Learning in Bayes Nets

Mausam

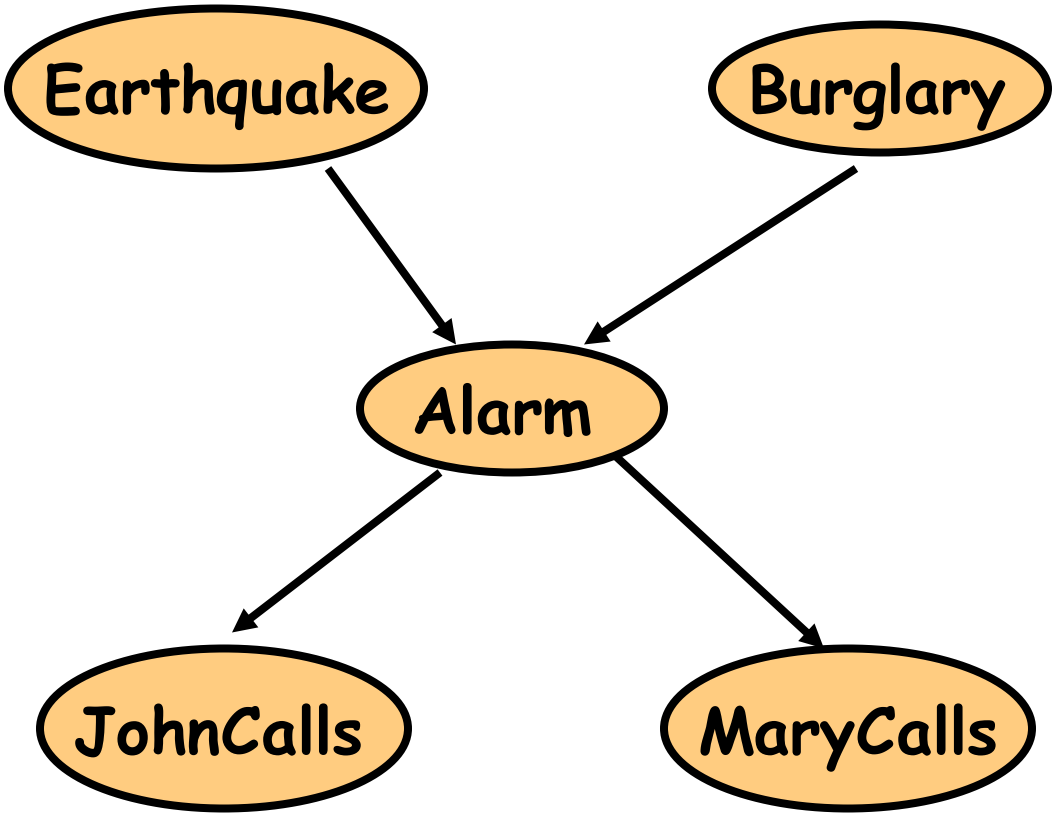
(Based on slides by Stuart Russell, Marie desJardins, Dan Weld)

Parameter Estimation

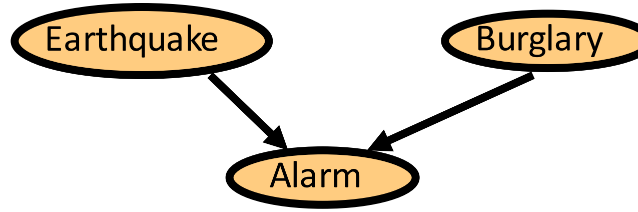
- Learn all the CPTs in a Bayesian Net

- Key idea: counting!

Burglars and Earthquakes



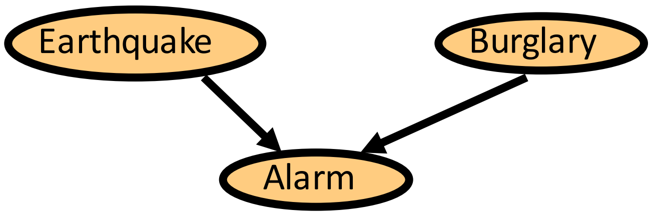
Counting



E	B	A	#
0	0	0	1000
0	0	1	10
0	1	0	20
0	1	1	100
1	0	0	200
1	0	1	50
1	1	0	0
1	1	1	5

	Pr(A|E,B)
e,b	
$e,\bar{b}$	
$\bar{e},b$	
$\bar{e},\bar{b}$	

Counting



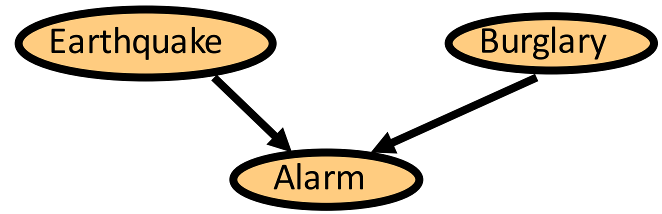
E	B	A	#
0	0	0	1000
0	0	1	10
0	1	0	20
0	1	1	100
1	0	0	200
1	0	1	50
1	1	0	0
1	1	1	5

	Pr(A|E,B)
e,b	
$e,\bar{b}$	
$\bar{e},b$	
$\bar{e},\bar{b}$	

$P(a \mid \bar{e}, \bar{b}) = ?$

$= 10/1010$

Counting



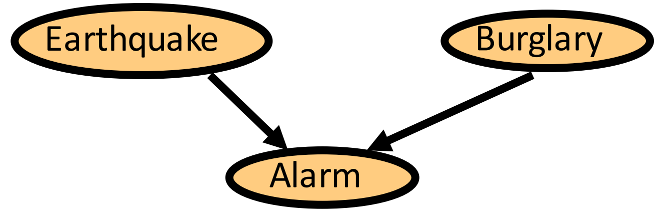
E	B	A	#
0	0	0	1000
0	0	1	10
0	1	0	20
0	1	1	100
1	0	0	200
1	0	1	50
1	1	0	0
1	1	1	5

	Pr(A|E,B)
e,b	
$e,\bar{b}$	
$\bar{e},b$	
$\bar{e},\bar{b}$	~0.01

$P(a \mid \bar{e}, b) = ?$

$= 100/120$

Counting

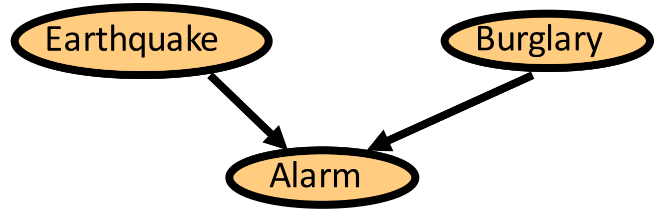


E	B	A	#
0	0	0	1000
0	0	1	10
0	1	0	20
0	1	1	100
1	0	0	200
1	0	1	50
1	1	0	0
1	1	1	5

	Pr(A|E,B)
e,b	
$e,\bar{b}$	
$\bar{e},b$	0.83
$\bar{e},\bar{b}$	~0.01

$P(a \mid e, \bar{b}) = ?$

$= 50/250$



Counting

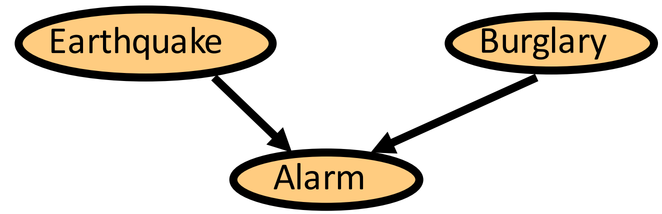
E	B	A	#
0	0	0	1000
0	0	1	10
0	1	0	20
0	1	1	100
1	0	0	200
1	0	1	50
1	1	0	0
1	1	1	5

	Pr(A|E,B)
e, b	
$e, \bar{b}$	0.2
$\bar{e}, b$	0.83
$\bar{e}, \bar{b}$	~0.01

$P(a \mid e, b) = ?$

$= 5/5$

Counting



E	B	A	#
0	0	0	1000
0	0	1	10
0	1	0	20
0	1	1	100
1	0	0	200
1	0	1	50
1	1	0	0
1	1	1	5

	Pr(A|E,B)
e,b	1
$e,\bar{b}$	0.2
$\bar{e},b$	0.83
$\bar{e},\bar{b}$	~0.01

Bad idea to have prob as 0 or 1

- stumps Gibbs sampling
- low prob states become impossible

Solution: Smoothing

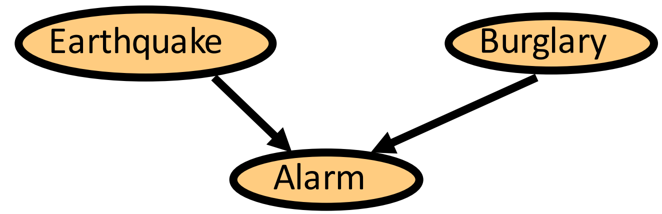
- Why?
  - To deal with events observed zero times.
  - "event": a particular ngram

- How?
  - To shave a little bit of probability mass from the higher counts, and pile it instead on the zero counts

- Laplace Smoothing/Add-one smoothing
  - assume each event was observed at least once.
  - add 1 to all frequency counts

- Add m instead of 1 (m could be > or < 1)

Counting w/ Smoothing



E	B	A	#
0	0	0	1000+1
0	0	1	10+1
0	1	0	20+1
0	1	1	100+1
1	0	0	200+1
1	0	1	50+1
1	1	0	0+1
1	1	1	5+1

	Pr(A|E,B)
e,b	0.86
$e,\bar{b}$	~0.2
$\bar{e},b$	~0.83
$\bar{e},\bar{b}$	~0.01

ML vs. MAP Learning

- ML: maximum likelihood (what we just did)
  - find parameters that maximize the prob of seeing the data D
  - $\text{argmax}_\theta \; P(D|\theta)$
  - easy to compute (for example, just counting)
  - assume uniform prior
- Prior: your belief before seeing any data
  - Uniform prior: all parameters equally likely
- MAP: maximum a posteriori estimate
  - maximize prob of parameters after seeing data D
  - $\text{argmax}_\theta \; P(\theta|D) = \text{argmax}_\theta \; P(D|\theta)P(\theta)$
  - allows user to input additional domain knowledge
  - better parameters when data is sparse…
  - reduces to ML when infinite data

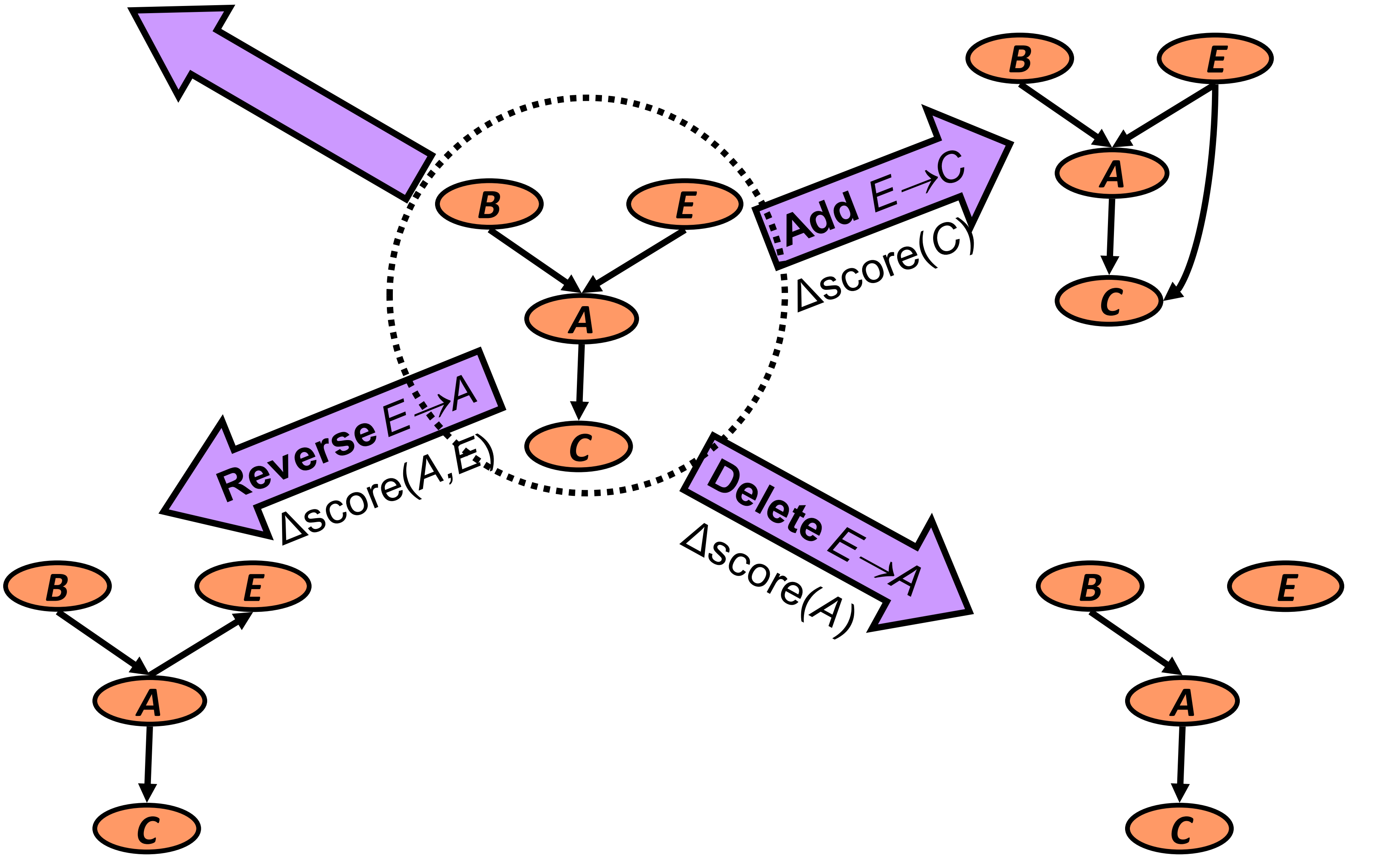
Learning the Structure

- Problem: learn the structure of Bayes nets
- Search thru the space…
  - of possible network structures!
  - Heuristic search/local search
- For each structure, learn parameters
- Pick the one that fits observed data best
  - Caveat – won't we end up fully connected????

When scoring, add a penalty

$\propto$ model complexity

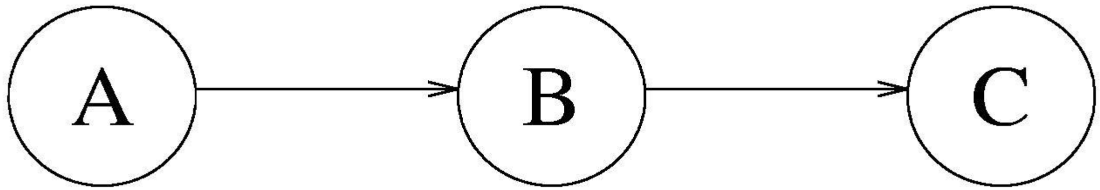
Local Search



How to learn when some data missing?

- Expectation Maximization (EM)

Example



Examples:	0	1	1
	1	0	0
	1	1	1
	1	?	0

Initialization: $P(B|A) =$ $P(C|B) =$

$P(A) =$ $P(B|\neg A) =$ $P(C|\neg B) =$

E-step: $P(? = 1) = P(B|A, \neg C) = \frac{P(A, B, \neg C)}{P(A, \neg C)} = \ldots = 0$

M-step: $P(B|A) =$ $P(C|B) =$

$P(A) =$ $P(B|\neg A) =$ $P(C|\neg B) =$

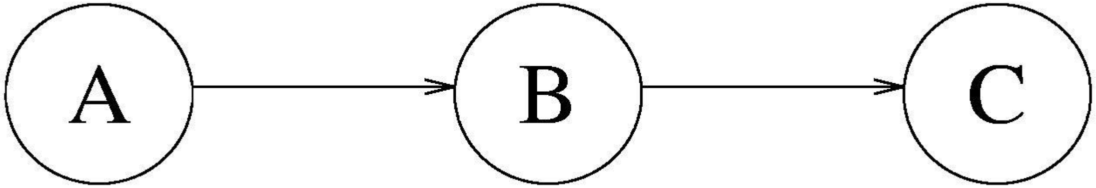
E-step: $P(? = 1) =$

Chicken & Egg Problem

- If we knew the missing value
  - It would be easy to learn CPT
- If we knew the CPT
  - Then it'd be easy to infer the (probability of) missing value
- But we do not know either!

Slide by Daniel S. Weld

Example



Examples:	0	1	1
	1	0	0
	1	1	1
	1	?	0

Initialization: $P(B|A) = 0$ $\quad P(C|B) = 0$

$P(A) = 0.75$ $\quad P(B|\neg A) = 0$ $\quad P(C|\neg B) = 0$

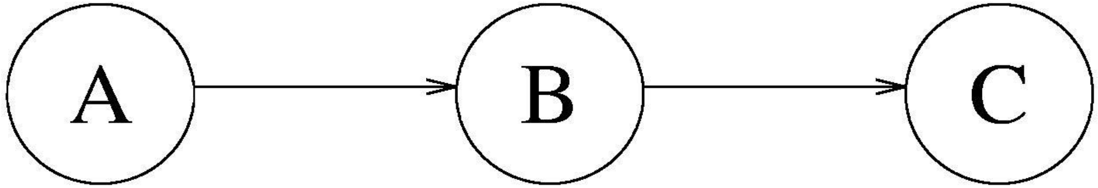
E-step: $P(? = 1) = P(B|A, \neg C) = \frac{P(A,B,\neg C)}{P(A,\neg C)} = \ldots = 0$

M-step: $P(B|A) =$ $\quad P(C|B) =$

$P(A) =$ $\quad P(B|\neg A) =$ $\quad P(C|\neg B) =$

E-step: $P(? = 1) =$

Example



Examples:	0	1	1
	1	0	0
	1	1	1
	1	? 0	0

Initialization: $P(B|A) = 0$ $\quad P(C|B) = 0$

$P(A) = 0.75$ $\quad P(B|\neg A) = 0$ $\quad P(C|\neg B) = 0$

E-step: $P(? = 1) = P(B|A, \neg C) = \frac{P(A,B,\neg C)}{P(A,\neg C)} = \ldots = 0$

M-step: $P(B|A) = 0.33$ $\quad P(C|B) = 1$

$P(A) = 0.75$ $\quad P(B|\neg A) = 1$ $\quad P(C|\neg B) = 0$

E-step: $P(? = 1) =$

Expectation Maximization

- **Guess** probabilities for nodes with **missing values** (e.g., based on other observations)
- **Compute the probability distribution** over the missing values, given our guess
- **Update the probabilities** based on the guessed values
- **Repeat** until convergence

- Guaranteed to converge to local optimum

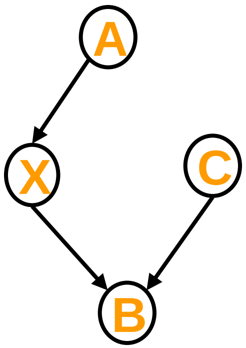
Learning Summary

- **Known structure, fully observable**: only need to do parameter estimation
- **Unknown structure, fully observable:** do heuristic/local search through structure space, then parameter estimation
- **Known structure, missing values:** use expectation maximization (EM) to estimate parameters

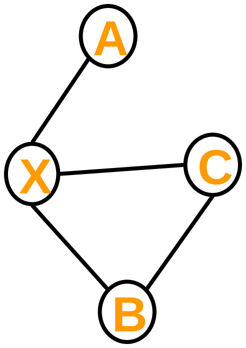
- **Known structure, hidden variables:** apply adaptive probabilistic network (APN) techniques
- **Unknown structure, hidden variables:** too hard to solve!

Other Graphical Models

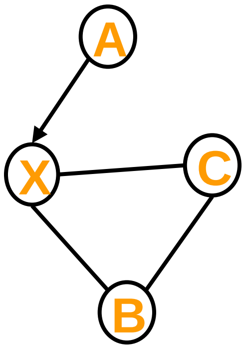
- Directed
  - Bayesian Networks



- Undirected
  - Markov Network (Markov Random Field)
  - BN → MN (moralization: marry all co-parents)



- Mixed
  - Chain Graph



Other Graphical Models