Investigating Logical Fallacies

WARWICK

CS310

Project Aims

- Logical fallacies are ever present, but they are abstract concepts
- Bayesian networks are a graphical method to represent statistical probabilities

Proposed Solution

- Seeking to represent logical fallacies visually
- Demonstrate the process through which they can be avoided
- Make use of Bayesian networks to show erroneous thinking
- Calculation of probabilities through Bayesian methods
- Develop a program to create, manipulate and load Bayesian networks
- Represent some logical fallacies in Bayesian networks

Background Knowledge

- A logical fallacy is an error in reasoning
- Existing logical fallacies include
  - Base rate fallacy
  - Conjunction fallacy
  - Many more
- There are existing Bayesian network programs, though none which are designed to highlight logical fallacies
- Some logical fallacies are formalised, though not all are easy to represent graphically
- Logical fallacies are erroneous reasoning, often occurring during causal inference

Bayesian Probability

Bayesian logic is derived from the equation

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Where P(A|B) represents the probability of event A occurring given that event B occurs. This equation can be reordered to produce.

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B|A)*P(A) + P(B|!A)*P(!A)}$$

Which provides much more flexibility for calculations within a Bayesian network.

Methodology

- Developed in Java
- Incremental approach, allowing for continuous testing and feedback
- Application of graph theory (adjacency lists, dynamic search algorithms)

Development

1. Build Bayesian network system - allows both creation and calculation across events and conditional probabilities (for command prompt)

```
-> help

These are the various commands for creating Bayesian Networks

New Event (no prior probability):   ne "<EVENT NAME>"
New Event (with prior probability): nep "<EVENT NAME>" <PROBABILITY>
New Conditional Probability:        ncp "<EVENT>"|"<COND EVENT>" <PROB>
Show all probabilities known:       list
Calculate probability for event:    get "<EVENT>"
Save Network:                       save "<FILENAME>"
Load Network:                       load "<FILENAME>"
Quit Program:                       exit

-> nep "New Event" 0.7 460 500

Added event New Event at 460, 500

-> ncp "New Event"|"C" 0.2

0.2
Added conditional event New Event|C=0.2

-> get "G"

Calculating probability for G
Searching for P(G|!E)
The probability of G is 0.56
```

Development

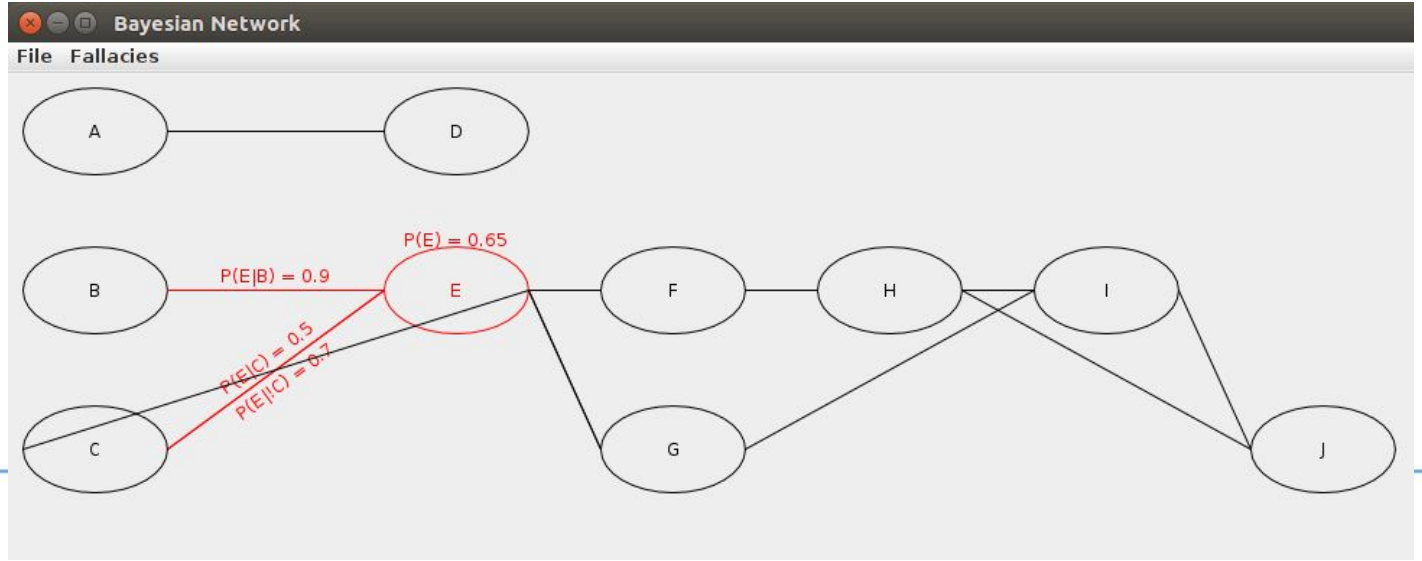
1. Build Bayesian network system - allows both creation and calculation across events and conditional probabilities (for command prompt)
2. Devise a file layout for Bayesian networks

```
H,□□□0□□□x/I,□□□□□□□x/F,□□□□□□□x/E=?&ff□□□□□□□x/A=?333□□□
□□□
/J,□□□\□□□□/B=>□□□□□□□
□□□x/C=>□□□□□□□
□□□□/G,□□□□□□□□/D,□□□□□□□
#H|F==□□□/I|H=?L□□/I|G=>□□□/F|E=?fff/E|B=?fff/E|C=?□□□/E|!C=?333/J|I=?□□□/J|H=?@□□/G|!E=>□□□/G|E=?333/D|A=>□□
```

```
A save file must first specify a collection of events that are in the system
This is done through writing the event name with an equals if it has a prior probability
as such

A=0.7

!UPDATE! Each event must now have a position, seperated out by a comma
!UPDATE 2! This is now a binary file, created in the program and then saved

Then to seperate out each event we use the / character

A=0.7/B=0.3/C=0.25/D/E/F/G/H/I/J

Since it is possible for some special characters to be a part of the event name, \ will
be used as a way to escape characters which may be recognised (including \ itself)

When the event names are specified, # character is used to seperate out for conditional
probabilities to begin being saved. Each conditional probability is written as A|B=X
where X is the probability

A=0.7/B=0.3/C=0.25/D/E/F/G/H/I/J#D|A=0.4/E|B=0.9/E|C=0.5/E|!C=0.7/J|I=0.5/J|H=0.75/H|F=0.1/G|!E=0.3/G|E=0.7/I|H=0.8/I|G=0.3/I|F=0.4/F|E=0.9

This means no spaces or new lines except from when it is part of the name of an event
```

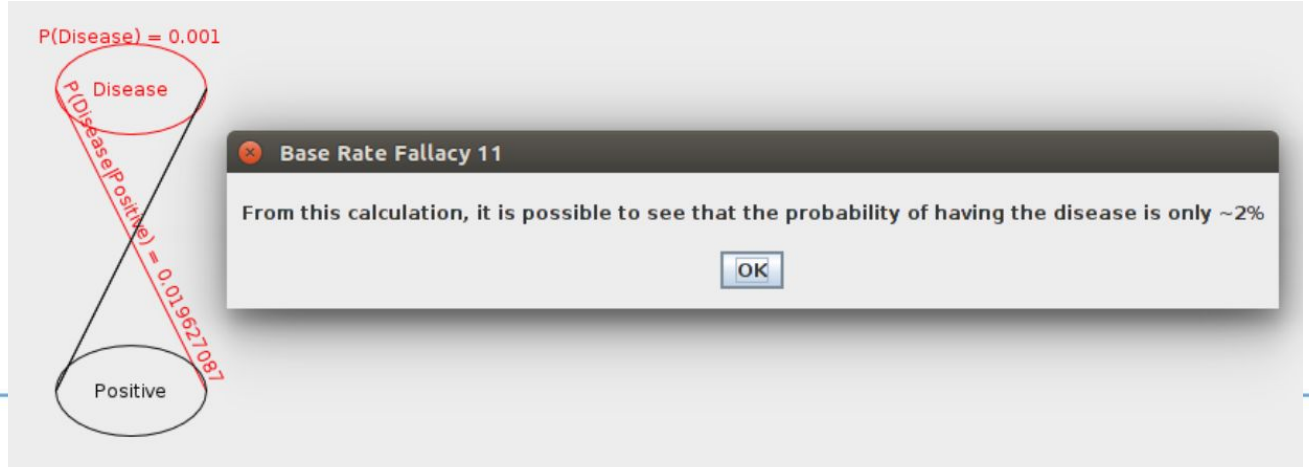
Development

1. Build Bayesian network system - allows both creation and calculation across events and conditional probabilities (for command prompt)
2. Devise a file layout for Bayesian networks
3. Visualise Bayesian networks



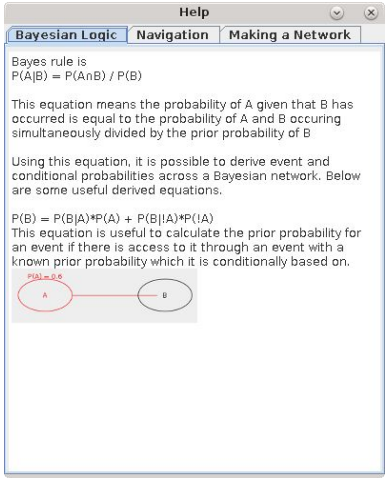
Development

1. Build Bayesian network system - allows both creation and calculation across events and conditional probabilities (for command prompt)
2. Devise a file layout for Bayesian networks
3. Visualise Bayesian networks
4. Incorporate logical fallacies into the program



Evaluation

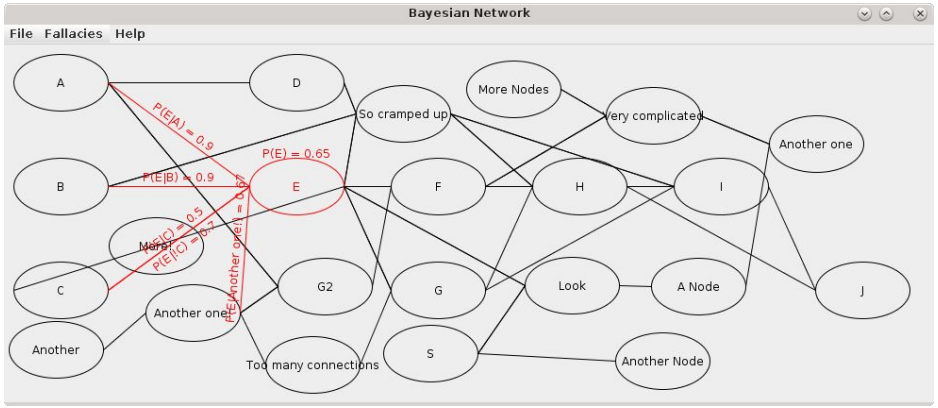
- Allows creation of Bayesian networks and calculations for probabilities
- Logical fallacies are presented and well explained, although there are few of them
- Needed improved help for new users
- Demonstrative of fallacies in Bayesian networks



Limitations

Works best with small networks, zooming functionality would be nice to add

Needs some work before functional for mobile



Project Management

Github to help track progress, rollback mistakes and maintain backups

Regular weekly meetings with project supervisor

Checklist for logical fallacies included

Regular unit testing resulting in incremental improvement

GitHub

Future Development

More Logical Fallacies

Scripting for Logical Fallacies

Mobile Development

Cloud sharing between platforms

WARWICK

Demo

WARWICK

Questions