Assignment 3

May 9, 2018

1 EECS 491: Probabilistic Graphical Models Assignment 3

David Fan
3/28/18

1.1 Problem Description

In this notebook I will attempt to create a larger graphical model (expanding on techniques used in the previous assignments) to model a specific domain. Using a dataset found on Kaggle, we will attempt to model the recruitment industry in India using a graphical model. In particular we will be exploring the effect of different factors on interview attendance.

1.2 Dataset

For this problem, we will explore a dataset found here. The author of the dataset describes the context of the dataset as follows:

> The data pertains to the recruitment industry in India for the years 2014-2016 and deals with candidate interview attendance for various clients ...
>
> The data have been collected by me and my fellow researchers over a period of over 2 years between September 2014 and January 2017.
>
> There are a set of questions that are asked by a recruiter while scheduling the candidate. The answers to these determine whether expected attendance is yes, no or uncertain.

```
In [170]: # Imported Packages
          import pandas as pd
          import numpy as np
          from dateutil import parser
```

```
In [171]: # Load dataset as a dataframe using pandas
          data = pd.read_csv('Interview.csv')
```

Let's take a quick look around the dataset so we can see what we're working with:

```
In [172]: data.head()
```

```
Out[172]:   Date of Interview Client name         Industry Location  \
          0        13.02.2015     Hospira  Pharmaceuticals  Chennai
          1        13.02.2015     Hospira  Pharmaceuticals  Chennai
          2        13.02.2015     Hospira  Pharmaceuticals  Chennai
          3        13.02.2015     Hospira  Pharmaceuticals  Chennai
          4        13.02.2015     Hospira  Pharmaceuticals  Chennai

            Position to be closed Nature of Skillset    Interview Type Name(Cand ID)  \
          0   Production- Sterile            Routine  Scheduled Walkin   Candidate 1
          1   Production- Sterile            Routine  Scheduled Walkin   Candidate 2
          2   Production- Sterile            Routine  Scheduled Walkin   Candidate 3
          3   Production- Sterile            Routine  Scheduled Walkin   Candidate 4
          4   Production- Sterile            Routine  Scheduled Walkin   Candidate 5

            Gender Candidate Current Location     ...      \
          0   Male                   Chennai     ...
          1   Male                   Chennai     ...
          2   Male                   Chennai     ...
          3   Male                   Chennai     ...
          4   Male                   Chennai     ...

            Are you clear with the venue details and the landmark.  \
          0                                              Yes
          1                                              Yes
          2                                              NaN
          3                                              Yes
          4                                              Yes

            Has the call letter been shared Expected Attendance Observed Attendance  \
          0                             Yes                Yes                  No
          1                             Yes                Yes                  No
          2                             NaN          Uncertain                  No
          3                             Yes          Uncertain                  No
          4                             Yes          Uncertain                  No

            Marital Status Unnamed: 23 Unnamed: 24 Unnamed: 25 Unnamed: 26 Unnamed: 27
          0         Single         NaN         NaN         NaN         NaN         NaN
          1         Single         NaN         NaN         NaN         NaN         NaN
          2         Single         NaN         NaN         NaN         NaN         NaN
          3         Single         NaN         NaN         NaN         NaN         NaN
          4        Married         NaN         NaN         NaN         NaN         NaN

          [5 rows x 28 columns]
```

Here we can see the different variables contained within the dataset. Of particular interest are: * **Date of interview:** This will be broken down into a month variable and a day of the week variable to explore if time of the year or day of the week has any effect on interview attendance. * **Industry:** To see if particular industries are more attractive than others resulting in a higher interview attendance rate. * **Location:** This appears to be candidate location. This value is equivalient to the variable 'Candidate Current Location'. Candidate location might have some effect on a candidate's ability to show up to an interview. * **Position to be closed:** The type of job the candidate is interviewing for. * **Nature of skillset:** The skills the candidate has (or claims to have). * **Interview Type:** Walkins/ Scheduled/ Scheduled walkins * **Gender** * **Candidate Job Location:** The location for the interview. * **Expected Attendance** * **Observed Attendance** * **Marital Status**

1.2.1 Formatting

Let's trim down the dataset to just what we need.

```
In [173]: data = data.dropna(axis=0, thresh=2)
```

```
In [174]: # Standardize Datestring format for parser
          for i in range(data["Date of Interview"].shape[0]):
              data.iloc[i,0] = data.iloc[i,0].replace(" -","-")
              data.iloc[i,0] = data.iloc[i,0].replace("","-")
              if data.iloc[i,0].find('&') is not -1:
                  data.iloc[i,0] = data.iloc[i,0][:data.iloc[i,0].find('&')]
```

```
In [175]: data.loc[:,"Date of Interview"] = data.loc[:, "Date of Interview"].apply(parser.pars
```

```
In [176]: to_trimmed = {
              "Month": data.loc[:,"Date of Interview"].apply(lambda x: x.month),
              "Day of the Week": data.loc[:,"Date of Interview"].apply(lambda x: x.weekday()),
              "Industry": data.loc[:, "Industry"],
              "Location": data.loc[:, "Location"],
              "Position to be closed": data.loc[:, "Position to be closed"],
              "Nature of Skillset": data.loc[:, "Nature of Skillset"],
              "Interview Type": data.loc[:, "Interview Type"],
              "Gender": data.loc[:, "Gender"],
              "Candidate Job Location": data.loc[:, "Candidate Job Location"],
              "Expected Attendance": data.loc[:, "Expected Attendance"],
              "Observed Attendance": data.loc[:, "Observed Attendance"],
              "Marital Status": data.loc[:, "Marital Status"]
          }
          trimmed = pd.DataFrame(to_trimmed)
```

```
In [177]: trimmed = trimmed.dropna(axis=0, how="any")
```

```
In [178]: trimmed.head()
```

```
Out[178]:   Candidate Job Location  Day of the Week Expected Attendance Gender  \
          0                  Hosur                4                 Yes   Male
          1              Bangalore                4                 Yes   Male
          2                Chennai                4           Uncertain   Male
          3                Chennai                4           Uncertain   Male
          4              Bangalore                4           Uncertain   Male
          Industry    Interview Type Location Marital Status  Month  \
0  Pharmaceuticals  Scheduled Walkin  Chennai         Single      2
1  Pharmaceuticals  Scheduled Walkin  Chennai         Single      2
2  Pharmaceuticals  Scheduled Walkin  Chennai         Single      2
3  Pharmaceuticals  Scheduled Walkin  Chennai         Single      2
4  Pharmaceuticals  Scheduled Walkin  Chennai        Married      2

  Nature of Skillset Observed Attendance Position to be closed
0            Routine                  No   Production- Sterile
1            Routine                  No   Production- Sterile
2            Routine                  No   Production- Sterile
3            Routine                  No   Production- Sterile
4            Routine                  No   Production- Sterile
```

We now have a trimmed down version of our dataset with only the variables of interest. Let's now define our model.

1.3 Model Definition

With some expert knowledge and hypotheses, we need to model these variables as a bayesian network. Our model shall be as follows:

We're going to be exploring using belief propagation and Monte Carlo sampling to infer different queries. Let's say that we were interested in learning about the probability that a married individual will show up to their interview in June so we want to infer the query:

$$P(ObservedAttendance \mid MaritalStatus = Married, Month = 6)$$

1.3.1 Belief Propagation

We're going to be using the pgmpy package to create our model and perform belief propagation. Pgmpy plays well with Pandas dataframes, so we can feed pgmpy our data and use the built-in Maximum Likelihood estimator to learn our conditional probability distributions.

```
In [179]: # Ignore this. This exists in case the MLE doesn't converge.
          # This function is needed to manually construct the CPDs in the model
          def calc_probability(variable):
              values = dict()
              for value in variable:
                  if value not in values:
                      values[value] = 1
                  else:
                      values[value] += 1
              total = variable.shape[0]
              for value in values:
                  values[value] /= total

              return values
```

Now we can build our pgmpy model:

```
In [180]: from pgmpy.models import BayesianModel
          from pgmpy.estimators import MaximumLikelihoodEstimator
          from pgmpy.inference import BeliefPropagation
```

```
In [181]: trimmed.columns
```

```
Out[181]: Index(['Candidate Job Location', 'Day of the Week', 'Expected Attendance',
                 'Gender', 'Industry', 'Interview Type', 'Location', 'Marital Status',
                 'Month', 'Nature of Skillset', 'Observed Attendance',
                 'Position to be closed'],
                dtype='object')
```

```
In [182]: model = BayesianModel([
              ("Industry", "Interview Type"),
              ('Candidate Job Location', "Interview Type"),
              ("Industry", "Location"),
              ("Candidate Job Location", "Location"),
              ("Day of the Week","Expected Attendance"),
              ("Day of the Week", "Observed Attendance"),
              ("Month", "Expected Attendance"),
              ("Month", "Observed Attendance"),
              ("Interview Type", "Expected Attendance"),
              ("Interview Type", "Observed Attendance"),
              ("Location", "Expected Attendance"),
              ("Location", "Observed Attendance"),
              ("Nature of Skillset", "Position to be closed"),
              ("Position to be closed", "Observed Attendance"),
              ("Position to be closed", "Expected Attendance"),
              ("Gender", "Marital Status"),
              ("Marital Status", "Observed Attendance"),
              ("Marital Status", "Expected Attendance")
          ])
```

Now we can estimate our CPDs. **WARNING: THIS WILL TAKE A LONG TIME TO RUN**

```
In [183]: model.fit(trimmed, estimator=MaximumLikelihoodEstimator)
```

```
In [189]: print(model.get_cpds()[0])
```

```
Candidate Job Location(- Cochin- )     0.00732899

Candidate Job Location(Bangalore)      0.20684

Candidate Job Location(Chennai)        0.727199

Candidate Job Location(Gurgaon)        0.0285016

Candidate Job Location(Hosur)          0.000814332
```

```
 Candidate Job Location(Noida)          0.012215

 Candidate Job Location(Visakapatinam)  0.017101
```

```
In [190]: print(model.get_cpds()[1])
```

```
 Day of the Week(0)  0.0350163

 Day of the Week(1)  0.192182

 Day of the Week(2)  0.185668

 Day of the Week(3)  0.324919

 Day of the Week(4)  0.108306

 Day of the Week(5)  0.12215

 Day of the Week(6)  0.031759
```

```
In [191]: print(model.get_cpds()[2])
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

```
In [192]: print(model.get_cpds()[3])
```

```
 Gender(Female)  0.218241

 Gender(Male)    0.781759
```

```
In [193]: print(model.get_cpds()[4])
```

```
Industry(BFSI)                    0.76873

Industry(Electronics)             0.0187296

Industry(IT)                      0.00895765

Industry(IT Products and Services) 0.036645

Industry(IT Services)             0.0187296

Industry(Pharmaceuticals)         0.134365

Industry(Telecom)                 0.0138436
```

```
In [196]: print(model.get_cpds()[5])
```

```
Candidate Job Location             Candidate Job Location(- Cochin- )  Candidate Job Location

Industry                           Industry(BFSI)                      Industry(Electronics)

Interview Type(Sceduled walkin)    0.16666666666666666                 0.16666666666666666

Interview Type(Scheduled )         0.16666666666666666                 0.16666666666666666

Interview Type(Scheduled Walk In)  0.16666666666666666                 0.16666666666666666

Interview Type(Scheduled Walkin)   0.16666666666666666                 0.16666666666666666

Interview Type(Walkin)             0.16666666666666666                 0.16666666666666666

Interview Type(Walkin )            0.16666666666666666                 0.16666666666666666
```

```
In [195]: print(model.get_cpds()[6])
```

```
Candidate Job Location  Candidate Job Location(- Cochin- )  Candidate Job Location(- Cochin-

Industry                Industry(BFSI)                      Industry(Electronics)

Location(- Cochin- )    0.09090909090909091                 0.09090909090909091

Location(Bangalore)     0.09090909090909091                 0.09090909090909091
Location(CHENNAI)       0.09090909090909091                  0.09090909090909091

Location(Chennai)       0.09090909090909091                  0.09090909090909091

Location(Delhi)         0.09090909090909091                  0.09090909090909091

Location(Gurgaon)       0.09090909090909091                  0.09090909090909091

Location(Gurgaonr)      0.09090909090909091                  0.09090909090909091

Location(Hyderabad)     0.09090909090909091                  0.09090909090909091

Location(Noida)         0.09090909090909091                  0.09090909090909091

Location(chennai)       0.09090909090909091                  0.09090909090909091

Location(chennai )      0.09090909090909091                  0.09090909090909091
```

```
In [197]: print(model.get_cpds()[7])
```

```
Gender                   Gender(Female)     Gender(Male)

Marital Status(Married)  0.582089552238806  0.321875

Marital Status(Single)   0.417910447761194  0.678125
```

```
In [198]: print(model.get_cpds()[8])
```

```
Month(1)   0.0781759

Month(2)   0.118893

Month(3)   0.0871336

Month(4)   0.236156

Month(5)   0.0936482

Month(6)   0.258958

Month(7)   0.0374593

Month(8)   0.0390879
Month(9)   0.0211726
Month(10)  0.00732899
Month(11)  0.0154723
Month(12)  0.00651466
```

```
In [199]: print(model.get_cpds()[9])
```

```
Nature of Skillset(- SAPBO, Informatica)                    0.00325733
Nature of Skillset(10.00 AM)                                0.000814332
Nature of Skillset(11.30 AM)                                0.00162866
Nature of Skillset(11.30 Am)                                0.000814332
Nature of Skillset(12.30 Pm)                                0.000814332
Nature of Skillset(9.00 Am)                                 0.000814332
Nature of Skillset(9.30 AM)                                 0.000814332
Nature of Skillset(ALS Testing)                             0.012215
Nature of Skillset(AML/KYC/CDD)                             0.0684039
Nature of Skillset(Accounting Operations)                   0.0700326
Nature of Skillset(Analytical R & D)                        0.0105863
Nature of Skillset(Analytical R&D)                          0.002443
Nature of Skillset(Automation Testing Java)                 0.00570033
Nature of Skillset(Banking Operations)                      0.0179153
Nature of Skillset(Banking operations)                      0.00162866
Nature of Skillset(BaseSAS Program/ Reporting)              0.000814332
Nature of Skillset(Biosimilars)                             0.000814332
```

Nature of Skillset(Biosimiliars)	0.00488599
Nature of Skillset(Biosimillar)	0.002443
Nature of Skillset(CDD KYC)	0.0423453
Nature of Skillset(COTS)	0.00325733
Nature of Skillset(COTS Developer)	0.0105863
Nature of Skillset(Core Java)	0.0138436
Nature of Skillset(Dot Net)	0.00732899
Nature of Skillset(EMEA)	0.00488599
Nature of Skillset(ETL)	0.00732899
Nature of Skillset(Fresher)	0.0700326
Nature of Skillset(Global Labelling)	0.00488599
Nature of Skillset(Hadoop)	0.00977199
Nature of Skillset(JAVA, J2ee)	0.000814332
Nature of Skillset(JAVA,J2ee)	0.000814332
Nature of Skillset(JAVA,SQL)	0.00407166
Nature of Skillset(JAVA/J2EE)	0.002443
Nature of Skillset(JAVA/J2EE/Struts/Hibernate)	0.179153
Nature of Skillset(JAVA/SPRING/HIBERNATE/JSF)	0.034202
Nature of Skillset(Java)	0.017101
Nature of Skillset(Java)	0.00814332
Nature of Skillset(Java ,J2ee)	0.00325733
Nature of Skillset(Java Developer)	0.0203583
Nature of Skillset(Java J2EE)	0.026873
Nature of Skillset(Java J2ee)	0.0130293

Nature of Skillset(Java JSF)	0.00325733
Nature of Skillset(Java Tech Lead)	0.00488599
Nature of Skillset(Java, J2Ee)	0.000814332
Nature of Skillset(Java, SQL)	0.002443
Nature of Skillset(Java, Spring, Hibernate)	0.00162866
Nature of Skillset(Java, XML, Struts, hibernate)	0.002443
Nature of Skillset(Java,J2EE)	0.00407166
Nature of Skillset(Java,J2ee, JSF)	0.00488599
Nature of Skillset(Java,SQL)	0.002443
Nature of Skillset(Java,spring,hibernate)	0.002443
Nature of Skillset(Java-SAS)	0.00407166
Nature of Skillset(Java/J2ee)	0.00570033
Nature of Skillset(Java/J2ee/Core Java)	0.00488599
Nature of Skillset(L & L)	0.00162866
Nature of Skillset(LCM -Manager)	0.00162866
Nature of Skillset(Lending & Liability)	0.00325733
Nature of Skillset(Lending And Liabilities)	0.002443
Nature of Skillset(Lending and Liabilities)	0.0179153
Nature of Skillset(Lending&Liablities)	0.00162866
Nature of Skillset(Licensing RA)	0.00325733
Nature of Skillset(Manager)	0.000814332
Nature of Skillset(Oracle)	0.0350163
Nature of Skillset(Oracle Plsql)	0.0203583
Nature of Skillset(Product Control)	0.00407166

```
Nature of Skillset(Production)                              0.000814332
Nature of Skillset(Production Support - SCCM)               0.00162866
Nature of Skillset(Publishing)                              0.00732899
Nature of Skillset(RA Label)                                0.00162866
Nature of Skillset(RA Publishing)                           0.00732899
Nature of Skillset(Regulatory)                              0.00977199
Nature of Skillset(Routine)                                 0.0382736
Nature of Skillset(SAS)                                     0.0179153
Nature of Skillset(SCCM)                                    0.0114007
Nature of Skillset(SCCM  SQL)                              0.000814332
Nature of Skillset(SCCM  Sharepoint)                       0.000814332
Nature of Skillset(SCCM-(Network, sharepoint,ms exchange))  0.000814332
Nature of Skillset(SCCm- Desktop support)                   0.00325733
Nature of Skillset(Sccm- networking)                        0.000814332
Nature of Skillset(Senior Analyst)                          0.00407166
Nature of Skillset(Senior software engineer-Mednet)         0.012215
Nature of Skillset(Sr Automation Testing)                   0.0105863
Nature of Skillset(Submission Management)                   0.00162866
Nature of Skillset(T-24 developer)                          0.012215
Nature of Skillset(TL)                                      0.002443
Nature of Skillset(Tech Lead- Mednet)                       0.000814332
Nature of Skillset(Tech lead-Mednet)                        0.00651466
Nature of Skillset(Technical Lead)                          0.000814332
Nature of Skillset(generic drugs  RA)                      0.00325733
```

```
 Nature of Skillset(production)                              0.00570033

 Nature of Skillset(sccm)                                    0.000814332

 Nature of Skillset(testing)                                 0.00895765
```

```
In [200]: print(model.get_cpds()[10])
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

```
In [201]: print(model.get_cpds()[11])
```

```
 Nature of Skillset                            Nature of Skillset(- SAPBO, Informatica)  Nature o

 Position to be closed(AML)                    0.0                                       0.0

 Position to be closed(Dot Net)                0.0                                       0.0

 Position to be closed(Niche)                  1.0                                       0.0

 Position to be closed(Production- Sterile)    0.0                                       0.0

 Position to be closed(Routine)                0.0                                       1.0

 Position to be closed(Selenium testing)       0.0                                       0.0

 Position to be closed(Trade Finance)          0.0                                       0.0
```

Now we can use pgmpy to perform belief propagation:

```
In [210]: belief_propagation = BeliefPropagation(model)
          belief_propagation.query(variables=['Observed Attendance'], evidence={'Marital Status
```

```
        ---------------------------------------------------------------------------
```

```
    TypeError                                 Traceback (most recent call last)

    <ipython-input-210-ec142ebf7bc5> in <module>()
      1 belief_propagation = BeliefPropagation(model)
----> 2 belief_propagation.query(variables=['Observed Attendance'], evidence={'Marital Sta


    ~/anaconda3/envs/491/lib/python3.6/site-packages/pgmpy/inference/ExactInference.py in q
    655         ...                           evidence={'A': 0, 'R': 0, 'G': 0, 'L': 1})
    656         """
--> 657         return self._query(variables=variables, operation='marginalize', evidence=e
    658
    659     def map_query(self, variables=None, evidence=None):


    ~/anaconda3/envs/491/lib/python3.6/site-packages/pgmpy/inference/ExactInference.py in _
    613         variable_elimination = VariableElimination(subtree)
    614         if operation == 'marginalize':
--> 615             return variable_elimination.query(variables=variables, evidence=evidenc
    616         elif operation == 'maximize':
    617             return variable_elimination.map_query(variables=variables, evidence=evi


    ~/anaconda3/envs/491/lib/python3.6/site-packages/pgmpy/inference/ExactInference.py in q
    125         """
    126         return self._variable_elimination(variables, 'marginalize',
--> 127                                           evidence=evidence, elimination_order=eli
    128
    129     def max_marginal(self, variables=None, evidence=None, elimination_order=None):


    ~/anaconda3/envs/491/lib/python3.6/site-packages/pgmpy/utils/state_name.py in __call__
    165
    166         if not method_self.state_names:
--> 167             return f(*args, **kwargs)
    168         else:
    169             self.state_names = method_self.state_names


    ~/anaconda3/envs/491/lib/python3.6/site-packages/pgmpy/inference/ExactInference.py in _
     54             for evidence_var in evidence:
     55                 for factor in working_factors[evidence_var]:
---> 56                     factor_reduced = factor.reduce([(evidence_var, evidence[evidenc
     57                     for var in factor_reduced.scope():
     58                         working_factors[var].remove(factor)
    ~/anaconda3/envs/491/lib/python3.6/site-packages/pgmpy/utils/state_name.py in __call__
    165 
    166         if not method_self.state_names:
--> 167             return f(*args, **kwargs)
    168         else:
    169             self.state_names = method_self.state_names


    ~/anaconda3/envs/491/lib/python3.6/site-packages/pgmpy/factors/discrete/DiscreteFactor
    416         if (any(isinstance(value, six.string_types) for value in values) or
    417                 not all(isinstance(state, (int, np.integer)) for var, state in valu
--> 418             raise TypeError("values: must contain tuples or array-like elements of
    419                             "(hashable object, type int)")
    420 


    TypeError: values: must contain tuples or array-like elements of the form (hashable obj
```

There appears to be a bug with pgmpy as how state-names are represented internally. Essentially, it doesn't appear to handle non-binary data all that well...