Bayesian Network for Collaborative Filtering of Movie Recommendations

Mihir Kulkarni
Computer Science and Engineering
University at Buffalo
Buffalo, NY-14214
UB Person number: 50168610
mihirdha@buffalo.edu

Pramod Rangaraju
Computer Science and Engineering
University at Buffalo
Buffalo, NY-14214
UB Person number: 50169514
prangara@buffalo.edu

***Abstract*— This project implements Bayesian Network by learning the Conditional Probability Distributions from the data. This Bayesian network is then used to answer queries that are in turn used for collaborative filtering. The project focusses on the queries and sampling which will help the system recommend movies for new users**

***Index Terms*—Bayesian Network, Probabilistic Graphical Models, Collaborative Filtering.**

I. Introduction

Collaborative filtering is a technique by which we can predict outcome of a particular query using the information about collaboration among multiple agents. Bayesian Networks can be applied to infer the outcome of collaborative filtering.

In this project we are applying Bayesian Network to predict the outcome of collaborative filtering. This is done by observing attributes of both the entities for which the collaboration is to be inferred.

For this project we have used R and Python for implementation. R is used to perform operations on the dataset to get the data in required format. Whereas Python is used to model the Bayesian Network and to perform inference and Maximum Likelihood Estimation queries as well as sampling.

In this project, for data analysis and data cleaning we have used pandas library of python which helps in rearranging the data faster. For learning parameters an d implementing queries, we have used libpgm library of python. This library mainly focusses on implementing Probabilistic Graphical Models in Python.

II. Dataset

For this project we are using the MovieLens dataset. This dataset is a dataset of 100k entries of users and their recommendations. Various attributes of users as well as the movies are observed. This dataset is 3 part dataset with 100k, 1M and 10M observations respectively.

The columns present in the data are-

1. User attributes- Gender, Age, and Occupation
2. Movie attributes- Movie title, Genre, Actor, Director
3. Movie Rating

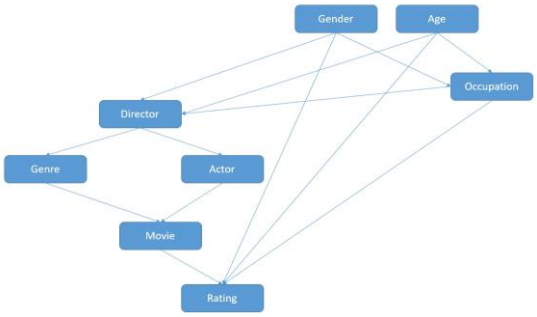
These attributes are divided over 3 different datasets. In this project we combine them into a single dataset containing all the values.

The dataset has all the discrete values except for the attribute Age- which is continuous. The attributes take 8-22 values.

This dataset is given in CSV format. This dataset needs to be combined into a single CSV file. This combined file is UnifiedMLData.csv. We have used this file to convert into JSON which is a required format to input in libpgm library.

III. Bayesian Network

For the dataset in question, we can draw a Bayesian network as follows. This network is not calculated by using the data, but drawn on intuition.



Users have attributes such as Age, Occupation and Gender. Movies have attribute Genre. Common attribute among the movies and users is the rating given by the user to the particular movie.

We have given this Bayesian Network to libpgm library in JSON format.

IV. Parameter Estimation

We have used Tabular conditional Probability Distributions in this project. These Conditional Probability Distributions are calculated using the data.

Parameter estimation is performed by using maximum likelihood estimation. In the context of this project, we are learning the Conditional probability distributions of the

Bayesian network using the most likely setting of the distribution, given the dataset.

Likelihood of parameter θ given data set: $L(\theta|D) = P(D|\theta)$, Where $L(\theta|D)$ is the likelihood of the parameter θ, given the dataset D.

V. Inference

An important application of Bayesian Network is to find the inference of from the given data. A simple form of exact inference can be easily found by summing over all the variables which are not part of the inference. This inference, although exact, is time consuming. This is because, the operation has exponential time complexity. To overcome this problem we have used various algorithms of exact and approximate inference.

A. Queries

Both directed and undirected graphical models represent a full joint probability distribution. Inferences are some of the main query types one might expect to answer with a joint distribution, and discuss the computational complexity of answering such queries using a PGMs.

1) Conditional Probability Query

The most common query type is the standard conditional probability query, P(Y | E = e). Such a query consists of two parts: the evidence, a subset E of random variables in the network, and an instantiation e to these variables; and the query, a subset Y of random variables in the network. Our task is to compute P(Y | E = e) = P(Y, e) / P (e), i.e., the probability distribution over the values y of Y, conditioned on the fact that E = e.

2) MAP Queries

Another type of query that often arises is that of finding the most probable assignment to some subset of variables. As with conditional probability queries, we have evidence E = e. In this case, however, we are trying to compute the most likely assignment to some subset of the remaining variables. This problem has two variants, where the first variant is an important special case of the second. The simplest variant of this task is the most probable explanation (MPE) queries. An MPE query tries to find the most likely assignment to all of the (non-evidence) variables. More precisely, if we let W = X − E, our task is to find the most likely assignment to the variables in W given the evidence E = e: $\text{argmax}_w P(w, e)$, where, in general, $\text{argmax}_x f(x)$ represents the value of x for which f(x) is maximal. Note that there might be more than one assignment that has the highest posterior probability. In this case, we can either decide that the MPE task is to return the set of possible assignments, or to return an arbitrary member of that set.

B. Algorithms

1) Variable Elimination

The Variable elimination (VE) is a simple and general exact inference algorithm in PGM. It can be used for inference of maximum a posteriori (MAP) state or estimation of marginal distribution over a subset of variables. The algorithm has exponential time complexity, but could be efficient in practice for the low-tree width graphs, if the proper elimination order is used.

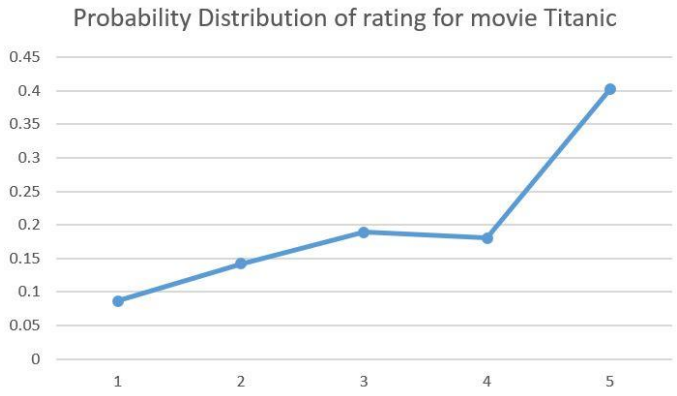
VI. Example Queries

In this project we have implemented queries which can give us an insight into the database. Some of the example queries which we ran on the Bayesian Network are as follows.

A. Rating of the movie given the Movie title.

This is particularly useful to find out the popularity of a movie among the users.

As an example we have calculated the probability distribution of rating given the movie is Titanic



We can see that Titanic being one of the most popular movies, most users have given rating 5.

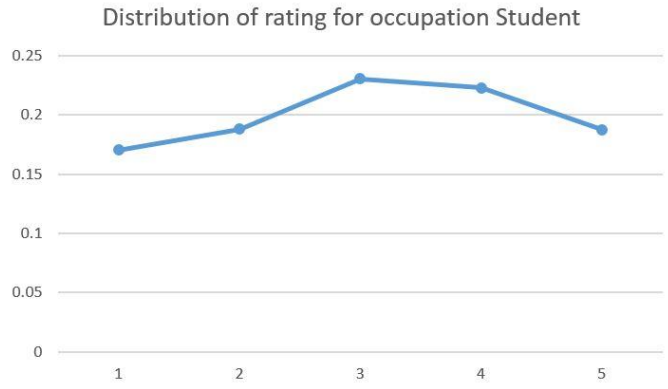
B. Rating of the Movie given Occupation of user

These queries are useful in understanding how people of a particular occupation perceive movies.

1) P(Rating=5|Occupation=student)

Answer is 0.187615695979

2) Distribution of Rating for occupation student



We can say that the students mostly give rating 3.

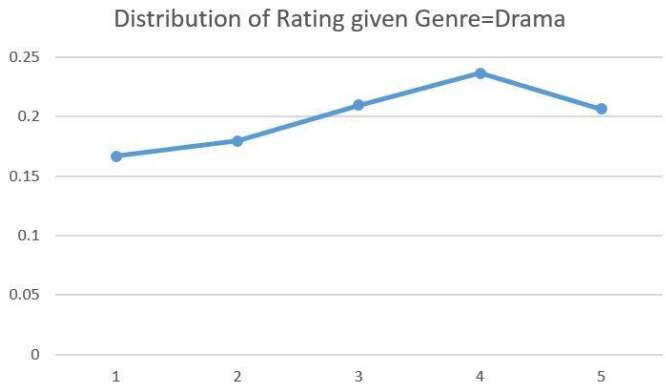
C. Rating given Genre

This query can be used to understand the ratings given to the movies of a particular Genre. Some genres are more popular among the users. This query can be used to understand the same.

1) P(Rating=4|Genre=multiple)
The answer to the query is 0.234122726279

2) P(Rating=1|Genre=multiple)
The answer to the query is 0.149013809903

3) Distribution of rating given Genre= drama



We can see from the distribution that most prominent rating given to the movies of Drama genre is 4. We can say that drama genre is sufficiently popular among the users.

D. Rating given Movie title and Occupation

In this query probability of the random variable Rating is found given two evidences namely- Movie title and Occupation

1) P(Rating=5|Occupation=student, Movie_title= Titanic)
Answer is 0.07549398151

This query can prove useful to find the popularity of a particular movie among the people of particular occupation.

This can also be used to recommend a particular movie given the occupation of the user.

VII. Sampling

Sampling if the process of drawing samples from the Bayesian Network and its conditional probability distributions. It is particularly useful to get that understanding of the data. Sampling is also used to find approximate inference where finding exact inference is not possible.

There are various method of sampling which are used for wide variety of purposes.

A. Random Sampling

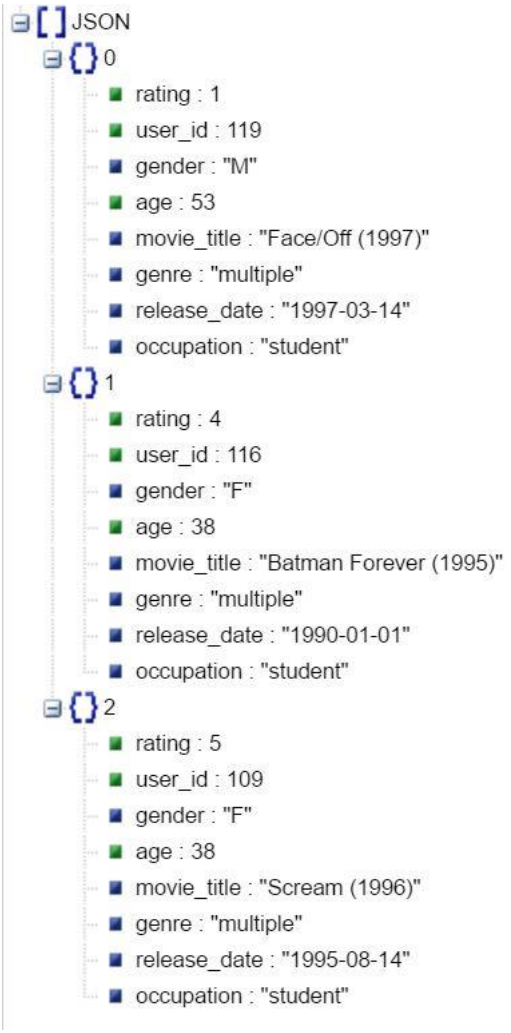
Subjects in the population are sampled by a random process, using either a random number generator or a random number table, so that each person remaining in the population has the same probability of being selected for the sample.

For our dataset the random sample is

B. Gibb's sampling

Gibb's sampling is a process of sampling where one random variable is changed in every iteration while others are kept at their generated values. The idea in Gibbs sampling is to generate posterior samples by sweeping through each variable (or block of variables) to sample from its conditional distribution with the remaining variables fixed to their current values.

For our project the Gibb's sampling generated is as follows. Please note that not all variables are kept at the previous values as not all values are in the distribution



VIII. SUMMARY

From this project we can see that we can use Bayesian Network to predict whether a particular user will like the movie or not. This information can then be used to recommend movies to the new as well as existing users of the system.

This can also be used to find out which are the most popular movies. We can also say what the gender of the user can be given that he/she liked the particular movie or not.

REFERENCES

[1] http://www.seas.upenn.edu/~taskar/pubs/gms-srl07.pdf section 2.3

[2] A. Becker and D. Geiger. A sufficiently fast algorithm for finding close to optimal clique trees. Artificial Intelligence, 125(1-2):3–17, 2001.

[3] W. Buntine. Chain graphs for learning. In Proceedings of the Conference on Uncertainty in Artificial Intelligence, 1995.

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

[5] http://www.itl.nist.gov/div898/handbook/apr/section4/apr412.htm

[6] https://en.wikipedia.org/wiki/Sampling_%28statistics%29

[7] http://www.ph.ucla.edu/epi/rapidsurveys/RScourse/RSbook_ch3.pdf

[8] http://www.uky.edu/~jmlhot2/courses/for480/Forest%20Sampling%20Forumla%20Sheet.pdf