gender','treatment'),('Country','treatment'),('family_history','treatment'),('self_employed','treatment'),('care_options','treatment'),('anonymity','treatment'),('treatment','leave'),('work_interfere','leave'),('coworkers','obs_consequence'),('obs_consequence','supervisor'),('mental_vs_physical','supervisor'),('obs_consequence','mental_health_consequence'),('mental_health_consequence','mental_health_interview'),('anonymity','wellness_program'),('no_employees','wellness_program'),('seek_help','wellness_program'),('tech_company','wellness_program'),('tech_company','benefits'),('seek_help','anonymity'),('mental_health_consequence','leave'),('wellness_program','treatment')])
```

Pgmpy library has inbuilt models including Bayesian Network model. The nodes that have edges are sent as parameters using the node names as attributes in the model. The attribute in the left represents the parent while the one in the right is the child node in the Bayesian network.

1.3 Conditional Probability Distributions

This model is now fitted with the Mental Health dataset using Maximum Likelihood Estimator. Likelihood of a dataset is the probability of obtaining that particular set of data, given the chosen probability distribution model. There are unknown model parameters in this expression whose values that maximize the sample likelihood are known as the Maximum Likelihood Estimates.

Snippet to fit using Maximum Likelihood Estimator:

```
Mental_health_model.fit(train, estimator = MaximumLikelihoodEstimator)
```

Train is the training data of 600 samples that have been retrieved from the dataset.

Conditional Probability Distributions for each node were also generated from the Maximum Likelihood Estimator using the edges in the Bayesian network. A conditional probability distribution over b via a conditional distribution with a means that the distribution over b depends on the value of a. The CPD’s will be represented in a tabular format as follows:

```
+------------+-----+
| Age(20-40) | 0.5 |
+------------+-----+
| Age(40-60) | 0.5 |
+------------+-----+
```

Age is an independent (parent) node in the network.

```
+----------------------+----------------------+----------------+
| seek_help            | seek_help(Don't know) | seek_help(No)  |
+----------------------+----------------------+----------------+
| anonymity(Don't know) | 0.333333333333       | 0.333333333333 |
+----------------------+----------------------+----------------+
| anonymity(No)        | 0.333333333333       | 0.333333333333 |
+----------------------+----------------------+----------------+
| anonymity(Yes)       | 0.333333333333       | 0.333333333333 |
+----------------------+----------------------+----------------+
```

Seek_help is the parent node while anonymity is the child node and their CPD’s based on their dependency is obtained as shown above.

2 Inference Models

2.1 Belief Propagation

Computing the a posteriori belief of a variable in a general Bayesian Network is NP-hard. Belief Propagation is an Approximate Inference algorithm. It is an efficient way to solve inference problems based on passing local messages[2]. It is available in the pgmpy.inference library as a class for performing inference using Belief Propagation model. It creates a junction tree or Clique tree for the input probabilistic graphical model and performs calibration of the junction tree so formed using belief propagation.

Code snippet for Belief Propagation implementation[3]:

belief_prop = BeliefPropagation(Mental_health_model)

Now that the inference is drawn from this model, queries can be run on the model to analyze the variation and independence of one or more variables over other such variables in the model. A few of the queries implemented are as follows:

Query1:

bp1 = belief_prop.query(variables=['leave','wellness_program'],evidence={'tech_company' : 0})

print(bp1['leave'])

print(bp1['wellness_program'])

Here are looking at how easy is it for the employee to take leave and whether they are aware of the wellness programs in their company provided that it is a tech company they are working for.

Tech_company : 0 maps to the ‘Yes’ value for the variable in the dataset.

Result:

```
+---------+--------------+
| leave   |   phi(leave) |
|---------+--------------|
| leave_0 |       0.2756 |
| leave_1 |       0.0977 |
| leave_2 |       0.2311 |
| leave_3 |       0.0977 |
| leave_4 |       0.2978 |
+---------+--------------+
+--------------------+------------------------+
| wellness_program   |   phi(wellness_program) |
|--------------------+------------------------|
| wellness_program_0 |                 0.2800 |
| wellness_program_1 |                 0.4400 |
| wellness_program_2 |                 0.2800 |
+--------------------+------------------------+
```

Query 2:

```
bp2 = belief_prop.query(variables=['treatment'],evidence={'Age' : 1, 'Gender' : 1, 'family_history' : 1})
print(bp2['treatment'])
```

The attribute ‘Treatment’ is conditionally dependent on the attributes ‘Age’, ‘Gender’ and ‘family_history’. This query calculates how many males within the age range 20-40 and had a family history of mental illness have opted for mental health treatment.

Result:

```
+-------------+------------------+
| treatment   |   phi(treatment) |
|-------------+------------------|
| treatment_0 |           0.4432 |
| treatment_1 |           0.5568 |
+-------------+------------------+
```

Query 3:

```
bp3 = belief_prop.query(variables=['benefits','treatment'],evidence={'tech_company' : 1})
print(bp3['benefits'])
print(bp3['treatment'])
```

The above query shows how being in a tech company relates to whether their employer provides any benefits for mental health illness and how many have taken treatment from the organization. It should be noted that tech_treatment has an indirect dependence on tech_company while benefits has a direct dependence on tech_company in our Bayesian network.

Result:

```
+------------+-----------------+
| benefits   |   phi(benefits) |
|------------+-----------------|
| benefits_0 |          0.0000 |
| benefits_1 |          0.0000 |
| benefits_2 |          1.0000 |
+------------+-----------------+
```

```
+-------------+------------------+
| treatment   |   phi(treatment) |
|-------------+------------------|
| treatment_0 |           0.4441 |
| treatment_1 |           0.5559 |
+-------------+------------------+
```

Query 4:

bp7 = belief_prop.query(variables=['treatment','leave'],evidence={'seek_help' : 2,'care_options' : 1})

print(bp7['treatment'])

print(bp7['leave'])

This query identifies the relation between treatment, leave attributes independently while being conditionally independent on seek_help and care_options. A value 2 in seek_options means that the employer provides enough resources to learn more about health issues and a value 1 in care_options means that the employee knows the options for mental health that the employer provides.

Result:

```
+-------------+------------------+
| treatment   |   phi(treatment) |
|-------------+------------------|
| treatment_0 |           0.3933 |
| treatment_1 |           0.6067 |
+-------------+------------------+
+---------+--------------+
| leave   |   phi(leave) |
|---------+--------------|
| leave_0 |       0.2825 |
| leave_1 |       0.0884 |
| leave_2 |       0.2340 |
| leave_3 |       0.0884 |
| leave_4 |       0.3068 |
+---------+--------------+
```

Query 5:

bp5 = belief_prop.query(variables=['mental_health_interview','supervisor'],evidence={'obs_consequence' : 1})

print(bp5['mental_health_interview'])

print(bp5['supervisor'])

This query evaluates how many people are willing to disclose their mental health state to prospective employers in an interview and also to their supervisor in the present organization based on the consequences they observed for coworkers with mental health conditions in their workplace. A value 1 in ‘obs_consequence’ maps to ‘Yes’ in the dataset.

Result:

```
+---------------------------+--------------------------------+
| mental_health_interview   |   phi(mental_health_interview) |
|---------------------------+--------------------------------|
| mental_health_interview_0 |                         0.2000 |
| mental_health_interview_1 |                         0.8000 |
| mental_health_interview_2 |                         0.0000 |
+---------------------------+--------------------------------+
```

supervisor	phi(supervisor)
supervisor_0	0.0000
supervisor_1	0.2000
supervisor_2	0.8000

3 Sampling

We draw samples from the Bayesian network so that we can better understand the data and make statistical inferences on them.

3.1 Bayesian Model Sampling

Bayesian Model Sampling is available in pgmpy.sampling library as a class for sampling methods specific to Bayesian Models. The model to be given as an input to the Sampling function should be an instance of the Bayesian Model. Forward_sample function generates samples from joint distribution of the Bayesian network, which we are using here as shown in the below code snippet[4]:

```
infer1 = BayesianModelSampling(Mental_health_model)

evidence1 = [State('treatment',1)]

sample1 = infer1.forward_sample(evidence1,5)
```

sample1 now contains a sample of the size 5 (rows) that correspond to the value 1 in 'treatment'. This generates a data frame from the dataset randomly based on the given conditions.

Likelihood_weighted_sample generates weighted samples from joint distribution of the Bayesian network that comply with the given evidence.

Sample code snippet:

```
infer1 = BayesianModelSampling(Mental_health_model)

evidence2 = [State('treatment',1)]

sample2 = infer1.likelihood_weighted_sample(evidence2,5)
```

As in the forward sampling, the data from the Mental health survey dataset is selected randomly based on the evidence, although here a new column will be added to the data frame sample2, namely weights.

4 Evaluation Metrics

We developed inference algorithms so far for determining the mean and entropy of each distribution.

4.1 Mean of a distribution

The Mean for a distribution p(x) is:

$$E[p(\mathbf{x})] = \sum_{\mathbf{x}} \mathbf{x} p(\mathbf{x})$$

Using N samples, the mean can be computed as:

$$\hat{E}[p(\mathbf{x})] = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k$$

Numpy package in Python has predefined functions for computing mean.
Calculation of mean from the forward sample obtained above:

```
np.mean(sample1)
```

Result:

```
self_employed                 0.0
coworkers                     1.2
obs_consequence               0.0
Country                      12.0
care_options                  1.8
work_interfere                2.4
mental_vs_physical            1.0
Age                           1.2
Gender                        0.8
mental_health_consequence     1.0
mental_health_interview       1.0
seek_help                     1.2
anonymity                     1.2
supervisor                    1.8
family_history                1.0
no_employees                  2.6
tech_company                  0.4
wellness_program              0.4
treatment                     0.0
leave                         0.4
benefits                      2.0
dtype: float64
```

4.2 Entropy

The entropy of a distribution p(x) is:

$$H[p(\mathbf{x})] = -\sum_{\mathbf{x}} p(\mathbf{x}) \ln p(\mathbf{x})$$

When using N samples, the entropy can be calculated as:

$$\hat{H}[p(\mathbf{x})] = -\frac{1}{N}\sum_{k=1}^{N} \ln p(\mathbf{x}_k)$$

In order to calculate the entropy, the data frame containing the sample is converted into an array in Python and the probabilities are computed for each cell column-wise. The entropy is calculated using the entropy function in ‘Scipy’ package as follows:

```
scipy.stats.entropy(s1)
```

Result:

```
array([       -inf,  1.56071041,        -inf,  1.60943791,  1.58109375,
        1.58902692,  1.05492017,  1.56071041,  1.38629436,  1.60943791,
        1.60943791,  1.09861229,  1.09861229,  1.58109375,  1.60943791,
        1.51938266,  0.69314718,  0.69314718,        -inf,  0.        ,
        1.60943791])
```

References

[1] Dimitris Margaritis (2003) Learning Bayesian Network Model Structure from Data. Carnegie Mellon University, PA.

[2] Jonathan S. Yedidia, William T. Freeman, Yair Weiss (2001) Understanding Belief Propagation and its Generalizations. *Mitsubishi Electric Research Laboratories TR2001-22*

[3] http://pgmpy.org/inference.html

[4] http://pgmpy.org/sampling.html