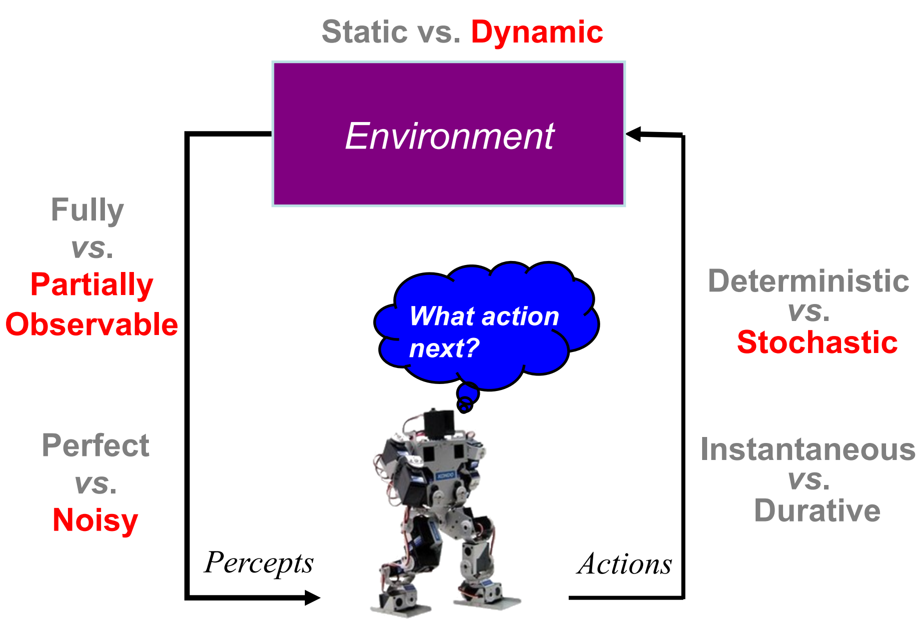
CSE 473: Artificial Intelligence

Bayesian Networks - Learning

Dan Weld

Slides adapted from Jack Breese, Dan Klein, Daphne Koller, Stuart Russell, Andrew Moore & Luke Zettlemoyer



AI Topics

- Search
  - Problem Spaces
  - BFS, DFS, UCS, A* (tree and graph)
  - Completeness and Optimality
  - Heuristics: admissibility and consistency
- CSPs
  - Constraint graphs, backtracking search
  - Forward checking, AC3 constraint propagation, ordering heuristics
- Games
  - Minimax, Alpha-beta pruning, Expectimax, Evaluation Functions
- MDPs
  - Bellman equations
  - Value iteration & policy iteration
  - RTDP, LAO* & UCT
  - POMDPs
- Reinforcement Learning
  - Exploration *vs.* Exploitation
  - Model-based *vs.* model-free
  - Q-learning
  - Linear value function approx.
- Hidden Markov Models
  - Markov chains
  - Forward algorithm
  - Particle Filter
- Bayesian Networks
  - Basic definition, independence (d-sep)
  - Variable elimination
  - Gibbs sampling
- Learning
  - BN parameters with data complete & incomplete (Expectation Maximization)
  - Search thru space of BN structures

Search thru a Problem Space / State Space

Ex. Proving a trig identity, e.g. $\sin^2(x) = \frac{1}{2} - \frac{1}{2}\cos(2x)$

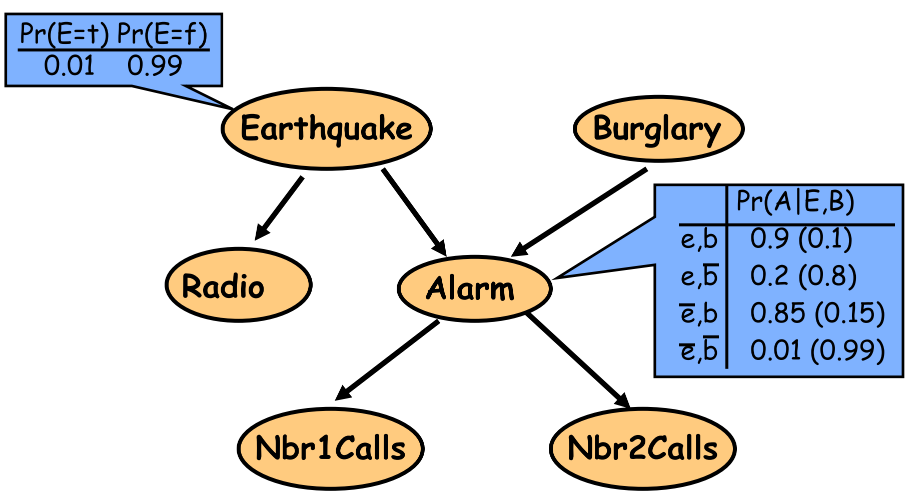
- Input:
  - Set of states
  - Operators [and costs]
  - Start state
  - Goal state [test]
- Output:
  - Path: start $\Rightarrow$ a state satisfying goal test
  - [May require shortest path]
  - [Sometimes just need state passing test]

Today

- Bonus Topic – Hybrid Bayes Nets
- Learning
  - Parameter Learning & Priors
  - Expectation Maximization
  - Structure Learning

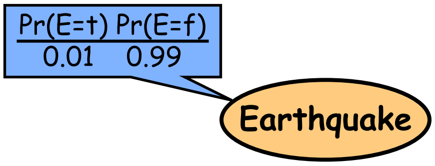
5

Bayes Nets

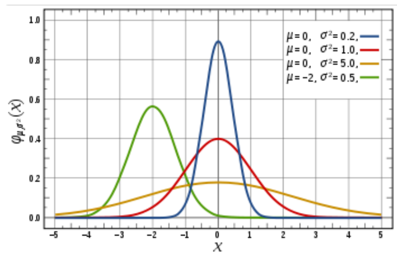


 6

Continuous Variables

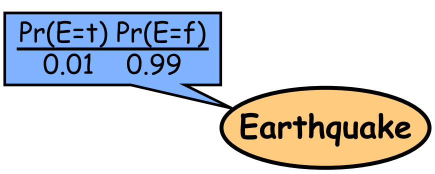


So far: assuming variables have discrete values
Could also allow continuous values, E $\in \mathbb{R}$,
How specify probabilities? (explicit CPT would be infinitely large)

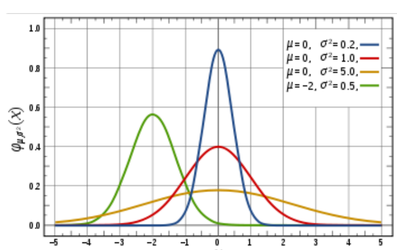


$$P(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Continuous Variables

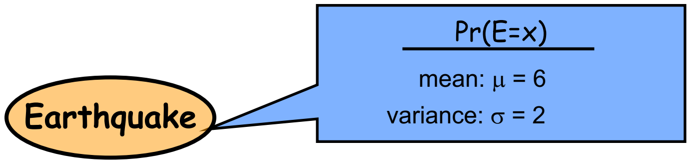


So far: assuming variables have discrete values
Could also allow continuous values, E $\in \mathbb{R}$,
And specify probabilities using a continuous distribution, such as a Gaussian

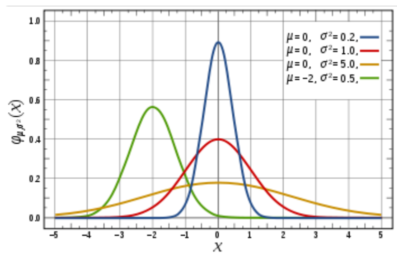


$$P(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Continuous Variables

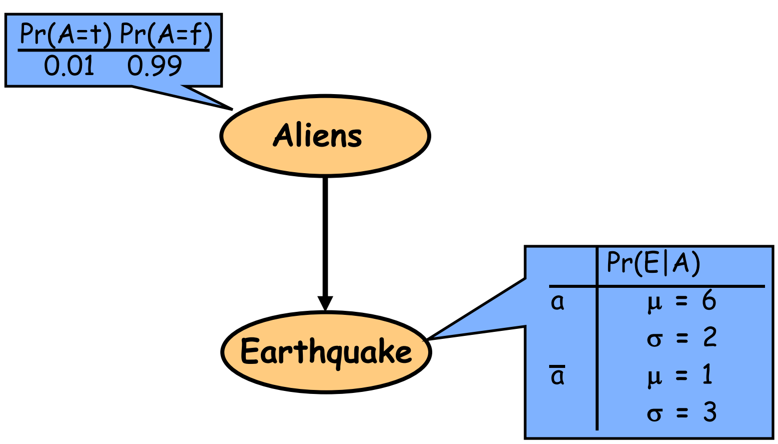


So far: assuming variables have discrete values
Could also allow continuous values, E $\in$ R,
And specify probabilities using a continuous distribution, such as a Gaussian

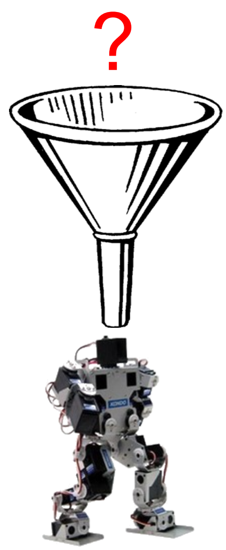


$$P(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Continuous Variables



Learning



Supremacy of Machine Learning

- Machine learning is preferred approach to
  - Speech recognition, Natural language processing
  - Web search – result ranking
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - Computational biology
  - Sensor networks
  - …
- This trend is accelerating
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for self-customization to user, environment

17

What is Machine Learning ?

Data → Machine Learning → **Understanding**

Machine Learning

Study of algorithms that

- improve their <u>performance</u>
- at some <u>task</u>
- with <u>experience</u>

Ability to accumulate reward

Executing actions

Executing actions

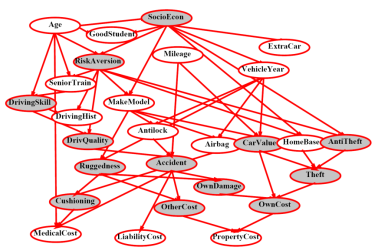
?? Reinforcement Learning ??

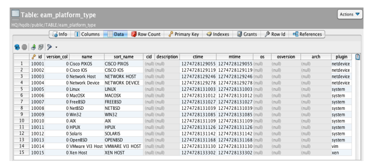
19

Machine Learning

Study of algorithms that

- improve their <u>performance</u> Prediction accuracy
- at some <u>task</u> Answering probabilistic queries
- with <u>experience</u> Seeing labeled data



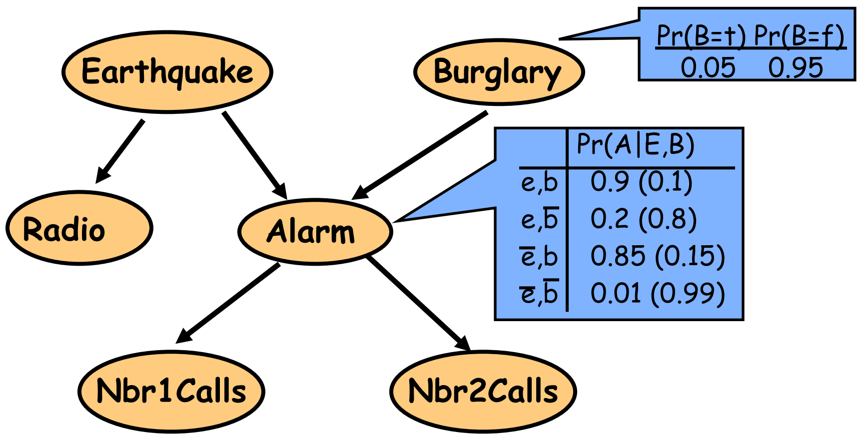


20

Learning Bayes Networks

- Learning Parameters for a Bayesian Network
  - Fully observable variables
    - Maximum Likelihood (ML), MAP & Bayesian estimation
    - Example: Naïve Bayes for text classification
  - Hidden variables
    - Expectation Maximization (EM)
- Learning Structure of Bayesian Networks

The Origin of Bayes Nets



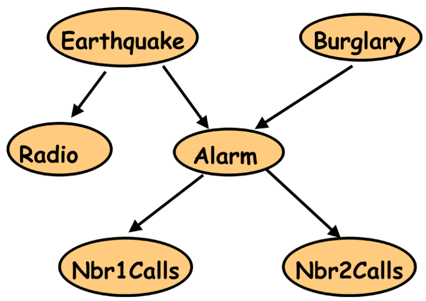
 22

Learning Bayes Nets

Suppose …

1. Know structure & get complete observations of every var
2. Know structure & get observations only of ***some*** vars
   Others are hidden (learn with EM)
3. Don't even know structure!

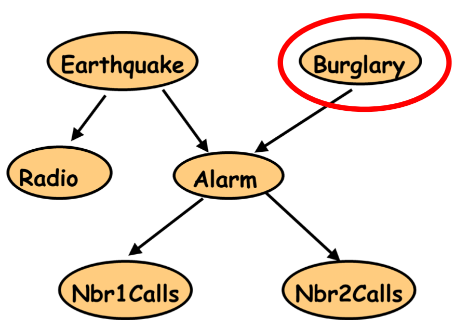
Parameter Estimation and Bayesian Networks



E	B	R	A	J	M
T	F	T	T	F	T
F	F	F	F	F	T
F	T	F	T	T	T
F	F	F	T	T	T
F	T	F	F	F	F
...					

We have:
- Bayes Net structure and observations
- We need: Bayes Net parameters

Parameter Estimation and Bayesian Networks

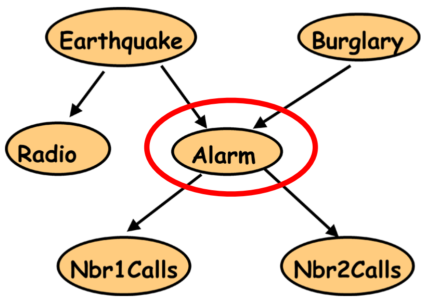


B
F
F
T
F
T

P(B) = ? = 0.4

P(¬B) = 1- P(B) = 0.6

Parameter Estimation and Bayesian Networks

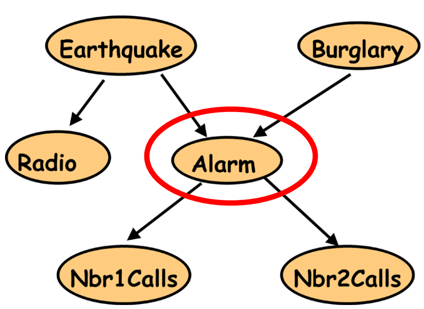


E	B
T	F
F	F
F	T
F	F
F	T
…	

A
T
F
T
T
F

P(A|E,B) = ?
P(A|E,¬B) = ?
P(A|¬E,B) = ?
P(A|¬E,¬B) = 0.5

Parameter Estimation and Bayesian Networks

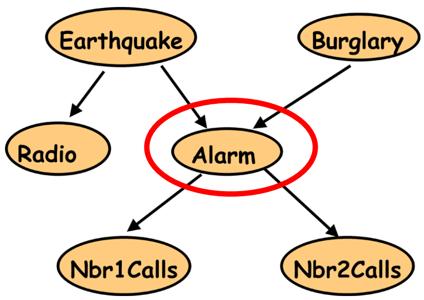


E	B
T	F
F	F
F	T
F	F
F	T
…	

A
T
F
T
T
F

P(A|E,B) = ?
P(A|E,¬B) = 1.0 ?
P(A|¬E,B) = ?
P(A|¬E,¬B) = ?

Parameter Estimation and Bayesian Networks



E	B	A
T	F	T
F	F	F
F	T	T
F	F	T
F	T	F
...		

P(A|E,B) = ?

P(A|E,¬B) = ?

P(A|¬E,B) = ?

P(A|¬E,¬B) = ?

Parameter Estimation and Bayesian Networks

Coin Flip

C_1

$P(H|C_1) = 0.1$

C_2

$P(H|C_2) = 0.5$

C_3

$P(H|C_3) = 0.9$

Which coin will I use?

$P(C_1) = 1/3$ $P(C_2) = 1/3$ $P(C_3) = 1/3$

Prior: Probability of a hypothesis before we make any observations

Coin Flip

C_1

$P(H|C_1) = 0.1$

C_2

$P(H|C_2) = 0.5$

C_3

$P(H|C_3) = 0.9$

Which coin will I use?

$P(C_1) = 1/3$ $P(C_2) = 1/3$ $P(C_3) = 1/3$

Uniform Prior: All hypothesis are equally likely before we make any observations

Experiment 1: Heads

Which coin did I use?

$P(C_1|H) = ?$ $P(C_2|H) = ?$ $P(C_3|H) = ?$

$$P(C_1|H) = \frac{\boxed{P(H|C_1)}\boxed{P(C_1)}}{\boxed{P(H)}}$$

$$\boxed{P(H) = \sum_{i=1}^{3} P(H|C_i)P(C_i)}$$

C_1	C_2	C_3
		
$\boxed{P(H\|C_1)=0.1}$	$P(H\|C_2) = 0.5$	$P(H\|C_3) = 0.9$
$\boxed{P(C_1)=1/3}$	$P(C_2) = 1/3$	$P(C_3) = 1/3$

Experiment 1: Heads

Which coin did I use?

$P(C_1|H) = 0.066$ $P(C_2|H) = 0.333$ $P(C_3|H) = 0.6$

Posterior: Probability of a hypothesis given data

C_1	C_2	C_3
		
$P(H\|C_1) = 0.1$	$P(H\|C_2) = 0.5$	$P(H\|C_3) = 0.9$
$P(C_1) = 1/3$	$P(C_2) = 1/3$	$P(C_3) = 1/3$

Using Prior Knowledge

- Should we always use a ***Uniform Prior*** ?
- Background knowledge:

Heads => we have to buy Dan chocolate

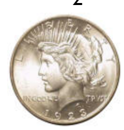
Dan ***likes*** chocolate…

=> Dan is more likely to use a coin biased in his favor

C_1

$P(H|C_1) = 0.1$

C_2



$P(H|C_2) = 0.5$

C_3

$P(H|C_3) = 0.9$

Using Background Knowledge

We can encode it in the prior:

$P(C_1) = 0.05$ $P(C_2) = 0.25$ $P(C_3) = 0.70$

C_1

$P(H|C_1) = 0.1$

C_2

$P(H|C_2) = 0.5$

C_3

$P(H|C_3) = 0.9$

Experiment 1: Heads

Which coin did I use?

$P(C_1|H) = 0.006$ $P(C_2|H) = 0.165$ $P(C_3|H) = 0.829$

Compare with ML posterior after Exp 1:

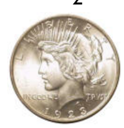
$P(C_1|H) = 0.066$ $P(C_2|H) = 0.333$ $P(C_3|H) = 0.600$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 0.05$

C_2



$P(H|C_2) = 0.5$

$P(C_2) = 0.25$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 0.70$

Probabilistic Estimation

Easy to compute

	Prior	Hypothesis
Maximum Likelihood Estimate (MLE)	Uniform	The most likely
Maximum A Posteriori Estimate (MAP)	Any	The most likely
Bayesian Estimate	Any	Weighted combination

Still easy to compute
Incorporates prior knowledge

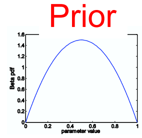
Minimizes error
Great when data is scarce
Potentially much harder to compute

Bayesian Learning

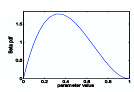
Use Bayes rule:

Data Likelihood

Prior



Posterior



$$P(Y \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid Y)\, P(Y)}{P(\mathbf{X})}$$

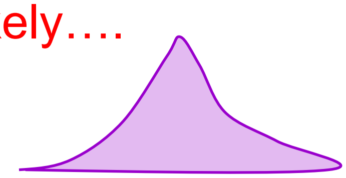
Normalization

Or equivalently: $P(Y \mid \mathbf{X}) \propto P(\mathbf{X} \mid Y)\, P(Y)$

Really? Only 3 Coins?

C_1	C_2	C_3
		
$P(H \mid C_1) = 0.1$	$P(H \mid C_2) = 0.5$	$P(H \mid C_3) = 0.9$

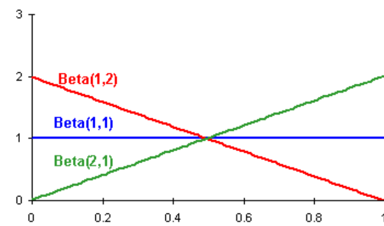
More Likely….



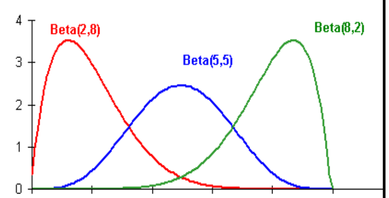
0.1 0.5 0.9

What Prior to Use?

- Two common priors for ***continuous variables***
  - Binary variable Beta
    - Posterior distribution is binomial
    - Easy to compute posterior
    - Easy to compute MAP estimate
      - MAP E[Beta(a, b)] = a/(a+b)



  - Discrete variable Dirichlet
    - Posterior distribution is multinomial
    - Easy to compute posterior



 41

Estimation: Laplace Smoothing

- Laplace's estimate:

  pretend you saw every outcome once more than you actually did

H H T

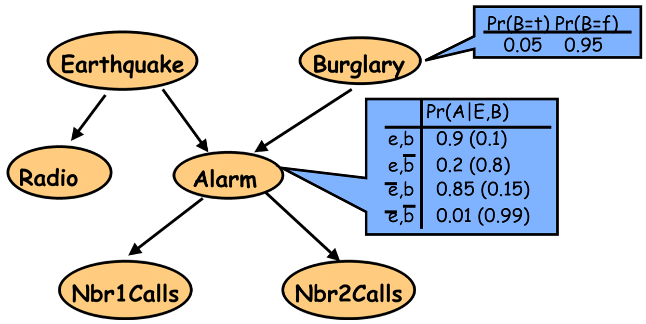
$$P_{LAP}(x) = \frac{c(x)+1}{\sum_x [c(x)+1]}$$

$$= \frac{c(x)+1}{N+|X|}$$

P_{LAP}(H) = (2+1) / (3+2)
= 3/5

Another name for computing the MAP estimate with *Dirichlet priors*
(Bayesian justification)

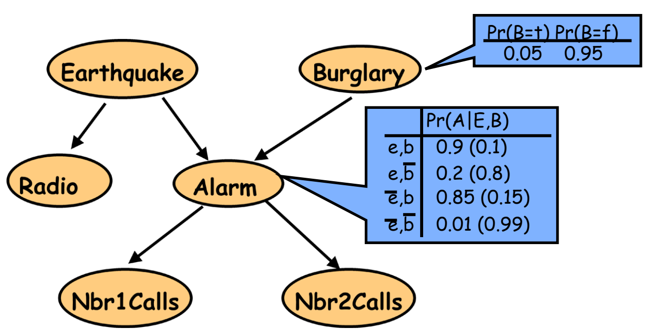
Output of Learning



E	B	R	A	J	M
T	F	T	T	F	T
F	F	F	F	F	T
F	T	F	T	T	T
F	F	F	T	T	T
F	T	F	F	F	F
...					

Prior

Did Learning Work Well?



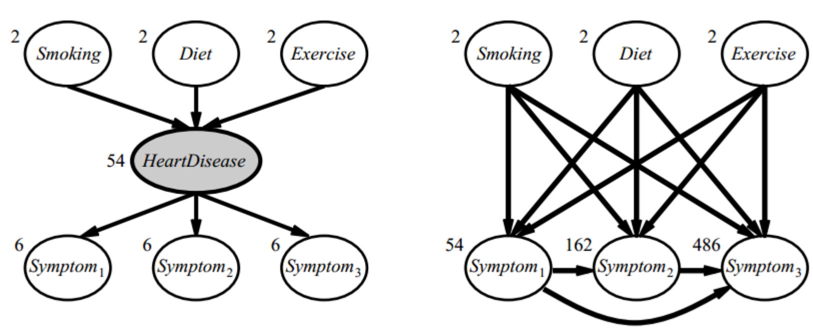
E	B	R	A	J	M
T	F	T	T	F	T
F	F	F	F	F	T
F	T	F	T	T	T
F	F	F	T	T	T
F	T	F	F	F	F
...					

Can easily calculate
P(data) for learned parameters

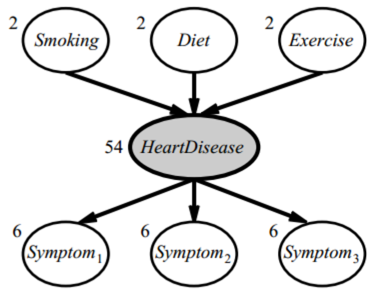
Topics

- Another Useful Bayes Net
  - Hybrid Discrete / Continuous
- Learning Parameters for a Bayesian Network
  - Fully observable
  - Hidden variables (EM algorithm)
- Learning Structure of Bayesian Networks

Why Learn Hidden Variables?

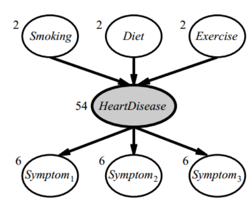


How Learn Hidden Variables?



Chicken & Egg Problem

- If we knew whether patient had disease
  - It would be easy to learn CPTs
  - But we can't observe states, so we don't!



- If we knew CPTs
  - It would be easy to predict if patient had disease
  - But we don't, so we can't!

Slide by Daniel S. Weld

Face It…

58

Continuous Variables

Pr(A=t)	Pr(A=f)
0.01	0.99

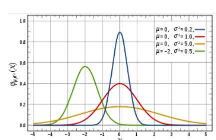
hidden

Aliens

Earthquake

	Pr(E|A)
a	μ = 6
	σ = 2
$\bar{a}$	μ = 1
	σ = 3

Learning with Continuous Variables



Earthquake

Pr(E=x)

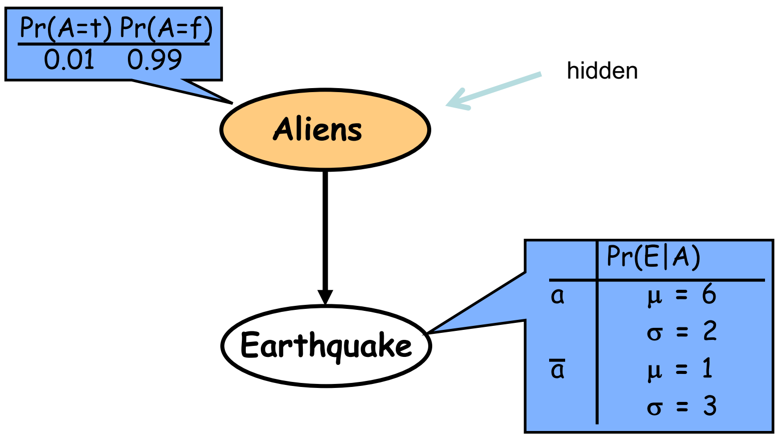
mean: μ = ?

variance: σ = ?

$$\widehat{\mu}_{MLE} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

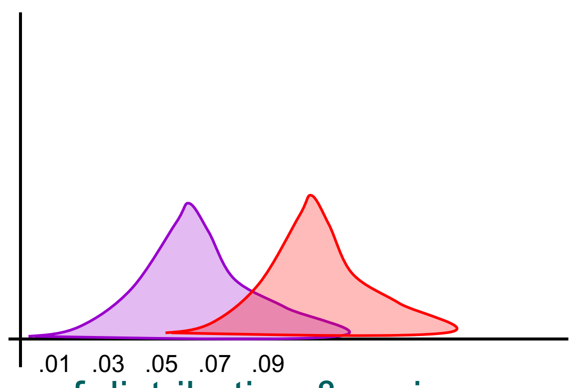
$$\widehat{\sigma}^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N} (x_i - \widehat{\mu})^2$$

Continuous Variables



Simplest Version

- Mixture of two distributions

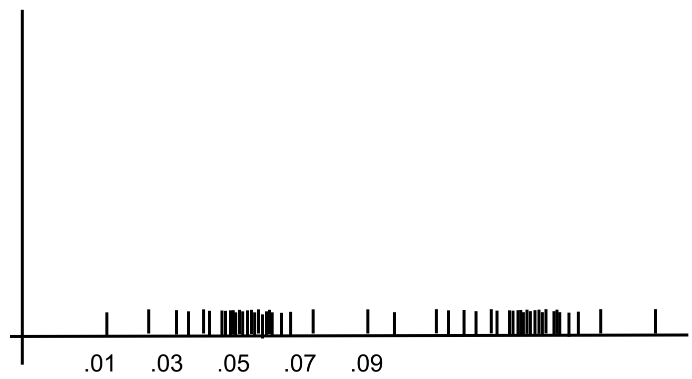


- Know: form of distribution & variance, $\sigma = .5$
- Just need *mean* of each distribution

Slide by Daniel S. Weld

63

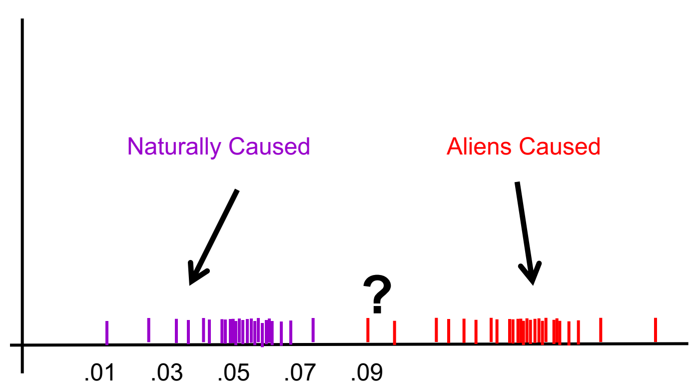
Input Looks Like



Slide by Daniel S. Weld

64

We Want to Predict

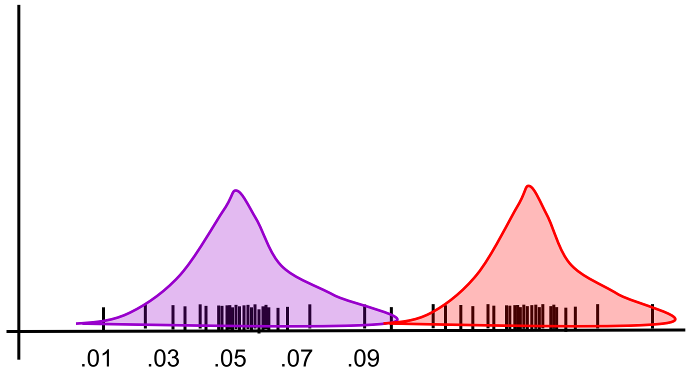


Slide by Daniel S. Weld

65

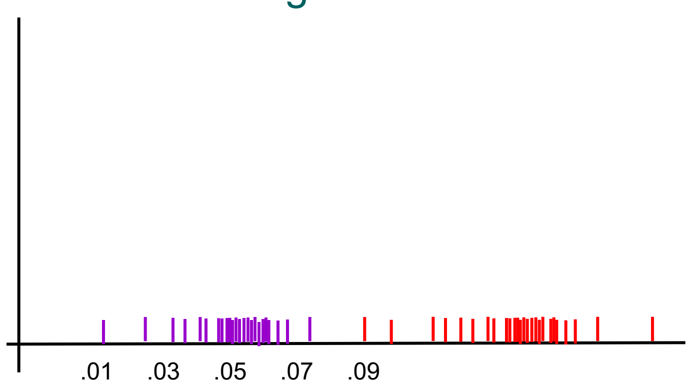
Chicken & Egg

Note that coloring instances would be easy *if* we knew Gausians….



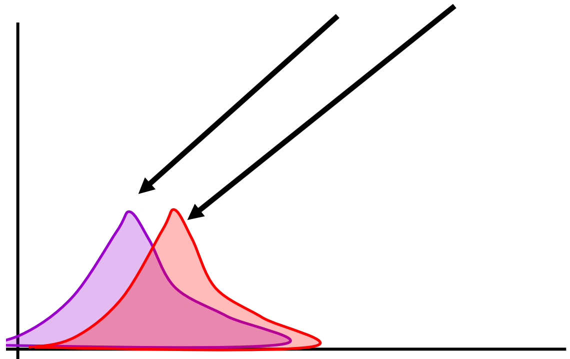
Chicken & Egg

And finding Gausian parameters would be easy *If* we knew the coloring



Expectation Maximization (EM)

- Pretend we *do* know the parameters
  - Initialize randomly: set θ_1=?; θ_2=?

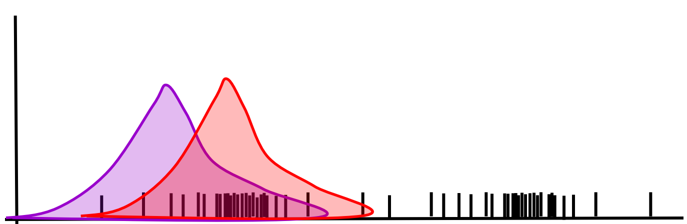


Slide by Daniel S. Weld

68

Expectation Maximization (EM)

- Pretend we *do* know the parameters
  - Initialize randomly
- [E step] Compute probability of instance having each possible value of the hidden variable

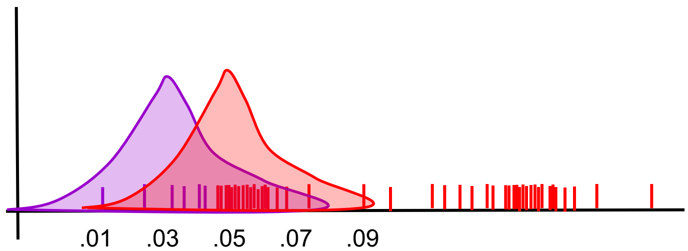


Slide by Daniel S. Weld

69

Expectation Maximization (EM)

- Pretend we *do* know the parameters
  - Initialize randomly
- [E step] Compute probability of instance having each possible value of the hidden variable



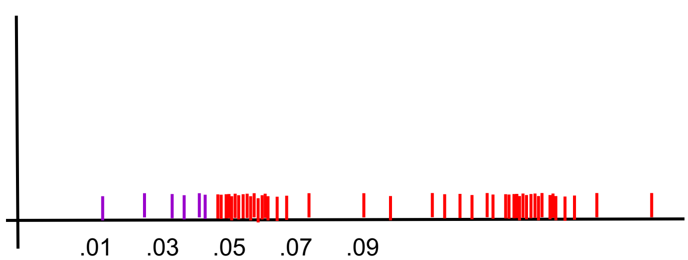
Slide by Daniel S. Weld

70

Expectation Maximization (EM)

- Pretend we *do* know the parameters
  - Initialize randomly
- [E step] Compute probability of instance having each possible value of the hidden variable

[M step] Treating each instance as ***fractionally*** having **both** values compute the new parameter values

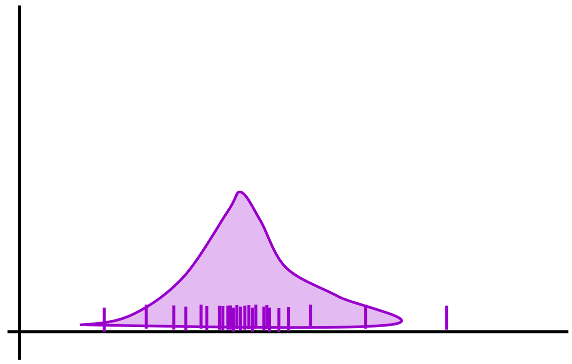


Slide by Daniel S. Weld

71

ML Mean of Single Gaussian

$U_{ml} = \text{argmin}_u \sum_i (x_i - u)^2$



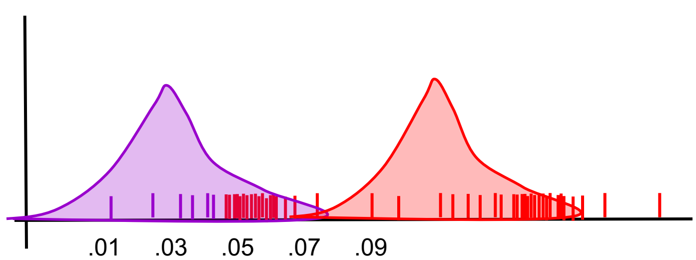
Slide by Daniel S. Weld

72

Expectation Maximization (EM)

-

[M step] Treating each instance as fractionally having **both** values compute the new parameter values

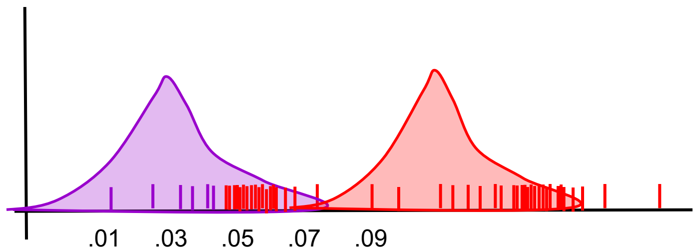


Slide by Daniel S. Weld

73

Expectation Maximization (EM)

- [E step] Compute probability of instance having each possible value of the hidden variable



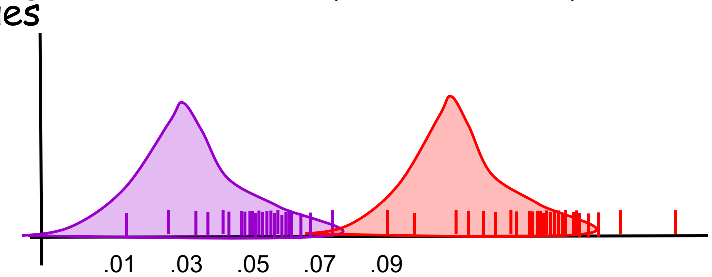
Slide by Daniel S. Weld

74

Expectation Maximization (EM)

- [E step] Compute probability of instance having each possible value of the hidden variable

[M step] Treating each instance as fractionally having both values compute the new parameter values



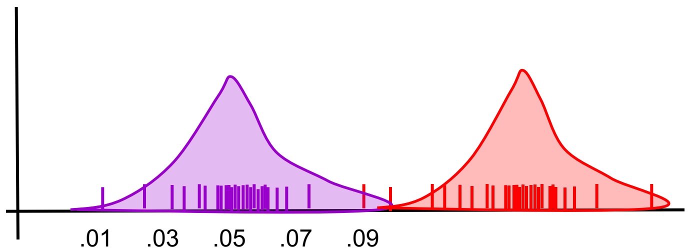
Slide by Daniel S. Weld

75

Expectation Maximization (EM)

- [E step] Compute probability of instance having each possible value of the hidden variable

[M step] Treating each instance as fractionally having both values compute the new parameter values



Slide by Daniel S. Weld

76

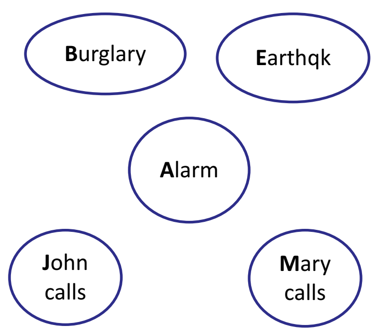
Topics

- Another Useful Bayes Net
  - Hybrid Discrete / Continuous
- Learning Parameters for a Bayesian Network
  - Fully observable
    - Maximum Likelihood (ML),
    - Maximum A Posteriori (MAP)
  - Hidden variables (EM algorithm)
- Learning Structure of Bayesian Networks

What if we ***don't*** know structure?

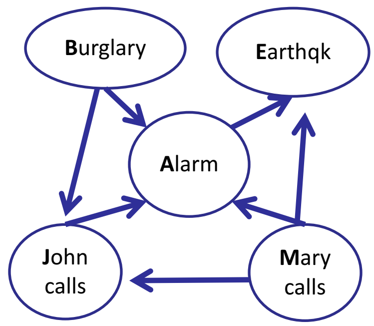
Learning The Structure of Bayesian Networks

E	B	R	A	J	M
T	F	T	T	F	T
F	F	F	F	F	T
F	T	F	T	T	T
F	F	F	T	T	T
F	T	F	F	F	F
…					



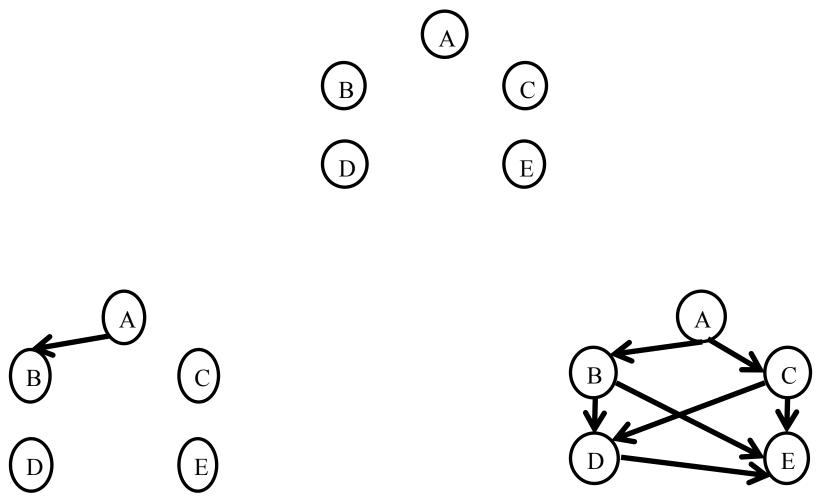
Learning The Structure of Bayesian Networks

E	B	R	A	J	M
T	F	T	T	F	T
F	F	F	F	F	T
F	T	F	T	T	T
F	F	F	T	T	T
F	T	F	F	F	F
…					



Learning The Structure of Bayesian Networks

- Search thru the space…
  - of possible network structures!
- For each structure, learn parameters
  - As just shown…
- Pick the one that fits observed data best
  - Calculate P(data)



Two problems:

- Fully connected will be most probable
- Exponential number of structures

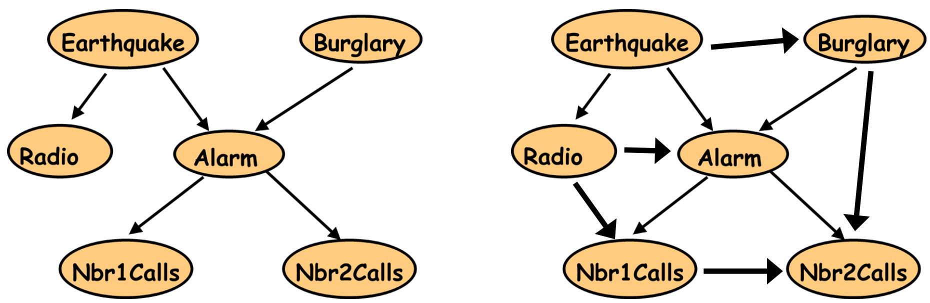
Learning The Structure of Bayesian Networks

- Search thru the space…
  - of possible network structures!
- For each structure, learn parameters
  - As just shown…
- Pick the one that fits observed data best
  - Calculate P(data)

Two problems:

- Fully connected will be most probable
  - Add penalty term (regularization) $\propto$ model complexity
- Exponential number of structures
  - Local search

Overfitting



Can represent strictly more P distributions

Can represent NOISE in training data

Often preforms WORSE on test data

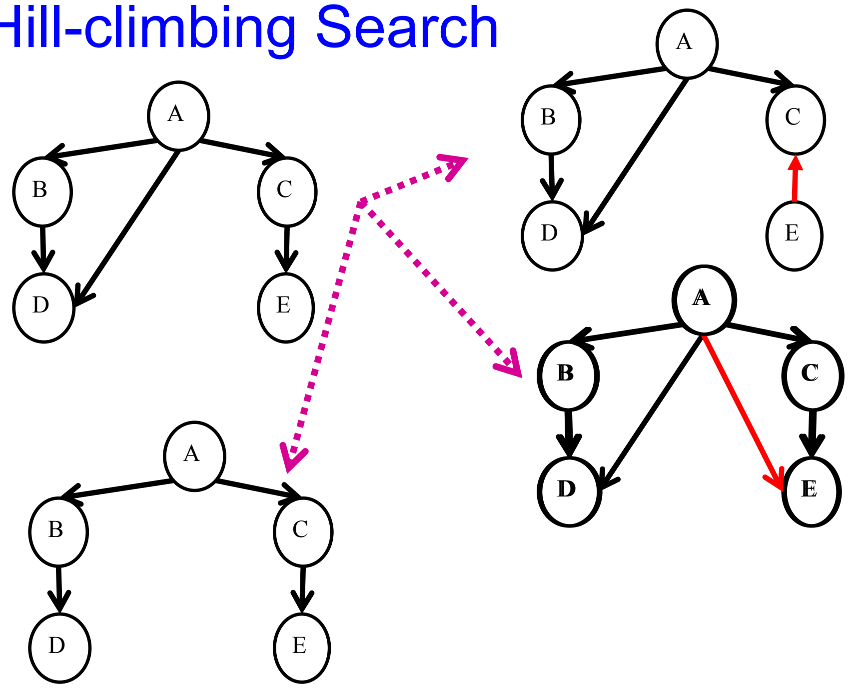
Augment Score Function

- Bayesian Information Criterion (BIC)
  - P(D | BN) – penalty
  - Penalty = α complexity
  - = α [½ (# parameters) Log (# data points)]

Instance of "***regularization***"

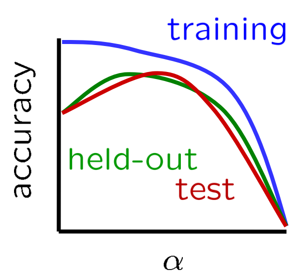
Solves problem of "***overfitting***"

Hill-climbing Search



Tuning on Held-Out Data

- Now we've got two kinds of unknowns
  - Parameters: the probabilities P(Y|X), P(Y)
  - Hyperparameters, like
    - the amount of smoothing to do: k, or
    - regularization penalty, α
- Where to learn?
  - Learn parameters from training data
  - Must tune hyperparameters on different data
    - Why?
  - For each value of the hyperparameters, train and test on the held-out data
  - Choose the best value and do a final test on the test data



Baselines

- First step: get a baseline
  - Baselines are very simple “straw man” procedures
  - Help determine how hard the task is
  - Help know what a “good” accuracy is

- Weak baseline: most frequent label classifier
  - Gives all test instances whatever label was most common in the training set
  - E.g. for spam filtering, might label everything as ham
  - Accuracy might be very high if the problem is skewed
  - E.g. calling everything “spam” gets 86%, so a classifier that gets 90% isn’t very good…

- For real research, usually use previous work as a (strong) baseline