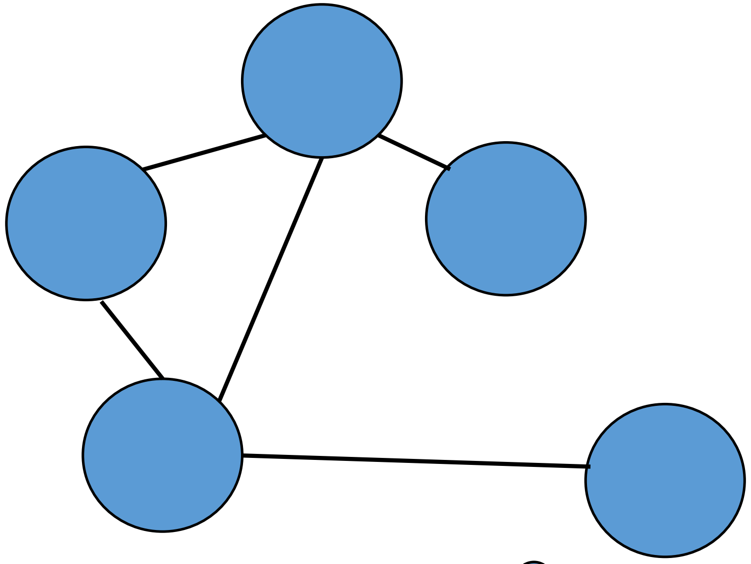
Bayesian Networks and Graph Theory

Mark Tierney

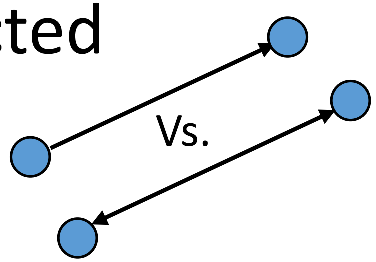
ASTR 503

Nov 9th, 2016

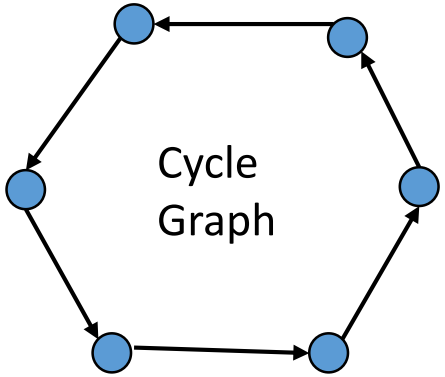
Introduction to Graph Theory/Graphical Models

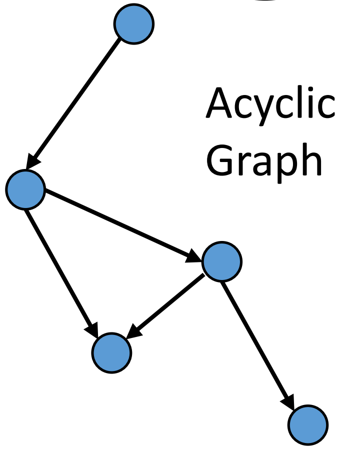


- Graphs model relationships between nodes using edges to connect them
- Can be directed or undirected

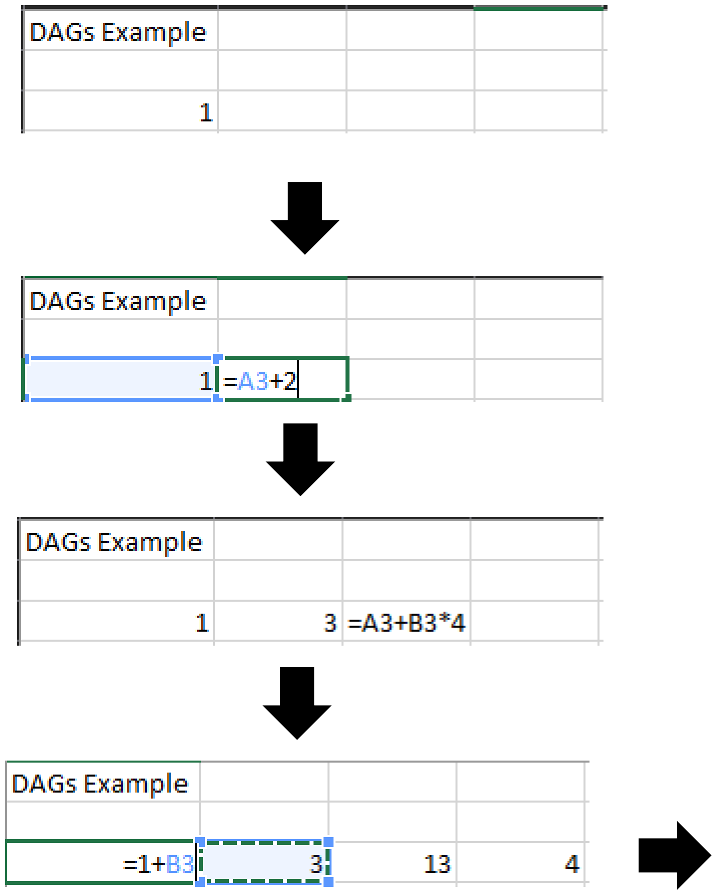


- Cycle Graphs and Acyclic Graphs
  - Bayesian Networks exclusively Acyclic
  - Cycle graph is simpler but restrictive

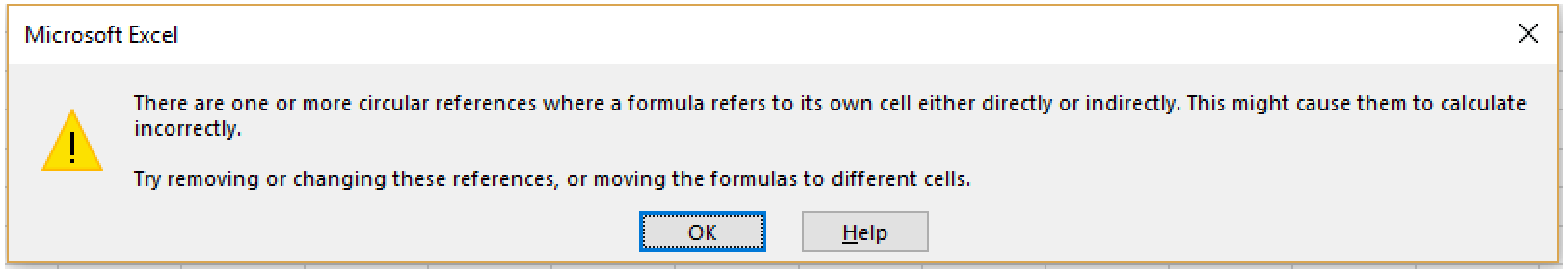




Directed Acyclic Graphs

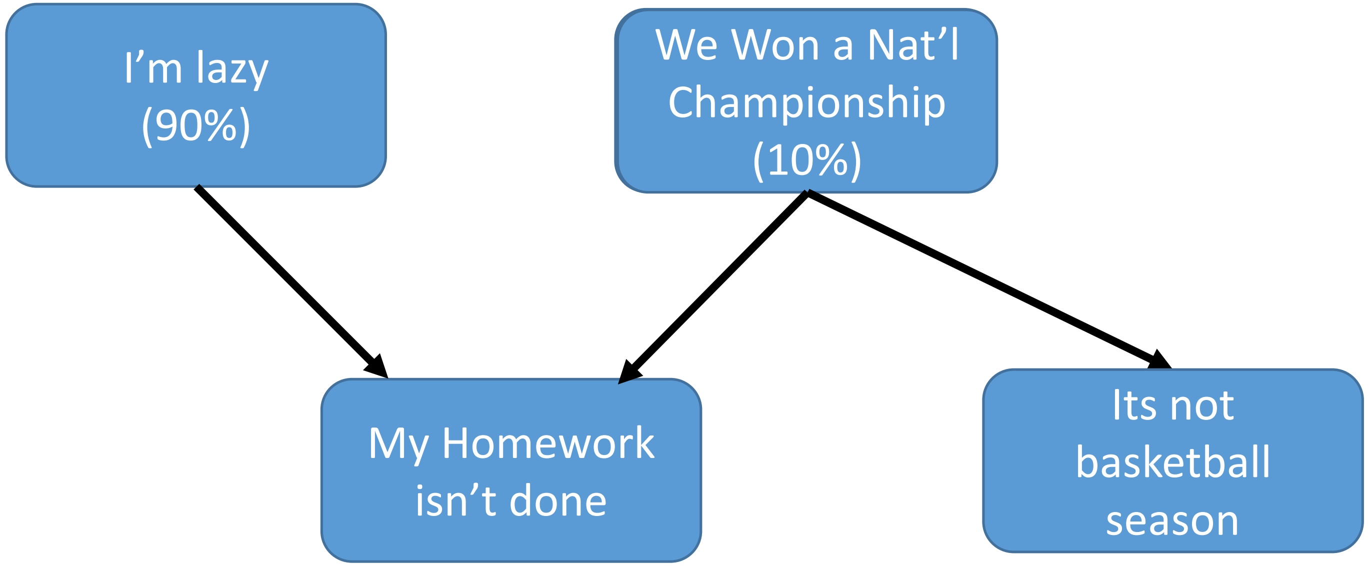


- Directed Acyclic Graphs (DAGs), every edge is directed
- No way to start at a node and end at same node*
- Can represent events, probabilities, and causality.



Belief Networks (Just another name)

- A belief network defines causal relationships in directed acyclic graphs
- Directed Acyclic Graph



- Can explain away certain possibilities with new events

Bayesian Networks

- Formally a probabilistic directed acyclic graph model
- Consists of random variables and their associated probabilities
- Relationships are defined by the specific DAG (ancestry, parents, children, etc…)
- Not necessarily Bayesian…
- Used for Bayesian Inference
- Undirected graph corresponds to Markov Network

An Example

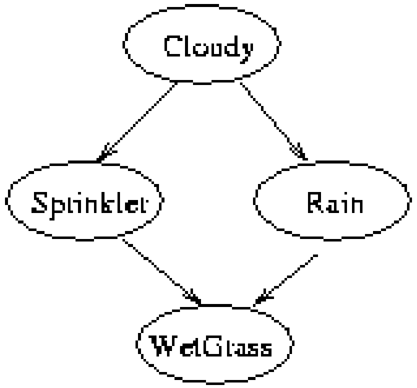
- Each node has a Conditional Probability Distribution (CPD)
- Graph is directed, arrows represent possible causality
- Joint Probability

$$P(C, S, R, W) = P(C) * P(S|C) * P(R|C,S) * P(W|C,S,R)$$

- Simplified (upside to Bayes Nets)

$$P(C, S, R, W) = P(C) * P(S|C) * P(R|C) * P(W|S,R)$$

P(C=F)	P(C=T)
0.5	0.5



C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

An Example (Cont.)

- Inferring information given an event from the joint probability
- i.e. given that the grass is wet…
- Two possible causes, sprinkler is on or its raining.
- Guess that sprinkler = True

$$\Pr(S = 1|W = 1) = \frac{\sum_{c,r} Pr\,(\mathrm{C = c, S = 1, R = r, W = 1})}{\Pr(W = 1)}$$

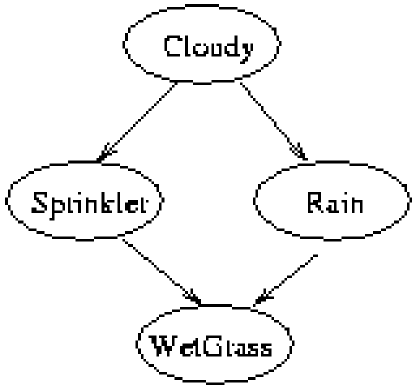
$$= 0.430$$

- Guess that rain = True

$$\Pr(R = 1|W = 1) = \frac{\sum_{c,r} Pr\,(\mathrm{C = c, S = s, R = 1, W = 1})}{\Pr(W = 1)}$$

$$= 0.708$$

P(C=F)	P(C=T)
0.5	0.5



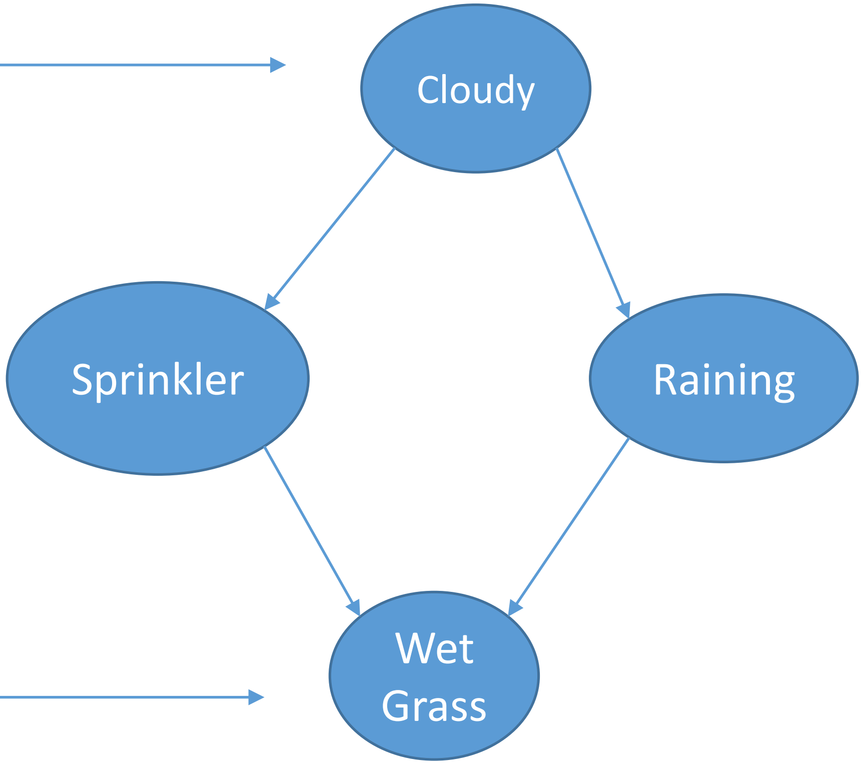
C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

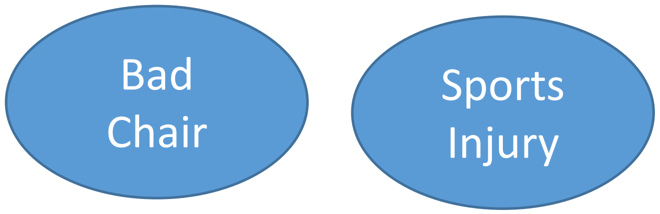
C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

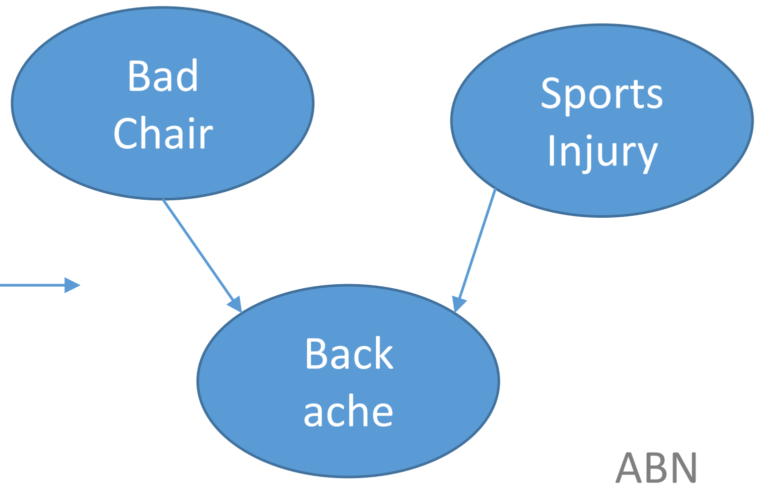
S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

Types of Node

- Unconditional Node
- Conditional
  - Hidden/Latent
  - Evidence





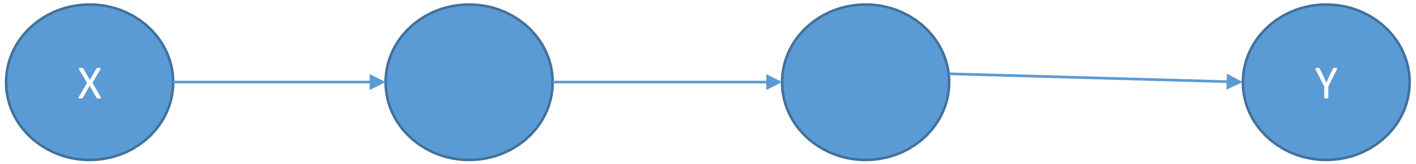


- Dependency
  - Marginally Independent
  - Conditionally Dependent

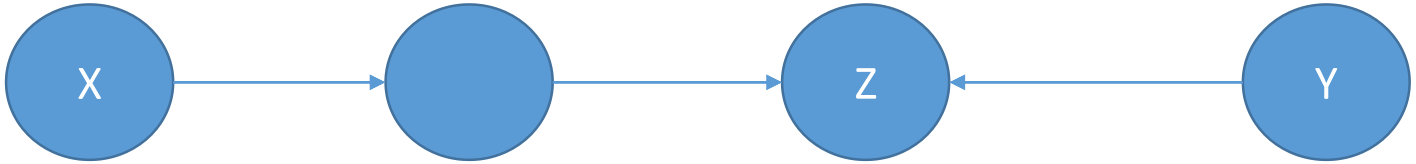
Error and Separation

- More robust to error-variance dilemma
  - Reduction of number of parameters and co-dependence

- d-Separation
  - Two nodes are said to be d-connected if there are no colliders between them



  - Two nodes are d-separated if there is a collider



Two Types of Inferencing

- Top Down
  - Also known as Predictive Support
  - Based on evidence nodes
  - Connections through parents
  - From earlier example: Rain=True, what is Pr(W=T|R=T)?

- Bottom Down
  - Diagnostic Support
  - Based on evidence nodes connected through children
  - In earlier example: Wet Grass=True, what is Pr(S=T|W=T)?

Computational Costs

- Summing/Integrating over the Joint Probability Distribution
  - Goes by O(2^n)
  - Takes exponential time for number of nodes
  - Summing/Integrating of the full JPD is called exact inference
    - NP-Hard Problem
  - Message Passing, MCMC, Loop Belief Propagation

- Instead use Message Passing Algorithm
  - Uses Polytrees (at most, one path between any two nodes)
  - Goes by O(n)
  - Computes marginal likelihood for unobserved nodes
  - Each edge carries the influence that the previous variable has on the next

Bayes Net Learning

- Given training data, causal relationships, etc…
  - Estimate graph topology
  - Estimate parameters of JPD
  - Use one of the four cases

Table 1 Four cases of BN learning problems

Case	BN structure	Observability	Proposed learning method
1	Known	Full	Maximum-likelihood estimation
2	Known	Partial	EM (or gradient ascent), MCMC
3	Unknown	Full	Search through model space
4	Unknown	Partial	EM + search through model space

(1) Known Structure/Fully Observable

- Simplest Case
- Find parameters for Conditional Probability Distributions
  - Must maximize the likelihood of the training set given
- Using Bayesian Method
  - For each node there is a vector of parameters
  - Assign each vector a probability density function
  - Use training data to compute the posterior distribution of parameters

(2) Known Structure/Partially Observable

- Expectation Maximization Algorithm
  - Expectation step creates a function for the likelihood given current parameters (first guess and then iterations)
  - Maximization step computes parameters that maximize the likelihood function predicted by the first step
  - Iterate (can become costly is first parameter guess is poor)

(3) Unknown Structure/Fully Observable

- Trying to find a DAG to represent outcome prob./cause
  - Ends up being an NP-Hard Problem
  - Assume variables are conditionally independent, come from one parent

(4) Unknown Structure/Partially Observable

- Generally intractable, but you can use Bayesian Information Criterion
  - Guessing starting parameters, calculate likelihood of model
  - Adding parameters can lead to overfitting, BIC adds a penalty for each param.

Python Application

- Bayespy package for this purpose
  - Used for Bayesian inference
  - User determines Bayesian network

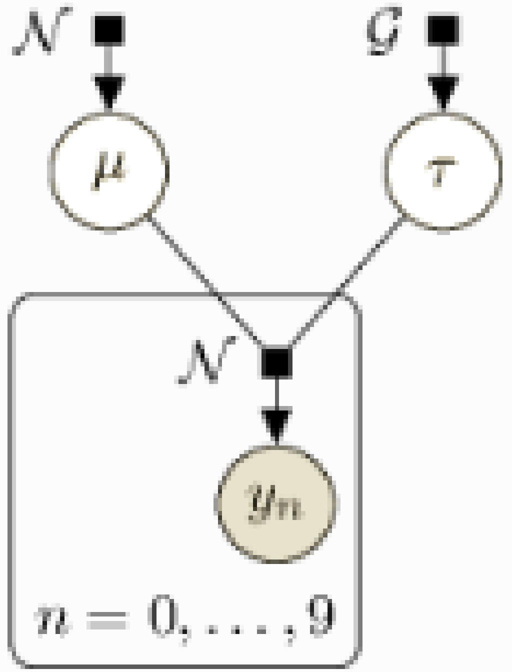
- Example
  - Observations based on Gaussian distribution, unknown mean/variance
  - Wish to determine parameters for model of data

```
>>> import numpy as np
>>> data = np.random.normal(5, 10, size=(10,))
```

Python Example (Cont.)

- Directed Graph of models, parameters, and evidence
- Define models using **bayespy** package

```
>>> from bayespy.nodes import GaussianARD, Gamma
>>> mu = GaussianARD(0, 1e-6)
>>> tau = Gamma(1e-6, 1e-6)
>>> y = GaussianARD(mu, tau, plates=(10,))
```



- GaussianARD and Gamma define nodes
  - More available in bayespy.nodes
  - Plates argument defines number of “plates” directed to by our two nodes

Python Example (Cont.)

- Inference once models are defined
  - Observed data is y, a Gaussian distribution

```
>>> y.observe(data)
```

- Posterior Distribution, calculated using variational Bayesian method
  - Could use MCMC or EP

```
>>> from bayespy.inference import VB
>>> Q = VB(mu, tau, y)
```

Python Example (Cont.)

- Iterate on the inference algorithm (variational Bayesian)

```
>>> Q.update(repeat=20)
Iteration 1: loglike=-6.020956e+01 (... seconds)
Iteration 2: loglike=-5.820527e+01 (... seconds)
Iteration 3: loglike=-5.820290e+01 (... seconds)
Iteration 4: loglike=-5.820288e+01 (... seconds)
Converged at iteration 4.
```

- Examine the output (posterior) as usual
  - Use Histogram
  - Plot marginal probability density functions

Summary

- Bayesian Networks are (strictly) DAGs with local conditional probability
- Used primarily for Bayesian inference, but have further application
  - Neural Networks
  - Hierarchical Networks
  - Hidden Markov Models
- Particularly useful in astronomy
  - Allows for “hand-picking” parametric models
  - Hierarchical modeling allows for several layers of complexion in a given node

References

- Ben-Gal, I. et al., *Bayesian Networks*. (BN)
- Lewicki, Michael S., *AI: Bayes Nets 2*. (ABN)
- Loredo, Tom, *Bayesian Inference in Astronomy*. (BIA)
- Murphy, Kevin, *A Brief Introduction to Graphical Models and Bayesian Networks*. (ABIGM)
- Gyftodimos, E., Flach, P., *Hierarchical Bayesian Networks*. (HBN)
- Bhat, H. S. et al., *On the derivation of the Bayesian Information Criterion*. (ODBI)
- Coughlan, James, *A Tutorial Introduction to Belief Propagation*. (ATIBF)
- Bayespy: http://bayespy.org/en/latest/user_guide/quickstart.html (BP)