Algorithms for Graphical Models (AGM)

Module overview

$Date: 2008/10/15 10:46:04 $

The red and the green

A simple example of probabilistic reasoning to get you started.

Crime and Punishment in Florida

Murders in Florida (1973-78)

	Sentence	
Murderer	Death	Other
Black	59	2547
White	72	2185

- 3.2% of white murderers get the death sentence

- 2.3% of black murderers get the death sentence

Crime and Punishment in Florida
The Yule-Simpson Paradox

		Sentence	
Victim	Murderer	Death	Other
Black	Black	11	2309
	White	0	111
White	Black	48	238
	White	72	2074

- For Black victims, 4.7% of Black murderers get death vs 0% for White murderers.

- For White victims, 16.8% of Black murderers get death vs 3.4% for White murderers.

The official (and true) learning outcomes

- Apply appropriate techniques from graphical modelling to handle uncertainty in real-world situations;

- Understand the sources of computational complexity in probabilistic inference;

- Understand the basics of Monte Carlo computation;

- Write Python programs to implement graphical modelling algorithms

Time to see a graphical model

- Bayesian networks are, by far, the most popular graphical model.

- Netica only runs under Windows
  `g:\apps\netica\netica 1.12\netica.exe`

- A password is required for full functionality.

Resources

Lecture overheads You're seeing some. Generally brief. Some slides are just prompts.

Scripts The scripts for the demos and animations will be provided.

gPy This is a collection of Python modules from which all scripts import their core functionality.

People Let's arrange a suitable time for the AGM surgeries now.

Draft text book

- *Bayesian Networks and Beyond: Probabilistic Models for Reasoning and Learning* by Daphne Koller & Nir Friedman

- I have access to the electronic version. You are only allowed hardcopy, purchasable (at cost) from the Departmental Office.

- This is a high-quality, very low cost text.

- The only catch is that you have to provide feedback!

Another draft text book!

- *Algorithms for Graphical Models* by James Cussens
- This would be the module textbook, but I have only written the first 141 pages.
- It should still be of some use to you, so the current version is available from the module web page.
- It's likely to be updated as the module progresses.

An actually-completed textbook

- *Probabilistic Networks and Expert Systems* by Cowell, Dawid, Lauritzen and Spiegelhalter

- Excellent book, now available in paperback at a little under £30.

- The library has 2 on ordinary loan, 2 on 1-week loan and one on short loan (in the 'key texts' section).

Why Python?

- This module focuses on the underlying *algorithms* for graphical models.

- On the one hand we don't want to clutter the presentation of the algorithms, on the other we want to avoid vagueness.

- Hence Python: high-level 'executable pseudo-code'

- Also has easy graphics (Tkinter).

Practicals

Answers to exercises will be provided, but exercises will not be marked. So it is up to you to

1. Do the exercises (in practical sessions and in your own time)

2. Ensure that you are getting a reasonable number of questions correct

3. Check that you understand the answers to exercises

4. Seek help (during practicals, during surgeries) if you are having problems.

Module web page

There is yet more teaching material which you can reach via the module page

`http://www-course.cs.york.ac.uk/agm/`

This is where all the online stuff is, including the code.

Lecture overheads are PDF. A good idea to use acroread to print to a PostScript file and then psnup them before printing.

Assessment

- It's open and partly assessed by whether your code passes various *unittests* — extreme assessment!

- 'Clarifications' are conducted entirely by email

- Submission is electronic

- Hand out: Wed/8/Autumn

- Hand in: Wed/2/Spring