Modelling of functional features of brain using Bayesian Network

Ronak Panchal
Department of Computer Science
University at Buffalo
Person #:50205775
ronakhem@buffalo.edu

Jayaram Kuchibhotla
Department of Computer Science
University at Buffalo
Person #:50208766
jayaramk@buffalo.edu

Abstract

Human Brain has many named functional regions. Each region is responsible for a certain part of cognitive process. FMRI scan helps to know about the activation of these regions. Multiple interactions occur between the regions that depend upon the cognitive process. This paper conveys about implementation of Bayesian Network to model the above said interactions.

1 Problem Statement

The problem statement belongs to cognitive neuroscience domain. The dataset that we have chosen for this project belongs to OpenfMRI community. This dataset consists 176 fMRI images of a single patient who has been diagnosed with ADHD. These images are scanned at equal intervals of time. We have used Nilearn package in Python to convert fMRI images to 2D numerical array. This array has 176 rows and 39 columns corresponding to the BOLD (blood oxygen level dependent) signal value in 39 functional regions of brain. The BOLD signal is floating point value that can be either positive or negative. Linear Gaussian Bayesian Network has to be created using these continuous data values.

2 Learning Structure of Bayesian Network

We have not developed the Bayesian network based on intuition or prior knowledge of domain.
To learn model structure (a DAG) from a data set, there are two techniques:

- score-based structure learning
- constraint-based structure learning

2.1 PC Algorithm

Constraint-based structure learning has the following features:

- Identify independencies in the data set using hypothesis tests
- Construct DAG (pattern) according to identified independencies

We found that constraint based approach is more suitable for our problem statement as all of the variables involved in our problem statement are of continuous nature. One of the most widely used constraint based algorithm is PC algorithm. We have used PC algorithm as a tool to build the structure of our Bayesian network

PC algorithm first finds the skeleton (underlying undirected graph) and on a posterior step makes the orientation of the edges. It tests pairwise (conditional) independence among all pairs of variables, i.e., for each edge in the graph. For every pair of variables where the test concludes independence, the edge is removed and the conditioning set for which the independence holds is recorded. Edges are typically tested for independence more than once.

The independence procedure is run with an iteratively increasing size of the conditioning variable set. Conditional independence between two variables is tested for every possible conditioning set. These sets are constructed from the set of adjacent variables of the two variables under consideration

Having discovered the skeleton of the Bayesian network, the next phase of the PC algorithm orients the edges. First, it searches for v-structures. V-structures are formed between triplets of nodes where we have edges between nodes A and B, and nodes C and B. A v-structure is created by orienting the edges so that we have an arc from A to B and an arc from C to B, if there exists a conditioning set S for which A is (conditionally) independent of B and C is not present in S, hereby forming a V-shape.

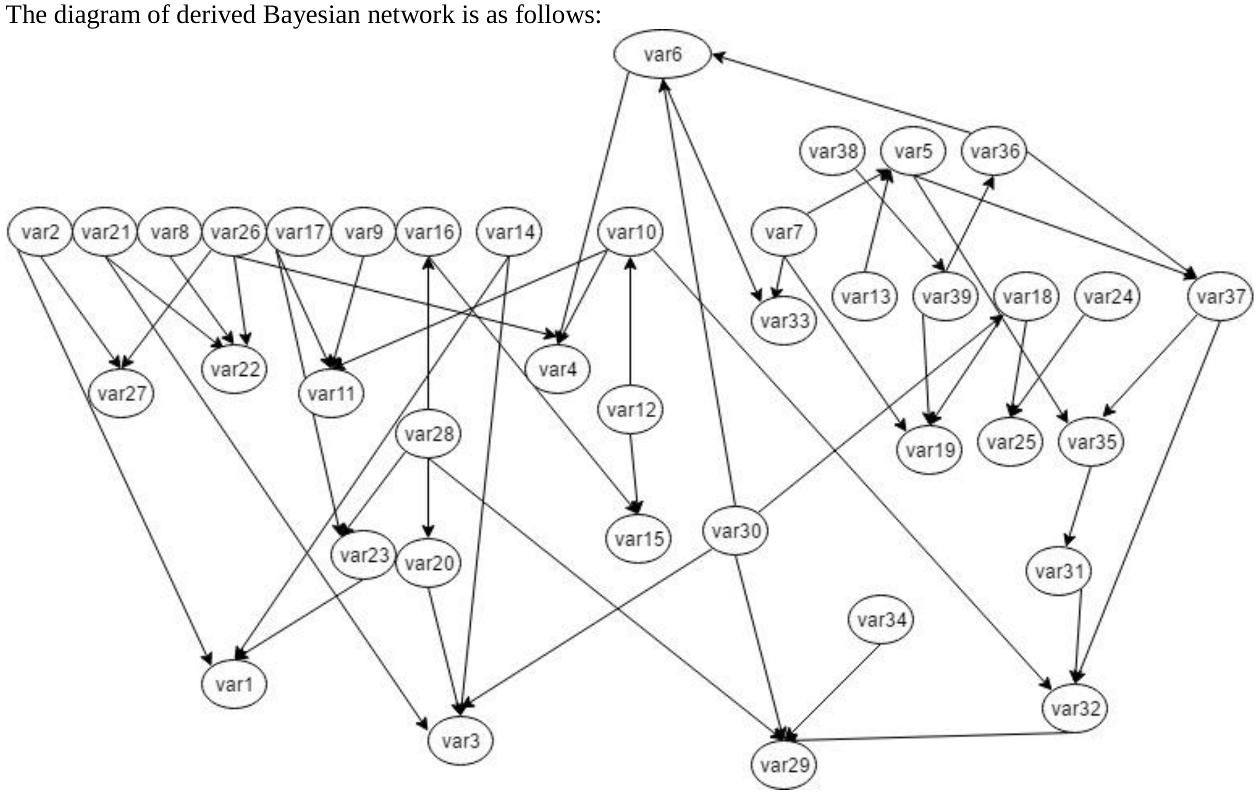
Once all v-structures have been discovered in the skeleton, the PC algorithm iteratively checks certain node/edge/arc patterns using two rules. If one of the rules matches, it is executed and the network structure is updated:
- If there is an arc from A to B, there is an edge between B and C, there is no edge between A and C, replace the edge between B and C with an arc if this does not create a new v-structure.
- If there is a directed path from A to B, meaning that there is a sequence of arcs starting at A and ending at B, and an edge between A and B, then replace the edge by an arc

We have used **pcalg** package in R to build the Bayesian network from the given dataset.
Below is the code that we have used to build the skeleton network

// Alpha represents the significance level for individual partial correlation tests

```
library(pcalg)
mydata = read.csv ("data_MRI.csv", header=TRUE)
gSkel <- pcAlgo (mydata, alpha=0.05, corMethod="pearson")
(gPDAG <- udag2pdag(gSkel))
Plot(gPDAG)
```

The diagram of derived Bayesian network is as follows:



Mapping of the functional regions and the nodes in the Bayesian network

var1	L Aud	var21	Vis
var2	R Aud	var22	R LOC
var3	Straite	var23	D ACC
var4	L DMN	var24	V ACC
var5	Med DMN	var25	R A Ins
var6	Front DMN	var26	L STS
var7	R DMN	var27	R STS
var8	Occ Post	var28	L TPJ
var9	Motor	var29	Broca
var10	R DLPFC	var30	Sup Front S
var11	R ront pol	var31	R TPJ
var12	R Par	var32	R Pars Op
var13	R Post Temp	var33	Cereb
var14	Basal	var34	Dors PCC
var15	L Par	var35	L Ins
var16	L DFPFC	var36	Cing
var17	L Front pol	var37	R Ins
var18	L IPS	var38	L Ant IPS
var19	R IPS	var39	R Ant IPS
var20	L LOC		

2.1 Parameters used in Bayesian Network creation

Pearson's $\chi 2$ test is used by PC algorithm to measure the conditional independence between two variables in a dataset. This concept is quite important while building a Bayesian network involving given variables, as it helps to determine whether an edge should exist between given pair of variables.

Log Loss of the Bayesian network is calculated as 1.4189

We did not assume any limitations on number of parents per node

3 Building Linear Gaussian Bayesian Model and calculation of CPDs

As we are dealing with continuous random variables in this dataset, we cannot use a table-based representation for calculating the conditional probability distributions (CPD's) of random variables. In such a scenario, one can circumvent this issue by discretizing the random variable values, but this approach will often lead to loss of structure that characterizes the very behavior of that variable. As an instance consider a Robot moving in a plane, the million values of robot positions cannot always be associated with random probability. Basic continuity assumptions imply relationships between probabilities of nearby discretized values. Such constraints are hard to capture in a discrete distribution where there is no notion that two values are close to each other.

While dealing with continuous variables in Bayesian networks, we can classify it into one of the three categories
1) Continuous child and parents
2) Continuous child with discrete and continuous parents (Xdisc and Xcont)
3) Discrete child Y with continuous parent X

Here we are dealing with continuous child as well as continuous parents. The inbuilt modules of pgmpy 'Linear Gaussian Bayesian Network' and Linear Gaussian CPD' to build the Bayesian Network and calculation of CPDs. In order to use this function, we first need to pre calculate the beta vector, which represents the coefficients of parent nodes in linear equation. Each of the parent nodes are assumed to have a Gaussian noise with mean 0 and variance σ2

Let Y be a continuous variable with continuous parents $X_1 \ldots X_k$
We say that Y has a linear Gaussian model with parameters $\beta_0 \ldots \beta_k$ and σ^2
such that $P(Y|x_1,..x_k) \sim N(\beta_0 + \beta_1 x_1 + \ldots \beta_k x_k ; \sigma^2)$
In our case ,the Beta vector is calculated using Linear Regression provided by sklearn.

CPDs calculated for 39 nodes are as follows.

P(var1 | var2, var14, var23) = N(0.748011140248*var2 + 0.122129899091*var14 + 0.104279336211*var23 + 0.000101396833929; 1.0)
P(var2) = N(0; 1.0)
P(var3 | var14, var20, var21, var30) = N(-0.160871888492*var14 + 0.14395222896*var20 + 0.50519722976*var21 + -0.161368368763*var30 + -0.000273447330196; 1.0)
P(var4 | var6, var10, var26) = N(0.472209861533*var6 + -0.207914792048*var10 + -0.317906750109*var26 + -0.000208583984668; 1.0)
P(var5 | var7, var13) = N(0.612729359694*var7 + 0.141762358707*var13 + 0.000345323950826; 1.0)
P(var6 | var30, var36) = N(0.316281430645*var30 + -0.419152394822*var36 + 0.000170436259728; 1.0)
P(var7) = N(0; 1.0)
P(var8) = N(0; 1.0)
P(var9) = N(0; 1.0)
P(var10 | var12) = N(0.337448384063*var12 + -0.000190041140001; 1.0)
P(var11 | var9, var10, var17) = N(-0.126538532496*var9 + 0.22976436421*var10 + 0.692552775779*var17 + -0.000173862156993; 1.0)
P(var12) = N(0; 1.0)
P(var13) = N(0; 1.0)
P(var14) = N(0; 1.0)
P(var15 | var12, var16) = N(0.364837980075*var12 + 0.32135097098*var16 + -0.000179503972552; 1.0)
P(var16 | var28) = N(0.308256198226*var28 + 8.47201631016e-05; 1.0)
P(var17) = N(0; 1.0)
P(var18 | var30) = N(-0.392229996829*var30 + -5.84363390858e-05; 1.0)
P(var19 | var7, var10, var18, var39) = N(0.219223146521*var7 + 0.292000475241*var10 + 0.39585186604*var18 + -0.224568727184*var39 + 0.000142865596228; 1.0)
P(var20 | var12, var22, var26, var28) = N(-0.259789924729*var12 + 0.364552729375*var22 + -0.318814907437*var26 + -0.198736013635*var28 + -9.57625817105e-05; 1.0)
P(var21) = N(0; 1.0)
P(var22 | var8, var21, var26) = N(0.23009983691*var8 + 0.457827132222*var21 + -0.31175530805*var26 + -4.09432874194e-05; 1.0)
P(var23 | var17, var28) = N(0.362645053502*var17 + 0.313423013829*var28 + 3.36790699307e-05; 1.0)
P(var24) = N(0; 1.0)
P(var25 | var10, var19, var24) = N(0.380799057427*var10 + -0.00570509488907*var19 + 0.221949580904*var24 + 0.00028431637559; 1.0)
P(var26) = N(0; 1.0)
P(var27 | var2, var26) = N(0.131565493534*var2 + 0.748502553937*var26 + 0.000112593911244; 1.0)
P(var28) = N(0; 1.0)
P(var29 | var28, var30, var32, var34) = N(0.18962578868*var28 + 0.173014624825*var30 + 0.277480342158*var32 + -0.352280868362*var34 + -1.65822297619e-05; 1.0)
P(var30) = N(0; 1.0)
P(var31 | var35) = N(0.437621989043*var35 + 8.10350788248e-05; 1.0)
P(var32 | var10, var31, var37) = N(0.212595832207*var10 + 0.326492497699*var31 + 0.374880002231*var37 + -0.000298875303486; 1.0)
P(var33 | var6, var7) = N(-0.296104669536*var6 + -0.236304436207*var7 + 8.7157666957e-05; 1.0)
P(var34) = N(0; 1.0)
P(var35 | var5, var14, var37) = N(-0.268239540802*var5 + 0.276469227659*var14 + 0.456672204003*var37 + 5.02332852235e-06; 1.0)
P(var36 | var39) = N(0.312885381245*var39 + 0.000191212723152; 1.0)
P(var37 | var5, var36) = N(-0.517239611541*var5 + 0.254355383522*var36 + 0.000216326603927; 1.0)
P(var38 | var39) = N(0.371947226006*var39 + 2.41025366904e-06; 1.0)
P(var39) = N(0; 1.0)

3.1 Parameters calculated from the dataset and samples

Hamiltonian Monte Carlo sampling has been used to draw 176 samples from the joint Gaussian probability distribution .HMC module of pgmpy is used for the same purpose

The tabulation of the calculated parameters from the samples and the dataset is as follows

Node	Mean for Given Values	Entropy for Given Values	Mean for Sampled Values	Entropy For Sampled Values	Relative Entropy
1	-2.47681E-14	1.4189385332046731	0.007738054	1.431131271	0.030391227
2	1.68764E-13	1.4189385332046724	0.105598593	1.423631202	-0.01391059
3	-1.02471E-13	1.4189385332046729	-0.000850368	1.440792605	0.070754772
4	6.04062E-15	1.4189385332046738	-0.114138778	1.340722534	0.188049119
5	-9.44194E-14	1.4189385332046722	0.013328141	1.56920503	0.033091002
6	1.31845E-13	1.4189385332046724	0.119402053	1.482966236	-0.037189338
7	-1.57914E-13	1.418938533204672	0.007954244	1.480621519	-0.015330289
8	8.17427E-14	1.4189385332046733	0.157143006	1.525400176	0.051554811
9	-1.65929E-13	1.4189385332046731	-0.010051501	1.561061257	0.03333172
10	2.47804E-13	1.4189385332046724	0.02049647	1.535144527	-0.057890128
11	-2.08924E-15	1.4189385332046722	0.076937081	1.511258465	0.040288676
12	-1.84607E-13	1.4189385332046727	-0.107001571	1.603817871	0.038390118
13	-8.02868E-14	1.4189385332046731	0.016045775	1.435158141	0.118777261
14	8.04798E-14	1.4189385332046722	0.108810313	1.548423107	0.023196814
15	-4.10783E-15	1.4189385332046731	0.037083217	1.41959978	-0.010815953
16	-1.95793E-13	1.4189385332046736	0.024938236	1.593451328	0.045606916
17	6.83292E-15	1.4189385332046724	0.020940768	1.451926207	0.008649333
18	1.01413E-13	1.4189385332046724	-0.006852401	1.474823344	-0.027676884
19	-5.30131E-14	1.4189385332046731	-0.079801792	1.530266897	0.047112283
20	1.56087E-13	1.418938533204672	-0.237011184	1.44473165	0.112170018
21	1.1617E-13	1.4189385332046731	-0.061182554	1.528966523	0.016451506
22	-1.93694E-13	1.4189385332046724	-0.136094299	1.456743442	0.010298343
23	3.58128E-13	1.4189385332046724	0.018807871	1.537116559	0.059924408
24	1.37698E-13	1.4189385332046716	-0.10931747	1.444137814	0.055736757
25	-6.05475E-14	1.4189385332046729	0.062148648	1.49749644	0.014268192
26	8.71121E-14	1.418938533204672	-0.12984327	1.473379637	-0.003841653
27	-2.56441E-13	1.4189385332046729	0.005720528	1.540264937	-0.001918243
28	3.41944E-13	1.4189385332046733	0.046777877	1.356128103	0.048482726
29	1.77696E-13	1.4189385332046724	-0.089990514	1.535058954	-0.010168372
30	3.18634E-14	1.4189385332046724	0.059358256	1.514587955	0.084010127
31	-7.65246E-14	1.4189385332046729	0.090943851	1.484946551	0.053277844
32	1.97786E-13	1.4189385332046731	-0.031285511	1.520194601	0.069272503
33	-3.24104E-13	1.4189385332046729	0.140237253	1.596958155	0.130930846

34	1.38849E-13	1.4189385332046731	0.012333523	1.585862176	0.076977812
35	-2.40161E-14	1.4189385332046729	0.115799226	1.522284059	0.0581792
36	5.27432E-14	1.418938533204672	0.013807929	1.460854198	0.153194419
37	8.99533E-14	1.4189385332046731	0.1297766	1.512607228	0.131951602
38	5.35228E-14	1.418938533204672	-0.012331545	1.492094395	0.018071648
39	-4.73863E-14	1.4189385332046727	0.190666498	1.476716031	0.130220343

The data values in the samples and the given dataset are very small in magnitude .Hence round-off is not considered.

4 Inference:

We have used 'Approximate Inference' which considers the samples obtained from the Hamiltonian Monte Carlo sampling technique. The sample technique is used on the joint probability distribution made with child node and parents in the corresponding CPD

Four Queries are being considered for making the inference. In these queries the mean of child is calculated using the linear equation in the CPD. The variance is calculated using the all the data values in the child. The resultant mean and variance values are used for finding the probability using the logpdf function for the childnode sample for single data point which is in row5 . The row5 corresponds to time point 5 among all the 176 time points

4.1 Query1:

P(var1 | var2, var14, var23) = N(0.748011140248*var2 + 0.122129899091*var14 + 0.104279336211*var23 + 0.00010139683392,0.515764973359)

Probability value of the query: 0.41

Node	Functional Region Name	Functionality	Node Status
var1	L Aud	Speech Perception	child
var2	R Aud	Pitch change Perception	parent
var14	Basal	Voluntary Movement Control	parent
var23	D Acc	Social Processes Detection	Parent

From this query it can be observed that when the patient observes people ,moves his body and senses change in pitch of sound of a person he perceives the content of speech with a probability value 0.41

4.2 Query2:

P(var4 | var6, var10, var26) = N(0.472209861533*var6 + -0.207914792048*var10 + -0.317906750109*var26 + -0.000208583984668; 1.11899421783)

Node	Functional Region Name	Functionality	Node Status
var4	L DMN	Paying attention	child
var6	Front DMN	Mind Wandering	parent
var10	R DLPFC	Abstract reasoning	parent
var26	L STS	Language recognition	Parent

Probability value of the query: 0.31

From this query it can be observed that when the patient recognizes language and reasons the content of speech and wanders his mind he pays attention with a probability of 0.3

4.3 Query3:

P(var11 | var9, var10, var17) = N(-0.126538532496*var9 + 0.22976436421*var10 + 0.692552775779*var17 + -0.000173862156993; 0.900560514493)

Node	Functional Region Name	Functionality	Node Status
var11	R ront pol	Non-verbal abilities	child
var9	Motor	Voluntary Movements	parent
var10	R DLPFC	Abstract reasoning	parent
var17	L Front pol	Language related movement	Parent

Probability value of the query: 0.30

From this query it can be observed that when the patient recognizes language and reasons the content of speech and moves his body , he may communicate non-verbally with a probability of 0.3

4.4 Query4:

P(var25 | var10, var19, var24) = N(0.380799057427*var10 + -0.00570509488907*var19 + 0.221949580904*var24 + 0.00028431637559; 0.741570405167)

Probability value of the query: 0.26

Node	Functional Region Name	Functionality	Node Status
var25	R A Ins	Awareness of body states	child
var10	R DLPFC	Abstract reasoning	parent
var19	R IPS	Visual Perception	parent
var24	V ACC	Decision making	Parent

From this query, it can be observed that when the patient makes a decision, observes something and reasons he perceives the states of his own body with a probability of 0.3

In reality, one functional region of brain corresponds to one or more cognitive processes. The inference that has been done may not be accurate but should be considered as a means to understand how one process could trigger another cognitive process based on the activation of regions in the brain.

5 Conclusion:

The dataset belongs to a single patient suffering from ADHD. We have tried to understand the cognitive processes based on the activation of neural regions of the brain using Bayesian modelling. Nevertheless same project work when executed on more subjects who are also healthy would give a better and broad picture of functionality of brain.

6 References

[1] https://staff.utia.cas.cz/files/mtr/pgm06/41_paper.pdf
[2] http://d-scholarship.pitt.edu/21786/1/deJongh_etd2014.pdf
[3] https://people.cs.umass.edu/~mccallum/courses/gm2011/03-cpd.pdf
[4] http://www.cedar.buffalo.edu/~srihari/CSE674/Chap5/5.1-Local%20CPDs.pdf
[5] https://cran.r-project.org/web/packages/pcalg/pcalg.pdf
[6] http://uregina.ca/~gingrich/ch10.pdf
[7] https://pdfs.semanticscholar.org/faf4/e454370b907f3eea903e0f6562339ecc3edb.pdf