**STA 546 – Home Work #3**

**Suruchi JaiKumar Ahuja**

Q1). The library and R package igraph is used for network analysis. The main goals of the igraph library is to provide a set of data types and functions for pain-free implementation of graph algorithms, fast handling of large graphs, with millions of vertices and edges, allowing rapid prototyping via high level languages like R.

The network Les miserables (miserables) has 77 vertices and 254 edges. The network zachary’s karate club (karate) has 34 vertices and 78 edges. The network Dolphin social network (dolphins) has 62 vertices and 159 edges. These three networks were loaded into R environment using:

>> library(igraph)

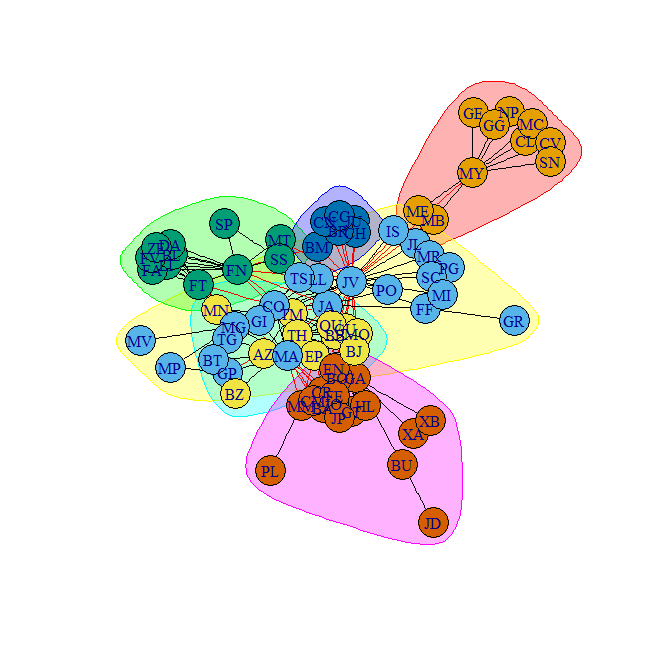
>> nexus.get(“karate”)

>> nexus.get(“misearbles”)

>> nexus.get(“dolphins”)



The Miserables Plot was plotted.



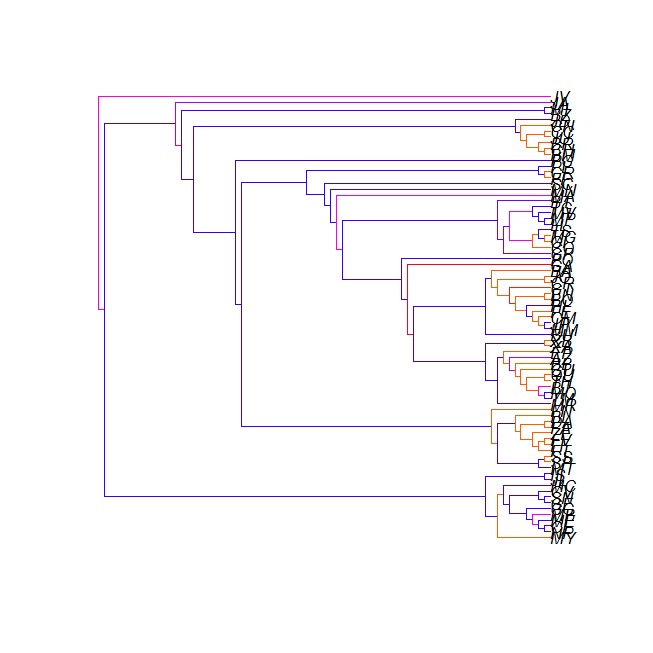
The cluster optimal is used to show communities on the Miserables network was performed. The ‘optimal\_cluster’ function samples the network in a fast greedy approach. The membership is assigned to the original network and the graph is plotted along with the optimal\_clusters

1. Hrg model at level at level 5

Hierarchical random graph for the Miserables network, at level 5:

|  |
| --- |
| g1 p= 0.47 JV  '- g65 p= 0  '- g25 p= 0  '- g47 p= 1 MY  '- g13 p= 0 GG MB ME CL NP CV SN GE MC  '- g34 p= 0 JL IS  '- g74 p= 0.25 JA  '- g20 p= 0.016  '- g8 p=0.0053 PG FN MT SS SP BL ZE FT FV LI DA FA SC MN MA PO GA MR EP AZ BB GU BJ MO TM QU TH XB XA  BU BA GT HL FE CM JP MM EN BO JO CR BT GP CO TS MG GI TG MV MP LL PL GR FF JD CN BR CH  BM CC JU  '- g73 p= 0 BZ MI |
|  |
| |  | | --- | |  | |

The miserables network, hrg model was plotted to form a dendogram.



Hierarchical random graph for Dolphins network, at level 5:

g1 p=0.016 Fork

'- g56 p= 0.12 SN100

'- g17 p=0.064

'- g42 p= 0

'- g18 p= 0.12 Vau Cross TSN103 SN96 TR77 Fish Beak Topless

MN83 Haecksel Patchback MN60 MN105 Trigger Jonah Five

CCL Zap Double Thumper Shmuddel Whitetip TR120 Zipfel

TSN83 Kringel Bumper Grin Scabs Hook SN4 SN9

TR99 Stripes SN63 TR88 SMN5

'- g33 p= 0.11 Wave Upbang SN90 Jet MN23 Ripplefluke Zig

SN89 Beescratch TR82 Quasi Feather Web Gallatin

DN21 DN16 Notch Number1 Mus Knit

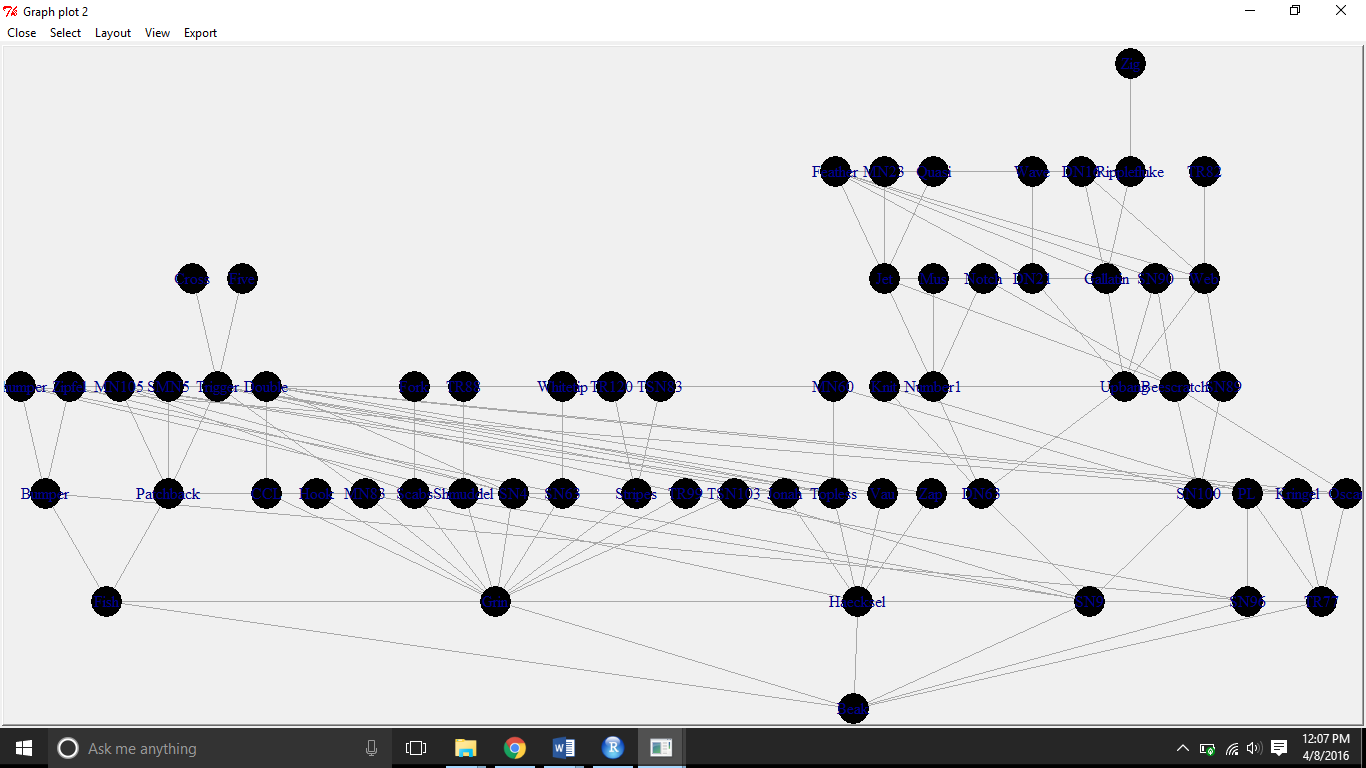
'- g25 p= 1 PL

'- g49 p= 0 DN63 Oscar

The dolphins network was fitted into an hrg model and then plotted to form a dendogram.



The tkplot function were used which serve as an interactive graph drawing facility. The various graphs can be observed by changing layouts and using selecting different edges and vertices. Here the Reingold-Tilford layout is shown.



1. Creating noisy datasets on the dolphins network:

The noisy datasets were created on the dolphin network using sampling methods, where values were generated randomly corresponding to the edges and then deleted from the true dolphins graph.

set.seed(1)

delete1 <- sample(159,size =length(E(dolphins))\*0.05, replace = FALSE, prob = NULL )

> graph1 <- delete\_edges(dolphins, delete1)

> graph1

IGRAPH UN-- 62 136 -- Dophin social network

+ attr: Description (g/c), name (g/c), Author (g/c), Citation (g/c), URL (g/c), name (v/c)

+ edges (vertex names):

[1] CCL --Double DN16 --Feather DN21 --Feather Beak --Fish

[5] Bumper --Fish DN16 --Gallatin DN21 --Gallatin Feather --Gallatin

[9] Beak --Grin Beak --Haecksel Grin --Hook Beescratch--Jet

[13] DN21 --Jet Feather --Jet Gallatin --Jet Haecksel --Jonah

[17] Beescratch--Knit DN63 --Knit Double --Kringel Hook --Kringel

[21] Jonah --Kringel Jonah --MN105 Jet --MN23 Grin --MN83

[25] Haecksel --MN83 Jet --Mus Beescratch--Notch Jet --Number1

[29] Mus --Number1 Notch --Number1 Beescratch--Oscar Double --Oscar

+ ... omitted several edges

This edges were deleted using : edges1<-E(dolphins)[delete1]

[1] Jet --Quasi TR99 --Trigger SN100 --SN4 Trigger--Vau

[5] SN89 --Web Feather--Gallatin Beak --SN96

the Predict\_edges function is used to predict the missing links for 5% of data deletion

dol\_predict<-predict\_edges(graph1)

The methods: MCMC sampling on the graph predicts the missing links. It displays a list of probabilities and edges and also the hierarchal Random Graph model of the prediction results.

The occurrence of the deleted edges in the results and the edges predicted can be matched to check for accuracy. A potential subset of the edge predictions is made which has edges in our actual set of missing values. A sample of top 100 predicted edges is undertaken and looked for the presence of a edge from our deleted edges list. The more the number of edges which can be traced, the better is our accuracy of prediction.

V1 V2

15 Feather Gallatin

53 Beak SN96

112 Beak SN4

150 SN100 SN4

167 SN100 Trigger

206 TR99 Trigger

301 SN100 SN96

373 Beak Trigger

468 SN100 Vau

622 Trigger Vau

662 Jet Quasi

686 TR99 Vau

695 Beak Vau

837 SN89 Trigger

845 SN100 Web

905 SN89 SN96

980 SN89 Vau

1052 Beak Quasi

1107 Jet Vau

1126 Feather Quasi

1242 SN89 Web

1251 Jet SN96

1252 Feather SN96

1378 Feather Vau

1465 Beak Gallatin

1466 Beak Web

1646 TR99 Web

1658 Jet Trigger

1681 Trigger Web

1689 Feather Trigger

1716 Jet SN4

1739 Feather SN4

It is observed that only two edge connections is accurately predicted within the top 100 predictions.

c)

The missing edges after deletion of 15% and 40% of the edges from the primary dataset were then predicted.

The deleted edges for 15% of edges removal are

[1] Feather --Jet Mus --Number1 Topless --TR99

[4] DN63 --Upbang Patchback --Trigger MN105 --Scabs

[7] Beescratch--Jet Knit --PL Beescratch--Oscar

[10] MN60 --SN100 Beescratch--SN100 Jet --Number1

[13] Hook --TR99 MN105 --Trigger Scabs --Stripes

[16] Double --Topless MN60 --Trigger SN100 --SN4

[19] Stripes --TR120 MN105 --Patchback Haecksel --MN83

[22] Patchback --Topless Topless --Trigger

And the predicted edges are:

from to

2 Feather Upbang

3 Scabs Stripes

5 Feather Jet

13 Jet Upbang

25 Topless Trigger

34 Patchback Topless

39 Haecksel MN83

49 Beescratch Jet

57 Jet Number1

67 Haecksel Trigger

71 MN105 Trigger

72 MN105 MN83

78 Haecksel Patchback

84 MN105 Patchback

86 Hook Stripes

90 Knit Number1

93 Patchback Trigger

101 Mus Number1

102 Patchback SN4

113 Haecksel SN4

114 Hook Patchback

118 Patchback Scabs

121 Patchback TR99

135 Double Patchback

145 Patchback PL

146 Stripes TR99

147 Double Scabs

148 SN100 SN4

150 Double SN100

152 Patchback SN100

160 Double Topless

167 Double Trigger

171 Hook TR99

172 Mus Upbang

179 Haecksel Scabs

194 Scabs Trigger

196 Hook Trigger

197 Stripes Trigger

201 Haecksel TR99

202 SN100 Trigger

203 Scabs SN100

204 MN60 Trigger

207 Haecksel SN100

211 SN100 Topless

219 Haecksel Stripes

225 MN60 Patchback

232 Scabs Topless

235 Double TR99

247 MN105 Scabs

257 Haecksel Oscar

261 Double Stripes

286 Haecksel PL

287 DN63 Jet

295 Topless TR99

298 Double MN83

318 Hook SN100

336 Hook Topless

340 DN63 Upbang

348 SN100 TR99

350 Double PL

352 Stripes Topless

360 Hook MN83

367 SN100 Stripes

422 MN105 TR99

425 MN60 SN4

430 Feather Number1

443 Hook Oscar

472 MN105 Stripes

476 MN105 SN100

516 MN105 Oscar

520 MN60 MN83

521 Hook PL

598 DN63 Patchback

618 DN63 Oscar

619 DN63 Trigger

621 DN63 SN4

630 DN63 Scabs

651 DN63 TR99

654 DN63 Stripes

656 MN60 Scabs

658 DN63 Topless

680 DN63 MN83

682 Patchback TR120

688 MN105 PL

690 DN63 SN100

695 MN60 SN100

728 MN60 TR99

731 DN63 TR120

It is observed that 9 out of the 23 of the deleted links are accurately predicted in the top 100 predicted edges list.

The deleted edge list for 40% of edges removal are

|  |
| --- |
| Fromedge Toedge  1 Hook TR99  2 Feather Jet  3 DN16 Feather  4 SN63 Thumper  5 SN4 SN9  6 Beak Fish  7 SN4 Stripes  8 Haecksel MN83  9 Patchback SMN5  10 MN105 Scabs  11 Feather Ripplefluke  12 SN100 SN9  13 Kringel Oscar  14 Scabs SN63  15 Scabs SN4  16 DN21 Feather  17 Jonah MN105  18 Shmuddel Thumper  19 Shmuddel TR88  20 Gallatin Jet  21 Gallatin Ripplefluke  22 MN105 SN4  23 Jonah Kringel  24 MN83 Patchback  25 MN60 SN100  26 SN100 SN89  27 SN90 Upbang  28 Grin SN63  29 SN96 TR77  30 DN21 Upbang  31 Beescratch Notch  32 Beescratch Oscar  33 Grin Stripes  34 Trigger Vau  35 Mus Number1  36 CCL Zap  37 DN63 SN9  38 Notch Number1  39 Jonah MN83  40 SN63 Stripes  41 SN9 TSN103  42 Ripplefluke Zig  43 MN105 Trigger  44 Topless TR99  45 Feather Web  46 DN63 Knit  47 Grin SN9  48 Fish TR77  49 Beak Grin  50 Bumper SN96  51 TR99 Trigger  52 SN100 Zap  53 DN63 PL  54 MN83 Topless  55 Oscar PL  56 Jonah Topless  57 Fish SN96  58 Kringel SN100  59 Knit PL  60 Mus Notch  61 SN63 TSN103  62 Gallatin Upbang  63 Beescratch Jet |
|  |
| |  | | --- | |  | |

And the predicted edge list is

V1 V2

1 SN4 SN63

2 Feather Upbang

4 DN16 Upbang

6 DN16 Jet

7 MN105 MN83

9 Haecksel Trigger

10 SN4 TR99

11 Haecksel Patchback

12 Scabs TSN103

13 Haecksel MN105

14 SN63 SN9

17 Shmuddel SN63

18 Ripplefluke Web

20 SN4 TSN103

22 Bumper TR77

23 SN63 TR99

24 Ripplefluke Upbang

25 Fish PL

29 Grin Kringel

32 Hook Stripes

33 Oscar SN96

34 Kringel Scabs

35 Kringel SN4

36 Hook TSN103

37 Fish Oscar

38 Hook SN9

40 Beescratch Web

43 Beescratch Feather

47 Kringel Stripes

49 Kringel SN9

53 Beak SN96

54 DN63 Jet

55 Notch Upbang

56 Shmuddel TSN103

59 Mus Upbang

61 Kringel TSN103

63 SN9 TR99

65 TR99 TSN103

66 Bumper PL

68 Kringel SN96

69 CCL Kringel

70 Kringel Zap

71 Knit Number1

72 SN9 Stripes

74 CCL Topless

75 Fish Kringel

76 Beescratch Ripplefluke

77 Shmuddel TR99

79 MN60 Patchback

80 Beak Kringel

81 Bumper Kringel

82 Bumper Oscar

85 DN21 Ripplefluke

86 DN63 Notch

88 Haecksel Kringel

89 Kringel Topless

90 Kringel Trigger

91 Kringel Patchback

92 Kringel MN83

93 Kringel MN105

94 Knit Notch

95 Grin Thumper

96 Beak Scabs

97 Beak PL

99 Kringel PL

100 MN60 MN83

103 DN16 Ripplefluke

104 SN4 Thumper

105 Scabs Thumper

107 Shmuddel SN9

108 Trigger Zap

110 Shmuddel Stripes

111 Beak SN63

112 Beak SN4

114 Hook Thumper

115 SN9 Thumper

125 Patchback Zap

126 Beak Oscar

128 MN83 Zap

122 Jonah Zap

123 Haecksel TR99

131 Beak Stripes

132 Haecksel SN100

134 SN100 Topless

135 MN105 Zap

136 Haecksel SN4

137 Haecksel TSN103

138 CCL SN100

140 Grin SN100

141 CCL Scabs

144 Haecksel SN63

146 Haecksel Stripes

147 Haecksel Scabs

150 SN100 SN4

152 SN100 SN63

154 Scabs SN100

161 Beak TR99

162 SN100 TR99

163 Shmuddel SN100

165 Hook SN100

166 SN100 Stripes

167 SN100 Trigger

169 CCL SN63

170 SN96 Thumper

171 Kringel TR88

172 Fish Thumper

173 CCL SN4

175 Patchback SN4

177 MN83 SN4

178 Jonah SN4

179 SN4 Trigger

181 Patchback SN100

182 Jonah SN100

183 MN83 SN100

184 MN105 SN100

186 Beak TSN103

187 SN100 TSN103

189 Knit Web

192 SN100 Thumper

194 SN4 Zap

195 Topless TSN103

196 SN9 Topless

197 Patchback SN9

198 Grin MN105

199 Jonah Stripes

203 Beak Thumper

204 Patchback Scabs

205 MN83 Stripes

206 TR99 Trigger

207 MN105 Stripes

208 Scabs Topless

209 Grin Patchback

210 Grin Topless

211 Jonah SN9

212 MN83 SN9

213 Hook Patchback

214 SN9 Trigger

215 Grin Trigger

216 MN105 SN9

218 MN83 Scabs

219 Hook Topless

220 Jonah Scabs

221 Scabs Trigger

222 Patchback SN63

223 SN63 Topless

224 MN105 SN63

225 Hook MN105

226 Hook MN83

228 Hook Trigger

229 MN60 Zap

230 MN83 SN63

231 SN63 Trigger

232 Jonah SN63

233 MN105 TSN103

235 Trigger TSN103

237 Shmuddel Topless

238 Feather Knit

239 Jonah TSN103

240 Gallatin Knit

241 MN83 TSN103

243 Shmuddel Trigger

247 Grin Zap

249 Haecksel Thumper

250 Beak SN100

252 SN63 Zap

255 Oscar SN100

258 TR99 Zap

259 SN9 Zap

260 Patchback TR99

261 Jonah TR99

263 Scabs Zap

265 Kringel Vau

266 Bumper SN100

267 Hook Zap

268 MN83 TR99

269 CCL SN9

272 MN105 TR99

275 DN21 Knit

276 Mus Ripplefluke

277 Shmuddel Zap

278 Oscar Thumper

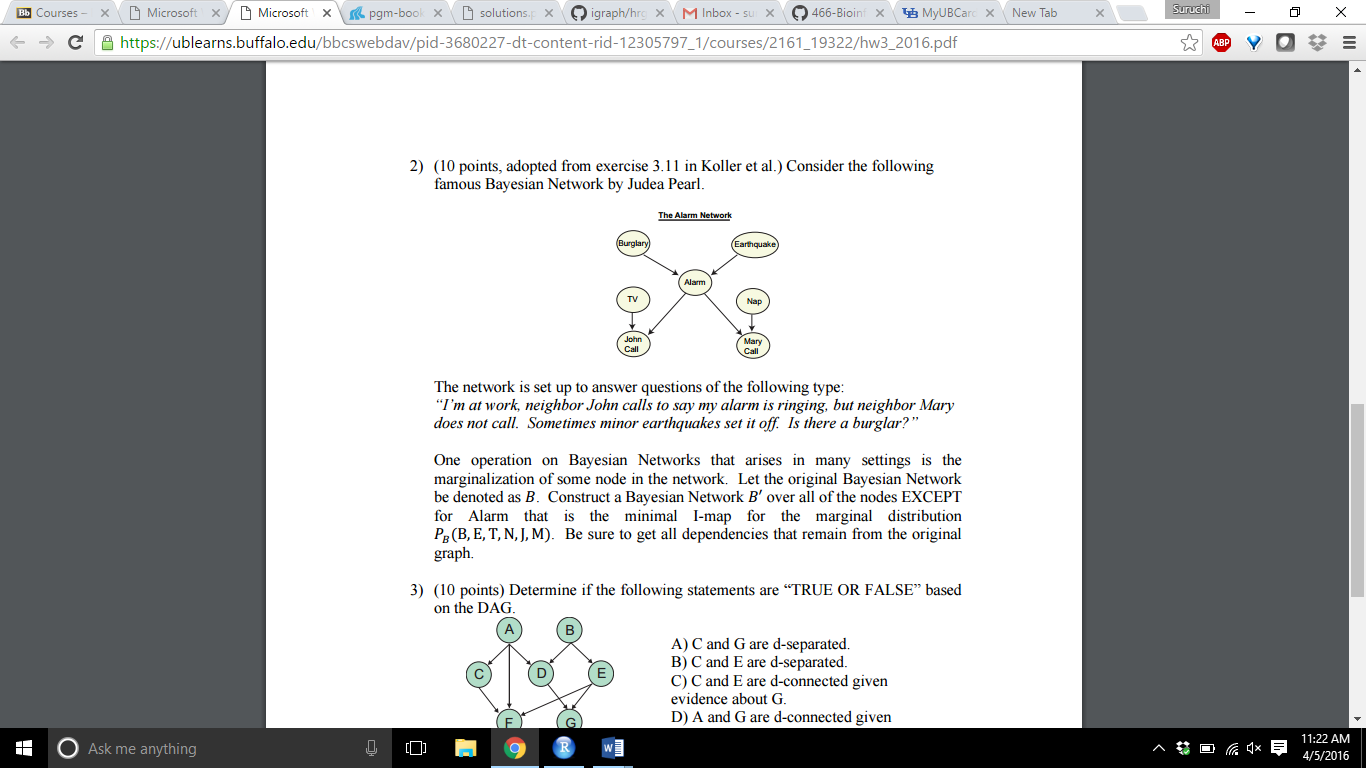
279 Knit Ripplefluke

282 DN16 Knit

285 Kringel SN89

286 CCL TR99

Q2). The following famous Bayesian Network by Judea Pearl is given below -



Consider the Original Bayesian Network be B and the Bayesian Network to be constructed as B’–

Let Burglary be R, TV be T, John be J, Alarm be A, Earthquake be E, Nap be N and Mary be M.

Active trails in B are:

R → A → J, an edge has to be inserted between R and J, to assert an independence R Ʇ J.

J ← A → M, an edge has to be inserted between J and M.

T → J ← A → M, an edge has to be added between T and M, to assert an independence (T Ʇ |J, B, E)

An edge from each parent to each child of A, has to be added.

Moreover, the active trail in G between N and J, when M is observed, is still active in B’, due to the v-structure between N and J.

No other edges are required to be added.

Now to construct the Bayesian Network B’ – The minimal I-map construction algorithm is used.

The minimal I-map is a map without redundant edges.

The variables are ordered in the topological order in B.

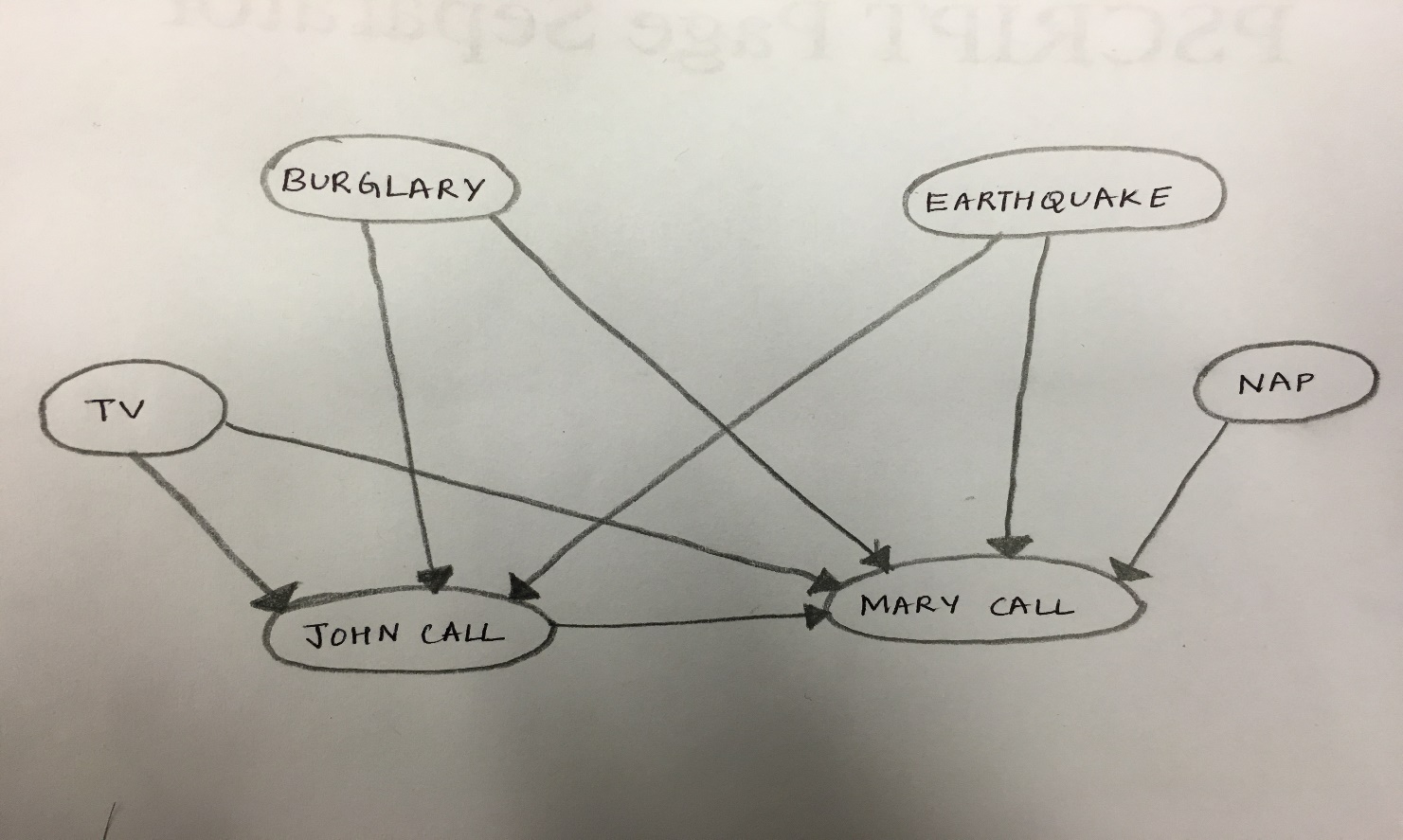
Without the loss of generality, the variables are renamed according to topological order:

. . , , ,. .,

Where is the alarm variable that is to be marginalized.

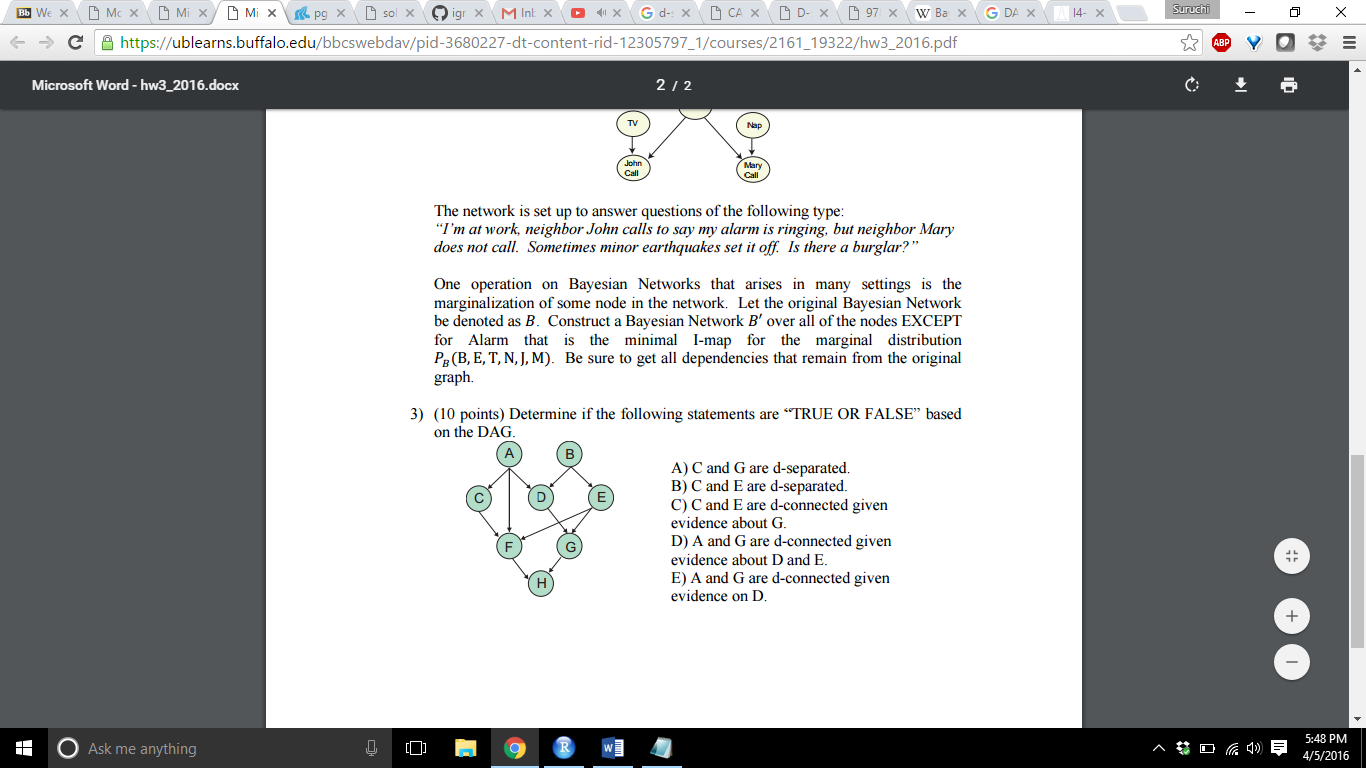
Add all the nodes to B’ using the above order, except the node and use the independencies to select a parent set for each variable. Add the variables . . , to the network B’, it results in the original network B. the parent set of variables from . . , will remain the same in constructed Bayesian Network B’, as it satisfies the local Markov assumption. The only variables that will be affected if the variable is marginalized (skipped) are those variables that has as a parent in B (‘s children). So for each child of , a new parent set is found. Now replace with a set of variables that act as a surrogate. Using the local Markov Assumption blocks the flow of information to the children variables, so the surrogate variables must block the same paths while being d-connected to the children variable of . Now consider the co-parents of , all the variables must appear in topological order. All the elements of the parent set are necessary to avoid creating independencies, so the siblings of the parents are to be added. These variables are required for the replacement of in parenting the variables of . These variables are a subset of Markov Blanket of , and they shield from other variables in the network, thus using them to replace . This ensures no other active trial can reach , and continue to the children variables.

Using this set of new parents, the local Markov Assumption is satisfied for the children variables in the Bayesian Network B’.

****

**Figure : Bayesian Network B’**

Q3) Consider the Directed Acyclic Graph shown below:



The d-separation is a criterion for deciding, whether a set X of variables is independent of another set Y, given a third set Z, from a graph.

The idea is to associate “dependence” with “connectedness” (i.e., the existence of a connecting path) and “independence” with “unconnectedness” or “separation.” To account for the orientations of the arrows the terms “d-separated” and “d-connected” (d connotes “directional”) are used.

To test whether 1 and 2 are d-separated by 3 in dag X, we need to consider every path between a node in 1 and a node in 2, and then ensure that the path is blocked by 3.

A path is blocked by 3 if at least one valve (node) on the path is ‘closed’ given 3. ν

A divergent valve or a sequential valve is closed if it is in 3, a convergent valve is closed if it is not on 3 nor any of its descendants are in 3.

a) C and G are d separated – FALSE

There is a serial connection A → D → G, allowing the transmission of information from A to G via D. So information can be transmitted from C to G via C → A → D → G, which shows that C and G are not d-separated. Hence the variables C and G are d-connected and not d-separated.

b) C and E are d separated – TRUE

The variables C and E are d-separated, since each path from C to E contains a converging connection, and since no evidence is available, each such connection will not allow transmission of information.

c) C and E are d connected given evidence about G – TRUE

The variables C and E contain a converging connection and there is no evidence available about other variables. But now since evidence is given about variable G, the variables C and E will become d-connected. Evidence about G, will show transmission of information from D to E via G, as a converging connection D → G ← E is present. Then the information from C to E can be transmitted via the diverging connection C ← A → D. So the information transmission occurs first via a divergent connection and then a convergent function,

C ← A → D → G ← E

d) A and G are d connected given evidence about D and E - FALSE

The variables A and G are d-separated, given evidence on D and E as it contains a convergent connection, D → G ← E and a straight path A →D → G, so no connection exists between A and G. Hence they are d-separated and not d-connected.

e) A and G are d connected given evidence on D – TRUE

The variables A an G are d- connected given evidence about variable D, as a straight connecting path between A and G exists, A → D → G.

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------THE END-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------