Young People Survey using Probabilistic Graphical Model

Anand P. Popat
Department of Computer Science
University at Buffalo
Buffalo, NY 14226
apopat@buffalo.edu

Chirag A. Yeole
Department of Computer Science
University at Buffalo
Buffalo, NY 14226
chiragar@buffalo.edu

Abstract

We perform inference on a popular dataset on Kaggle, Young People Survey using Probabilistic Graphical Model. We modified the raw dataset to have 30 variables and as the variables are categorical, a variable can have multiple values, and so we designed a graphical model to perform inference on it. We have used two different methods of inference here namely variable elimination and belief propagation to answer some meaningful queries from the dataset. As the dataset talks about the hobbies and interests of people with different demographs, our model can be used to infer interesting facts related to demographics and gender differences. The inference from these methods is also compared to the common societal belief/trends as some of the questions relate to human nature and extensive research has been done in the field of psychological sciences to answer them.

1 Introduction

1.1 Problem domain

The basic element of answering questions is to have some knowledge about what caused the condition in the first place e.g. If a patient has cold, the first thing we need to know to diagnose the cold is whether the patient has been in cold atmosphere or in contact with somebody having cold. Based on that we can diagnose whether the patient has cold or not. Going to a more complex question, if we want to diagnose whether a patient has cancer, we need to know a lot of things (lets call them variables) before we reach the diagnosis. As the number of variables increase, the interaction between those variables increases in the power of 2^n. The number reaches into millions when the variables are more than 20. Unfortunately, most real life questions include a huge magnitude of variables and hence to find all the combinations of behavior is not possible.

Human brain deals with thousands of variables every second to make decisions which seem trivial in day to day life but are not. It is believed that we don't see reality as it is but the human brain filters the variables in order to process the information most of the times and hence we never get a complete picture of the reality. The same kind of concept is applied in this machine learning technique where we deal with many variables but only deal with the significant ones and develop relationship among variables to make what is called a bayesian network. Based on that, we develop a graphical model which represents the selected variables as nodes and associated with each node is its Conditional Probability Table (CPT) which is formed based on the relationship of that node with other nodes. This graphical model is known as Probabilistic Graphical Model. In this project, we are dealing with a database, Young People Survey, that has many variables, hence the need to use Probabilistic Graphical Model, to find the answers to some common traits of people.

2 Dataset

We chose this dataset as it is related to human interests and hobbies which for one is interesting to know and think about but at the same time has application in sales and advertising. For e.g., if we know that the user is a guy who is interested in sports, we can use adds showing him different sports equipment to buy. That is the way marketing is taking its course using social networking sites like Facebook and snapchat as their platform.

Our dataset, Young People Survey, is available on Kaggle. It is uploaded by Miroslav Sabo and the participants were of slovakian nationality. The dataset has then been converted to english. It has mainly 8 categories: Music Interests, which has 19 variables related to that, Movie Preferences, having 12 variables, Hobbies and Interests which has 32 variables in it, Phobias, Personality traits, Habits and Demographics having 10, 57, 10 and 10 variables in them respectively.

The dataset in total has 151 variables and the variables are of categorical type in which most of the variables will have values between 1-5. Other variables have more than 5 values. Clearly the dataset is too large to deal with. Even with Probabilistic Graphical Model, it can be difficult to deal with all the variables. Hence, we chose only the variables necessary for the evaluation of queries related to specific topics and normalized them to have 2 values instead. We kept 30 variables which are inevitable and enough for deriving inference. They are related to Gender, Education, Left - right handed, Movies, Mathematics, Music, Art, Sci-fi, Pop, Metal, Horror, Romantic, Loneliness, Internet, Politics, Alcohol, Happiness, Smoking, Empathy, Giving, Friends, Public Speaking, God and Religion.

3 Bayesian Network

First step after finalizing the database is to create a Bayesian Network. It is the essence of Probabilistic Graphical Model. A bayesian network is used to define causal relationship between the variables of the dataset. Variables are represented in the form of nodes and each such node is connected to other nodes via links which describe the causality between the nodes. In a relationship, there can be a parent node and a child node. This is important as it removes the need to evaluate the inter-relationship between all the variables. Depending on the network, the relationship is evaluated which reduces the computational time by a huge factor.

A bayesian network can be made using intuition or some algorithm like K2 or chi-square which measure the correlation between the variables to show the causality. If the topic to be dealt with is completely off the chart for the designer then these algorithms can be used to design a bayesian network. Other technique is to use intuition or knowledge to design a bayesian network. If we know the basic causal relationship between the variables of the dataset then this method can be used, it is also more practical to adopt this technique as the algorithms cannot be counted upon to give the optimal solution and even then some basic knowledge about the dataset is still expected.

As the variables in our dataset are fairly easy to relate to, we have used intuition and knowledge to develop our bayesian network. The network is shown in figure 1 below.

To make use of this bayesian network in Probabilistic Graphical Model, we need to develop the conditional probability of each node in the structure and the conditional probability of each node depends on its parents or children which can be known from the bayesian network. This is also called parameterized learning.

Even though we have developed a bayesian network, finding probability of nodes with many parents can be difficult. To further simplify it, we have used structured learning which made it easy to find independencies in the graph which in turn lowered the computation time. The four types of effects are:

1) x -> z -> y

2) y -> z -> x

3) x <- z -> y

4) x -> z <- y

In our graph, we had many indirect causal effect i.e. the first effect out of the four which means if z is present, the y is independent of x. This along with D-seperation are used to develop Conditional Probability Tables for the nodes and even in inference which is described in later sections.

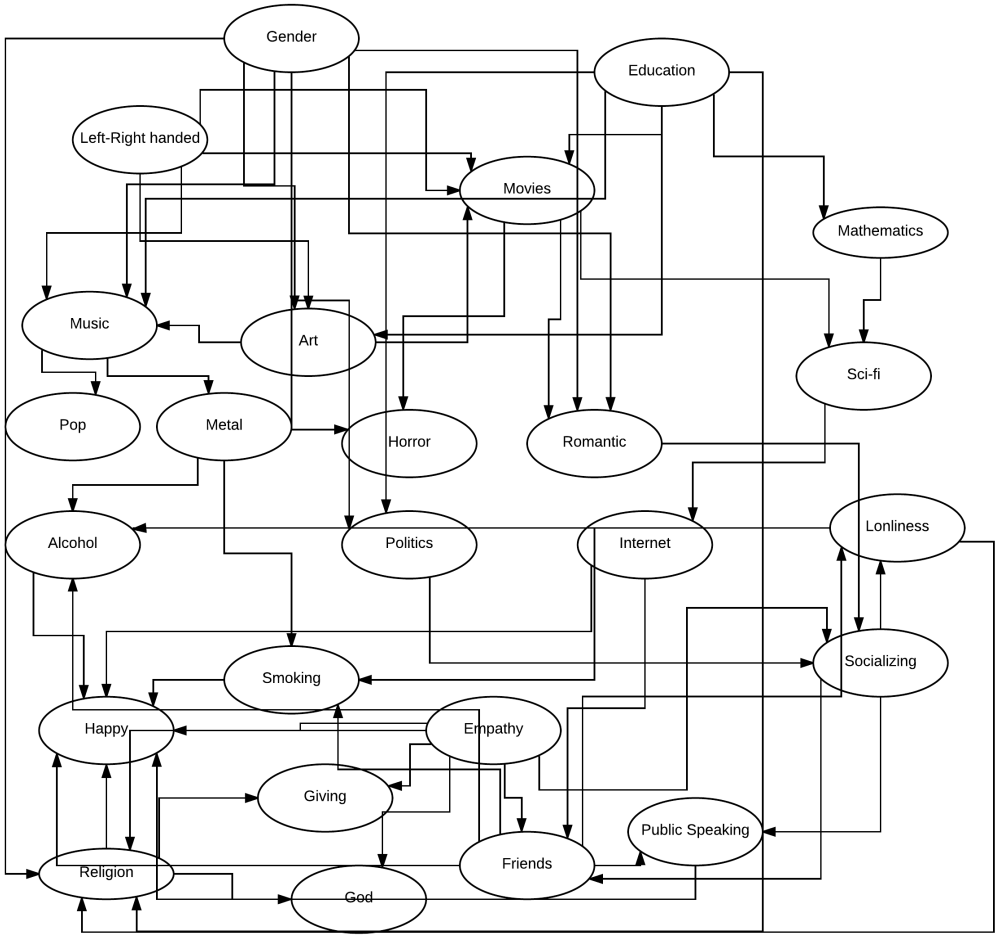

Figure 1: Bayesian Network

4 Probabilistic Graphical Model

A Probabilistic Graphical Model can be represented using two graphs namely bayesian network and markov random fields. For structured graph, we use bayesian network and for unstructured graph, markov random fields are generally used. As our dataset requires us to use a structured graph, we are using bayesian network with Tabular Conditional Probability Table for each node, using local independencies and D-seperation, as our Probabilistic Graphical Model to make inference. An example of using local independencies is below.

From the bayesian network, in figure 1, to calculate the CPT for Music node, we need to know the joint probability, P(Movies, Gender, Left - right handed, Education, Art). From the dependencies, we can calculate that as

$$P(M, G, L, E, A) = P(M)P(G)P(L)P(E)P(A/G, L, E)$$

where, M = Movies, G = Gender, L = Left -right handed, E = Education, A = Art.

Another example of local independency due to indirect causal effect is for node Sci-fi in figure 1,

```
P(S _|_ NonDescendents(S)/M,Ma)
```

where S = Sci - fi, M = Movies and Ma = Mathematics. Here Sci -fi is independent from all the non-descendents of Sci - fi (which are many) and it has Movies and Mathematics as parents. Note that Education is NOT a parent of Sci-fi as Mathematics is observed. So it creates indirect causal effect, x -> z -> y, hence Education can not influence Sci - fi due to the observance of Mathematics. This relationship along with other independency effects are observed through out the graph and due to them the computational time of the inference algorithms, described in the next section, reduces by a huge margin.

5 Inference

Inference is the technique to be used on the dataset to answer queries. Based on the graphical model we prepare, there are some inference methods that we can use for inference. They are divided into two categories namely, general inference and domain specific inference.

5.1 General Inference

In general inference, we derive the mean and entropy of the distribution using a sampling of dataset. Sampling are of many types such as Gibbs sampling, Ancestral sampling, Hamiltonian Monte Carlo sampling and No U-turn sampling. We have used Bayesian Model sampling to generate samples. Pgmpy is used for this and the code looks like this:

```
sample = BayesianModelSampling(bayesian_model)
sample.forward_sample(100)
```

The next step is to find the mean and entropy of the distributions for which we have used functions like `numpy.mean()`, `scipy.stats.entropy()`, `scipy.stats.norm().pdf()`.

The relative entropy of the two distributions is calculated using Kullback-Leibler divergence function of `scipy.stats.entropy()`.

5.2 Domain - Specific Inference

To answer queries specific to our dataset, we need to develop Conditional Probability Tables for each of the node in the bayesian network. Based on that we can use some algorithms for exact inference or approximate inference. Some algorithms are Variable Elimination, Markov Chain Monte Carlo, Variational, Belief Propagation and Max-Product Linear Programming method. For this project we are using Variable Elimination and Belief Propagation to infer the queries and compare the working and time complexity of the two algorithms.

5.3 Variable Elimination

Variable elimination is a type of exact inference in which as the name suggests, a variable is eliminated at each step of the algorithm. The variable to be eliminated has its product of factors summed-up before elimination and this process is repeated for all the variables. The time complexity of this algorithm is 2^O(k) where k is the factors of the variables. The space complexity is also the same, 2^O(k). As the time is exponential, it may take a while to produce the inference. Using pgmpy, Variable Elimination can be done by importing the package:

```
from pgmpy.inference import VariableElimination
```

and using the code below to subject the bayesian model to variable elimination:

```
infer = VariableElimination(bayesian_model)
```

How to make inference and query building using variable elimination is given in the next

section.

A better understanding of the time complexity of variable elimination can be achieved through the graph below.

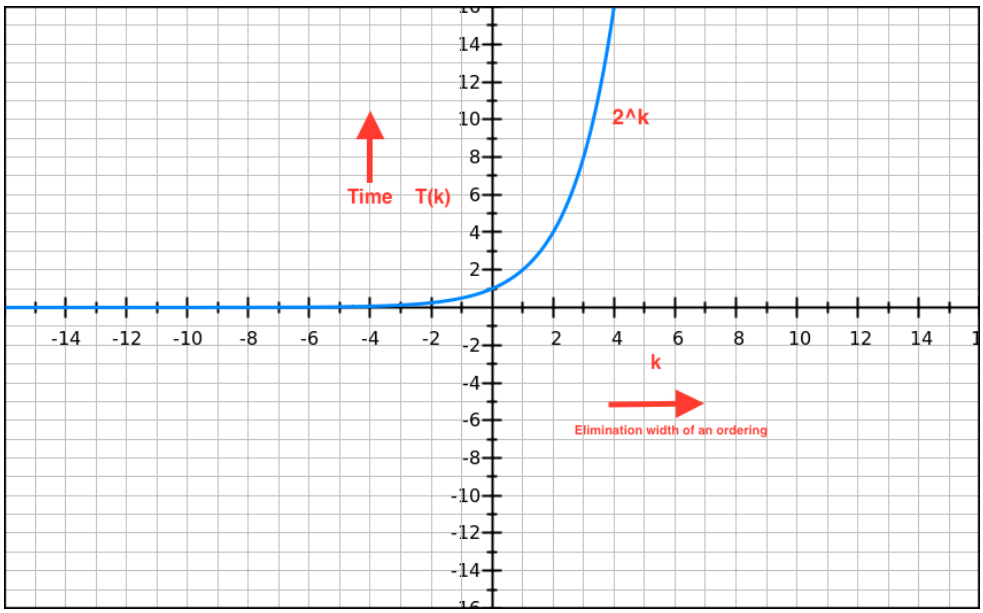

Figure 2: Time complexity of Variable Elimination

Suppose we have a graph below which represent all the factors of variable elimination.

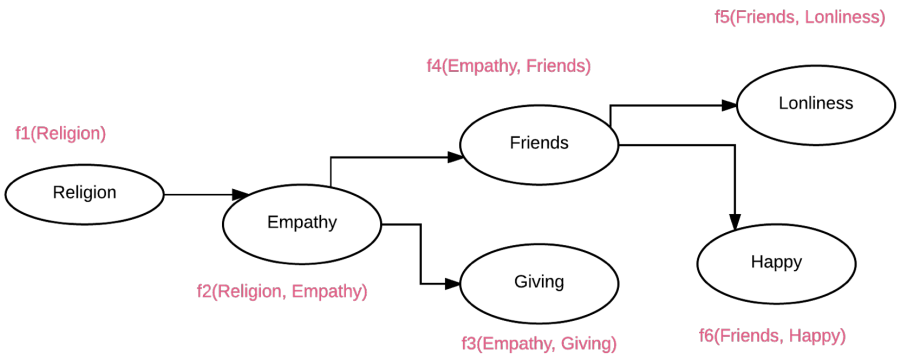

Figure 3: Variable Elimination Factors

To eliminate any variable, the algorithm forms a new factor containing all the edges of the variable that is eliminated. The same procedure is followed for all the variables.

From the graph,

Empathy: f2(Religion, Empathy), f4(Empathy, Friends), f3(Empathy, Giving)
Lonliness: f5(Friends, Lonliness)
Religion: f1(Religion)
Friends: f6(Friends, Happy)
Giving: (empty because it is already included in Empathy)

Happy: (empty because it is already included in Friends)
The events below will take place if we eliminate Empathy,

Empathy: eliminated
Lonliness: f5(Friends, Lonliness)
Religion: f1(Religion), f7(Religion, Friends, Giving)|
Friends: f6(Friends, Happy)
Giving:
Happy:

Now if we eliminate each variable one-by-one in this manner except Happy, in the end we will be left with f11(Happy).

So the factor k in the time complexity is the maximum number of factors during the elimination of each variable. So for all the variables, the time complexity will be 2^O(k).

5.4 Belief Propagation

Belief Propagation is a type of exact inference and it is similar to Variable Elimination in a way that it represents the graph in a tree form and the elimination process is interpreted as messages which contains the information of the number of factors created during the elimination process. We used pgmpy to use Belief Propagation inference by importing from

```
pgmpy.inference import BeliefPropagation
```

and using the code below to subject the bayesian model to Belief Propagation:

```
bp = BeliefPropagation(bayesian_model)
```

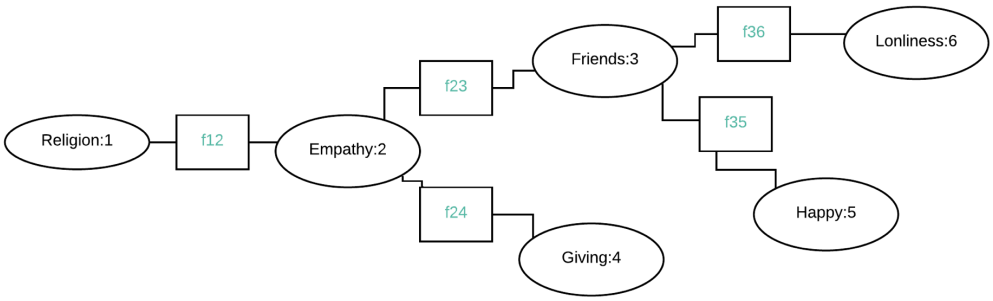

Figure 4: Message passing in Belief Propagation

The propagation starts at the bottom of the tree i.e the leaf nodes. When the leaf nodes collapse, they pass the message containing the information of all the integrating factors to the root node. All the nodes in the way also pass their messages. This procedure is continued for all the nodes in the tree.

The time complexity of Belief Propagation is O(n^2) as there is O(n) message passing at each iteration stage i.e for each node. So the total time for the algorithm to run for all the nodes is O(n^2).

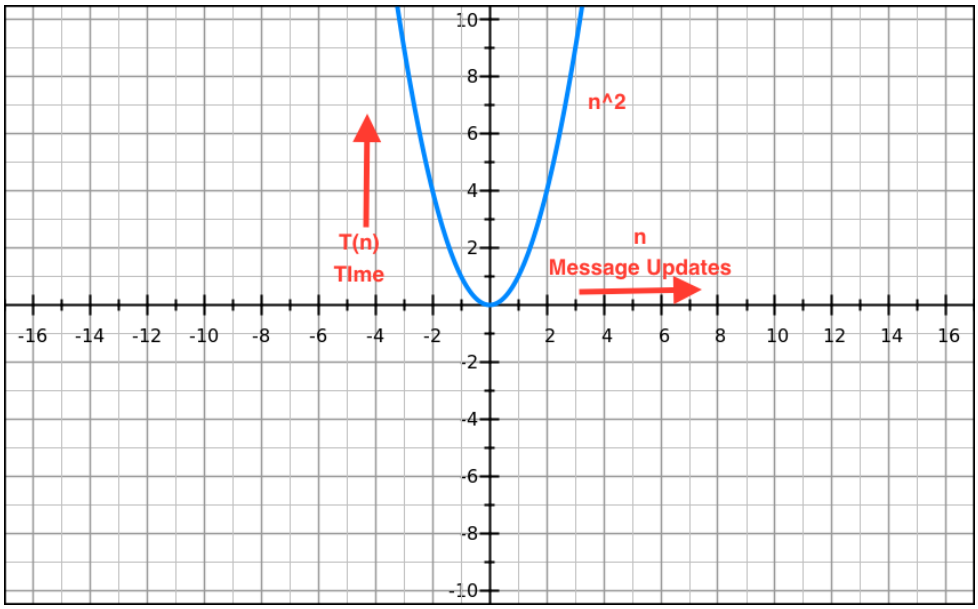

Figure 5: Time complexity of Belief Propagation

6 Conclusion

The main purpose of using Probabilistic Graphical Model is to reduce the computational time when the number of variables to deal with is too high. First, we had to clean the dataset, keep the variables that were needed, fill out the missing values. Second, after the dataset was ready, we developed a Bayesian Network, as we needed structured learning, to define relationship with the nodes. Third, we prepared Conditional Probability Tables for each node in the bayesian network keeping in mind the independencies related with the nodes. Based on that CPT, we performed inference using pgmpy and the algorithms defined. Some of the queries and their inferences are presented in the table below. Lot many queries of similar type can be answered based on this technique.

Queries & Inference

1. Do people who like socializing have more friends?
A. Sociable people have more number of friends.

```
variable_elimination.query(['Friends'],evidence={'Socializing': 1}) ['Friends']
```

Friends	phi(Friends)
Friends_0	0.0062
Friends_1	0.9938

```
belief_propagation.map_query(variables=['Friends'],evidence={'Socializing': 1})
```

{ '**Friends**' : 1 }

2. Which gender likes mathematics more?
A. Females like mathematics more than male.

```
variable_elimination.query(['Mathematics'],evidence={'Gender':
1}) ['Mathematics']
```

Mathematics	phi(Mathematics)
Mathematics_0	0.5790
Mathematics_1	0.4210

```
belief_propagation.map_query(variables=['Mathematics'],evidence=
{'Gender': 1})
```

{ '**Mathematics**' : 0 }

3. Do people who like metal songs and smoking tend to drink more?
Smoking_0: never
Smoking_1: tried smoking
Smoking_2: current smoker

A. People drink more if they like metal and smoking.
Alcohol_0: drink a lot
Alcohol_1: never
Alcohol_2: social drinker

```
variable_elimination.query(['Alcohol'],evidence={'Metal':
1,'Smoking': 2 }) ['Alcohol']
```

Alcohol	phi(Alcohol)
Alcohol_0	0.2655
Alcohol_1	0.1300
Alcohol_2	0.6045

```
belief_propagation.map_query(variables=['Alcohol'],evidence={'Me
tal': 1,'Smoking': current smoker})
```

{ '**Alcohol** : 2 }

4. Do people who are empathetic and generous have more friends?
A. People who are empathetic and generous have more number of friends.

```
variable_elimination.query(['Friends'],  evidence={'Empathy':
1,'Giving': 1}) ['Friends']
```

Friends	phi(Friends)
Friends_0	0.0062
Friends_1	0.9938

```
belief_propagation.map_query(variables=['Friends'],
evidence={'Empathy': 1,'Giving': 1})
```

{ '**Friends**' : 1 }

5. What is the higher level of education of female who like music, movies, art and don't like mathematics ?

A. Most female who like music, movies, art and don't like mathematics have secondary level education.
Education_0: college/bachelor degree
Education_1: doctorate degree
Education_2: masters degree
Education_3: currently a primary school pupil
Education_4: primary school
Education_5: secondary school

```
variable_elimination.query(['Education'], evidence={'Music': 1,'Movies': 1,'Art': 1,'Mathematics':0}) ['Education']
```

Education	phi(Education)
Education_0	0.1972
Education_1	0.0000
Education_2	0.0000
Education_3	0.0805
Education_4	0.0670
Education_5	0.6553

```
belief_propagation.map_query(variables=['Education'], evidence={'Music': 1,'Movies': 1,'Art': 1,'Mathematics':0})
```

{ '**Education**' : 5 }

Table 1: Queries and Inference table

References

[1] http://pgmpy.org

[2] https://github.com/pgmpy/pgmpy/tree/dev/examples

[3] http://stackoverflow.com/questions/37121515/pandas-how-to-group-and-unstack-on-multiple-variables

[4] http://www.hlt.utdallas.edu/~vgogate/ml/2012s/notes/BN_notes.pdf

[5] http://www.utdallas.edu/~nrr150130/cs6347/2016sp/lects/Lecture_5_Exact.pdf

[6] http://www.cs.toronto.edu/~hojjat/384f06/Lectures/Lecture17-4up.pdf

[7] http://zhenkewu.com/assets/pdfs/slides/teaching/2016/biostat830/lecture_notes/Lecture8.html#12

[8] https://en.wikipedia.org/wiki/Variable_elimination

[9] https://en.wikipedia.org/wiki/Belief_propagation

[10] https://ublearns.buffalo.edu/bbcswebdav/pid-3992648-dt-content-rid-14569921_1/courses/2171_19895/ProjectDescription.pdf

[11] https://ublearns.buffalo.edu/bbcswebdav/pid-4038139-dt-content-rid-14640103_1/courses/2171_19895/7.5-Reasoning%26D-Separation.pdf

[12] https://web.stanford.edu/~montanar/RESEARCH/BOOK/partD.pdf

[13] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.309.8086&rep=rep1&type=pdf