Learning

LESSON 13-14

Reading

Chapter 18

Chapter 20

Outline

1. Inductive Learning
   - Learning agents
   - Inductive learning
   - Decision tree learning
2. Statistical Learning
   - Parameter Estimation:
     - Maximum Likelihood (ML); Maximum A Posteriori (MAP); Bayesian; Continuous case
   - Learning Parameters for a Bayesian Network
   - Naive Bayes
     - Maximum Likelihood estimates; Priors
   - Learning Structure of Bayesian Networks

Learning

Learning is essential for unknown environments,

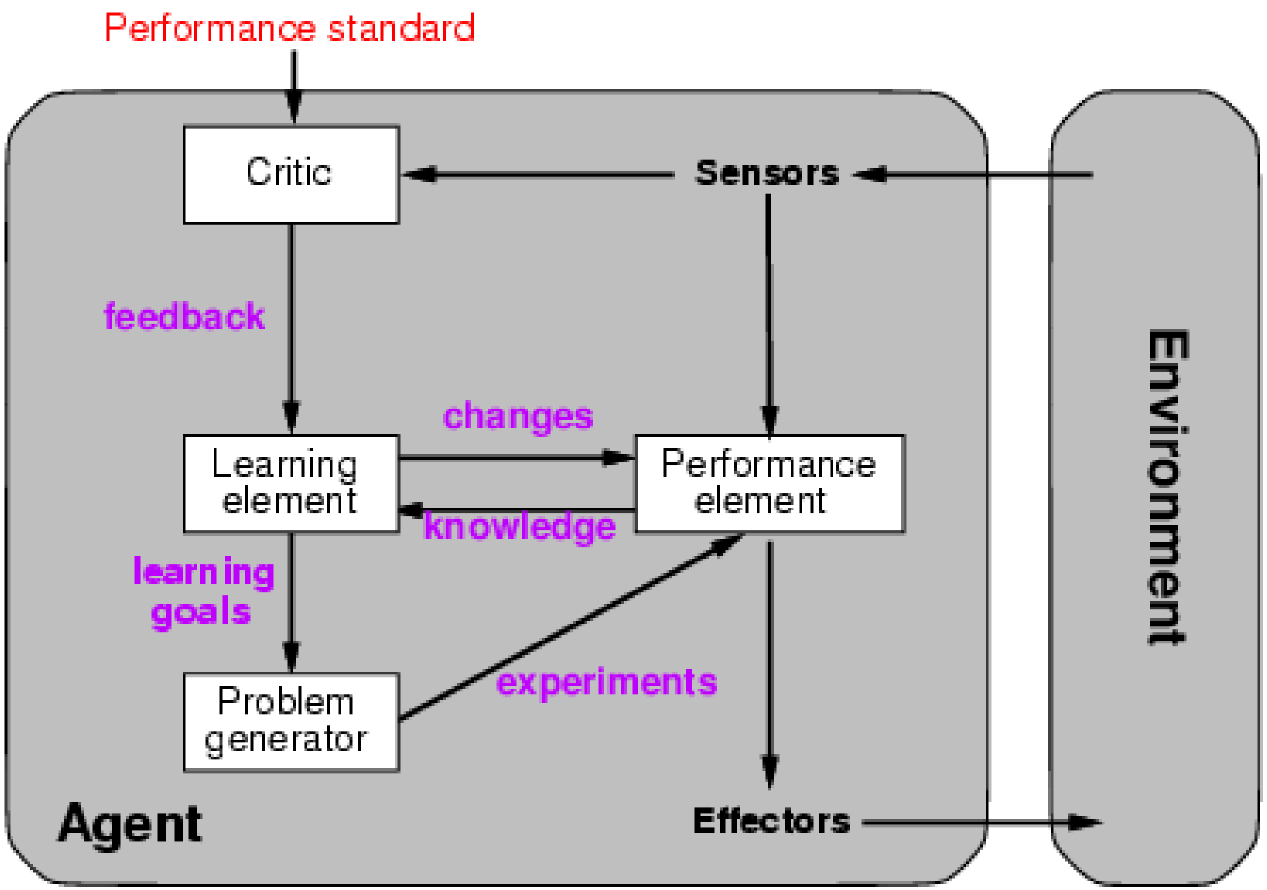
- i.e., when designer lacks omniscience

Learning is useful as a system construction method,

- i.e., expose the agent to reality rather than trying to write it down

Learning modifies the agent's decision mechanisms to improve performance

Learning agents



Learning element

Design of a learning element is affected by

- Which components of the performance element are to be learned
- What feedback is available to learn these components
- What representation is used for the components

Type of feedback:

- Supervised learning: correct answers for each example
- Unsupervised learning: correct answers not given
- Reinforcement learning: occasional rewards

Inductive learning

Simplest form: learn a function from examples

f is the target function

An example is a pair (*x*, *f(x)*)

Problem: find a hypothesis *h*

such that $h \approx f$

given a training set of examples

(This is a highly simplified model of real learning:

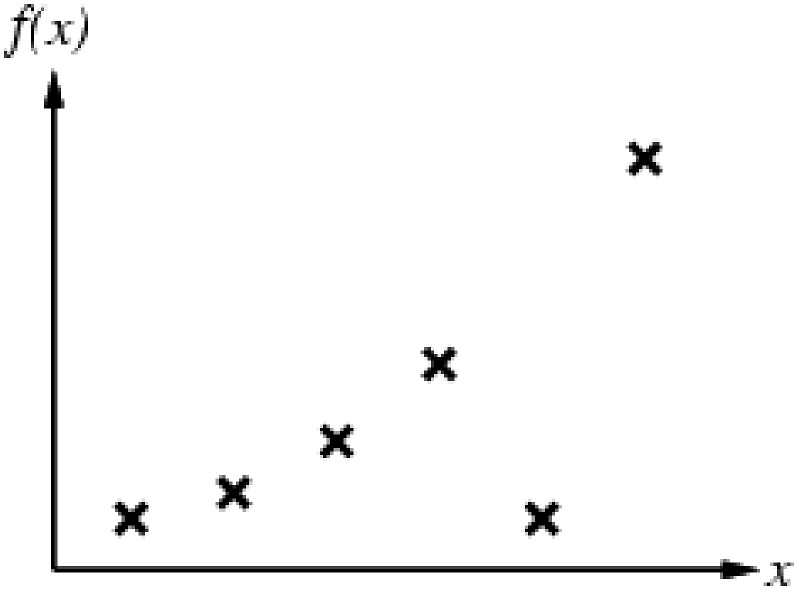
- Ignores prior knowledge
- Assumes examples are given)

Inductive learning method

Construct/adjust *h* to agree with *f* on training set

(*h* is consistent if it agrees with *f* on all examples)

E.g., curve

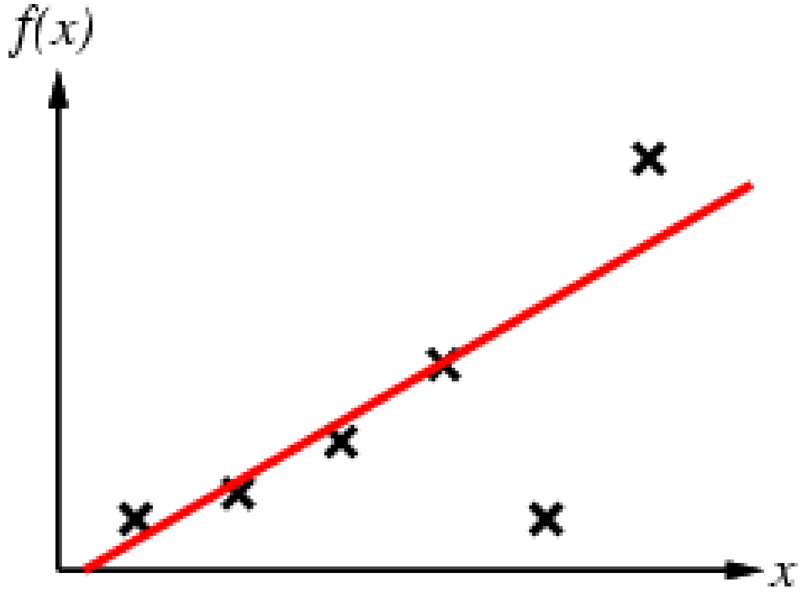


Inductive learning method

Construct/adjust *h* to agree with *f* on training set

(*h* is consistent if it agrees with *f* on all examples)

E.g., curve

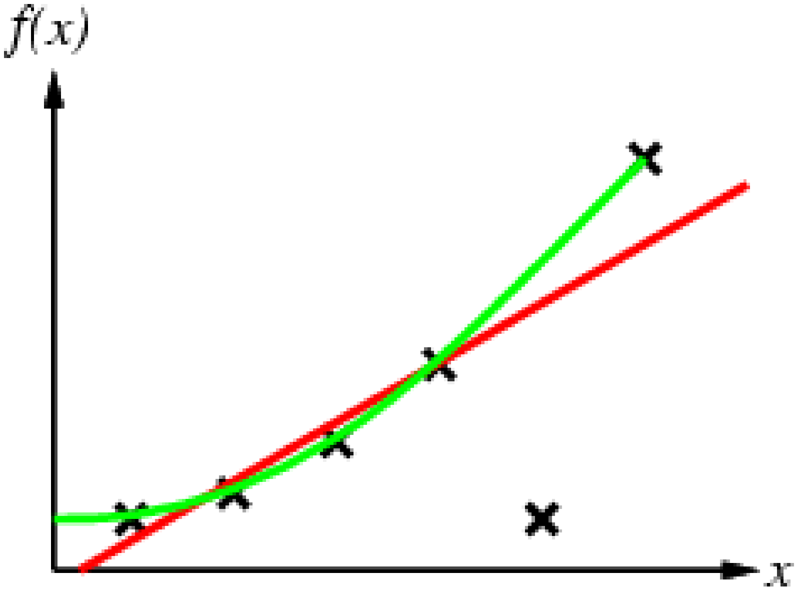


Inductive learning method

Construct/adjust *h* to agree with *f* on training set

(*h* is consistent if it agrees with *f* on all examples)

E.g., curve

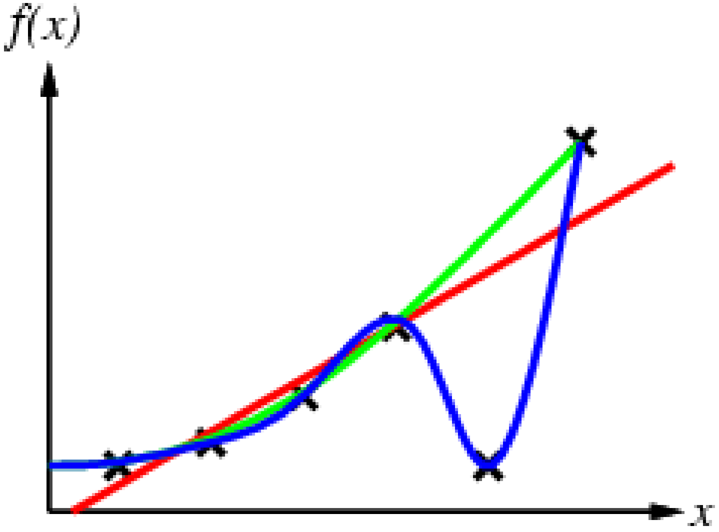


Inductive learning method

Construct/adjust *h* to agree with *f* on training set

(*h* is consistent if it agrees with *f* on all examples)

E.g., curve

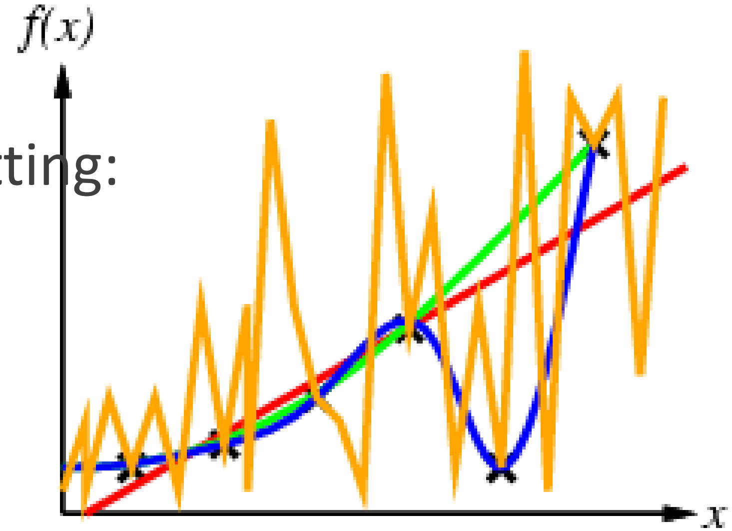


Inductive learning method

Construct/adjust *h* to agree with *f* on training set

(*h* is consistent if it agrees with *f* on all examples)

E.g., curve fitting:

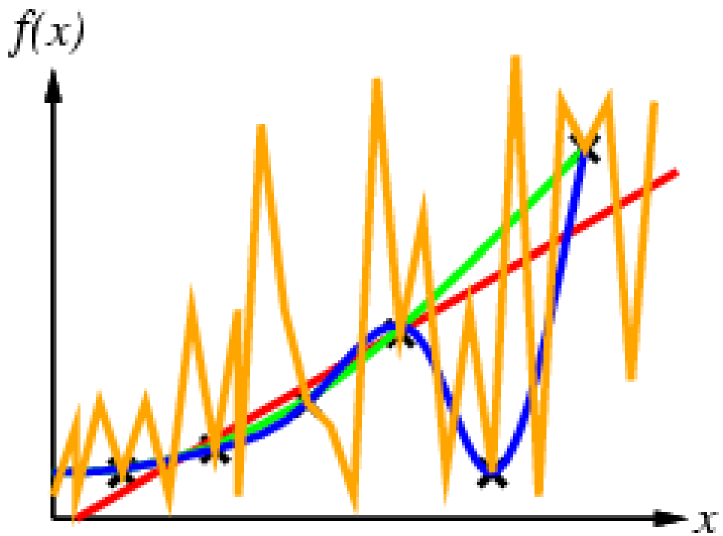


Inductive learning method

Construct/adjust *h* to agree with *f* on training set

(*h* is consistent if it agrees with *f* on all examples)

E.g., curve fitting:



Ockham's razor: prefer the simplest hypothesis consistent with data

Learning decision trees

Problem: decide whether to wait for a table at a restaurant, based on the following attributes:

1. Alternate: is there an alternative restaurant nearby?
2. Bar: is there a comfortable bar area to wait in?
3. Fri/Sat: is today Friday or Saturday?
4. Hungry: are we hungry?
5. Patrons: number of people in the restaurant (None, Some, Full)
6. Price: price range ($, $$, $$$)
7. Raining: is it raining outside?
8. Reservation: have we made a reservation?
9. Type: kind of restaurant (French, Italian, Thai, Burger)
10. WaitEstimate: estimated waiting time (0-10, 10-30, 30-60, >60)

Attribute-based representations

Examples described by attribute values (Boolean, discrete, continuous)

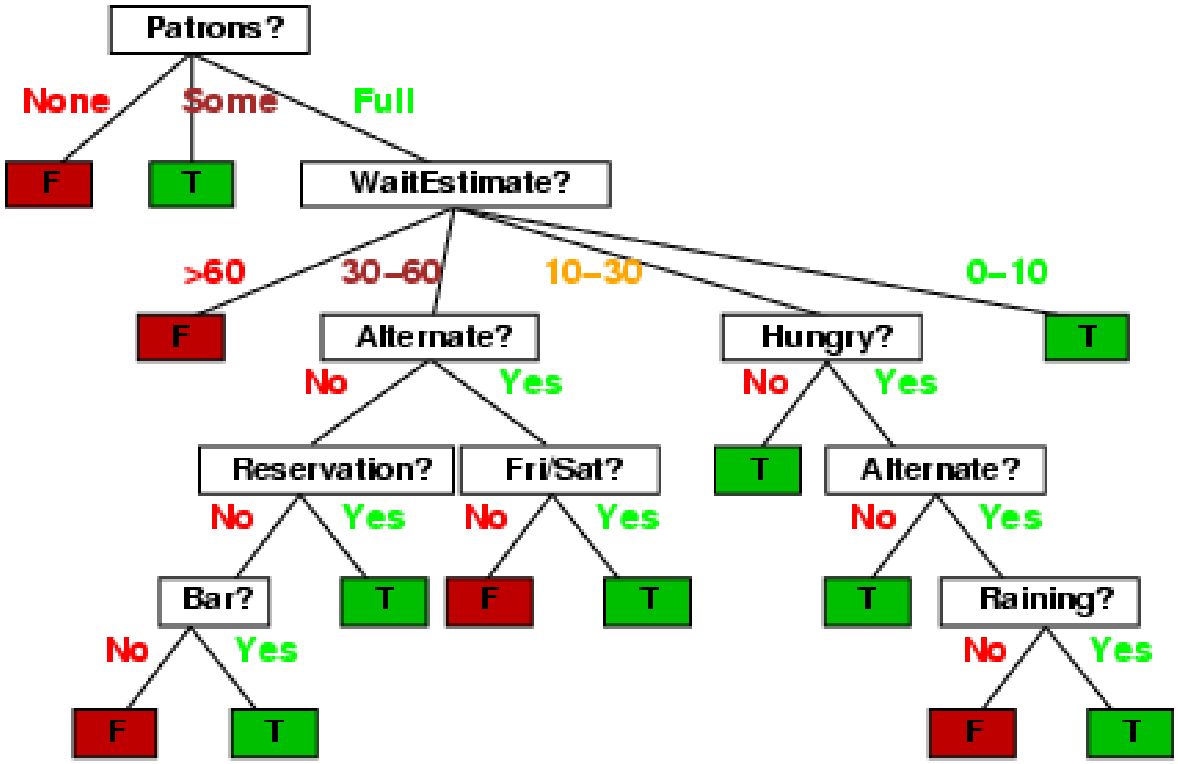
E.g., situations where I will/won't wait for a table:

Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	$Price$	$Rain$	Res	$Type$	Est	$Wait$
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Classification of examples is positive (T) or negative (F)

Decision trees

One possible representation for hypotheses E.g., here is the “true” tree for deciding whether to wait:

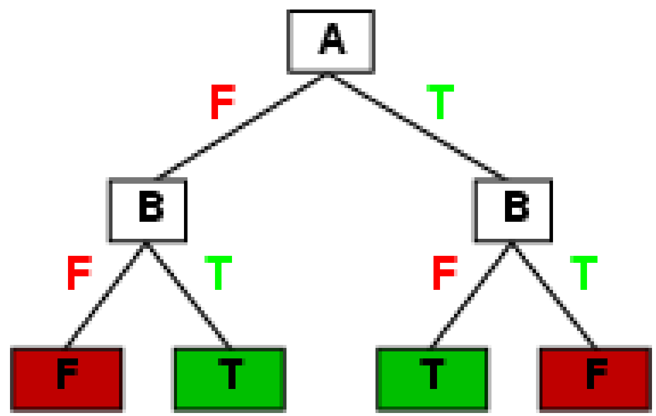


Expressiveness

Decision trees can express any function of the input attributes.

E.g., for Boolean functions, truth table row → path to leaf:

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



Trivially, there is a consistent decision tree for any training set with one path to leaf for each example (unless *f* nondeterministic in *x*) but it probably won't generalize to new examples

Prefer to find more compact decision trees

Hypothesis spaces

How many distinct decision trees with *n* Boolean attributes?

= number of Boolean functions

= number of distinct truth tables with 2^n rows = 2^{2^n}

E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees

Hypothesis spaces

How many distinct decision trees with *n* Boolean attributes?

= number of Boolean functions

= number of distinct truth tables with 2^n rows = 2^{2^n}

E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees

How many purely conjunctive hypotheses (e.g., $Hungry \wedge \neg Rain$)?

Each attribute can be in (positive), in (negative), or out

$\Rightarrow$ 3^n distinct conjunctive hypotheses

More expressive hypothesis space

- increases chance that target function can be expressed
- increases number of hypotheses consistent with training set

$\Rightarrow$ may get worse predictions

Decision tree learning

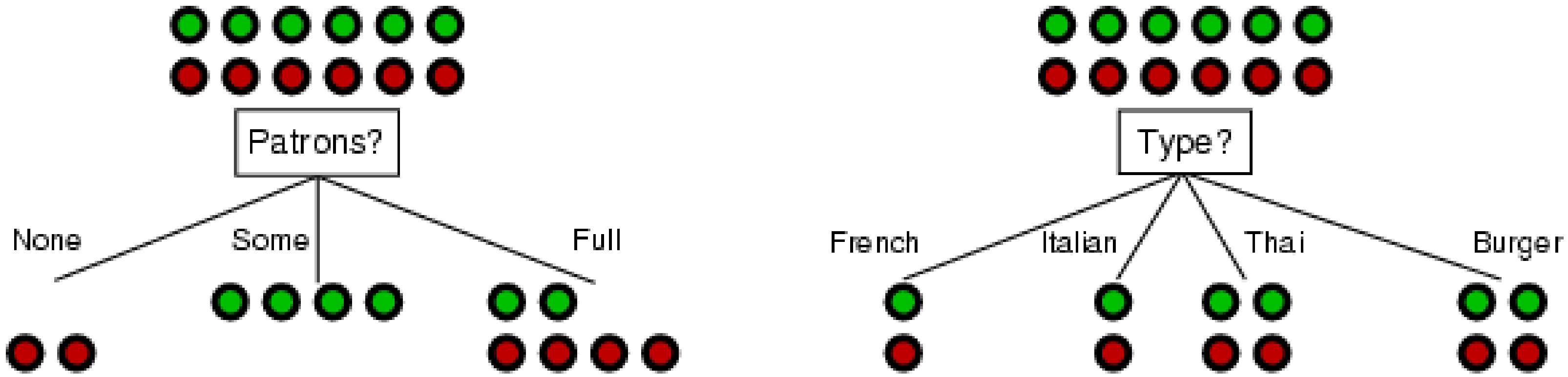
Aim: find a small tree consistent with the training examples

Idea: (recursively) choose "most significant" attribute as root of (sub)tree

```
function DTL(examples, attributes, default) returns a decision tree
    if examples is empty then return default
    else if all examples have the same classification then return the classification
    else if attributes is empty then return MODE(examples)
    else
        best ← CHOOSE-ATTRIBUTE(attributes, examples)
        tree ← a new decision tree with root test best
        for each value v_i of best do
            examples_i ← {elements of examples with best = v_i}
            subtree ← DTL(examples_i, attributes − best, MODE(examples))
            add a branch to tree with label v_i and subtree subtree
        return tree
```

Choosing an attribute

Idea: a good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative"



Patrons? is a better choice

Using information theory

To implement `Choose-Attribute` in the DTL algorithm

Information Content (Entropy):

$$I(P(v_1), \dots , P(v_n)) = \Sigma_{i=1} -P(v_i) \log_2 P(v_i)$$

For a training set containing *p* positive examples and *n* negative examples:

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Information gain

A chosen attribute *A* divides the training set *E* into subsets E_1, … , E_v according to their values for *A*, where *A* has v distinct values.

$$remainder(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

Information Gain (IG) or reduction in entropy from the attribute test:

$$IG(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - remainder(A)$$

Choose the attribute with the largest IG

Information gain

For the training set, $p = n = 6$, $I(6/12, 6/12) = 1$ bit

Consider the attributes *Patrons* and *Type* (and others too):

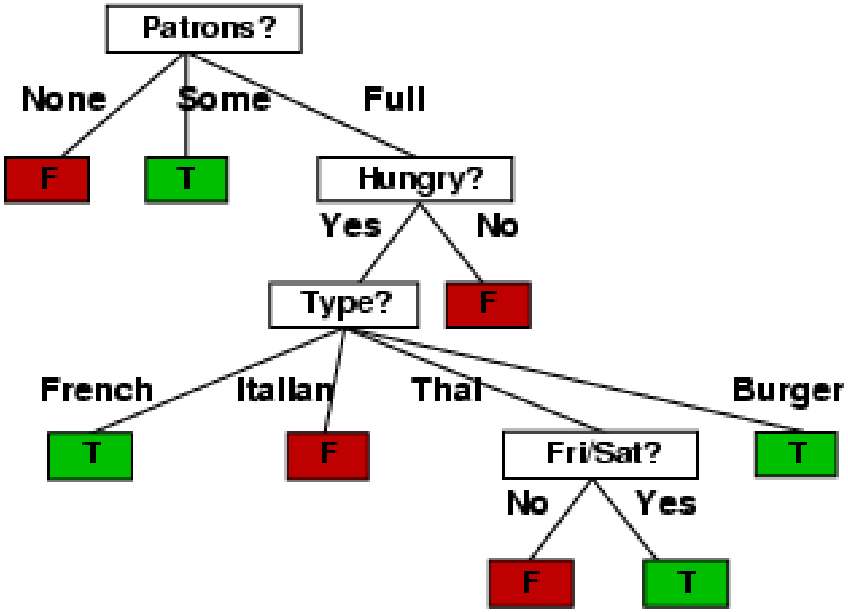
$$IG(Patrons) = 1 - [\frac{2}{12} I(0,1) + \frac{4}{12} I(1,0) + \frac{6}{12} I(\frac{2}{6}, \frac{4}{6})] = .0541 \text{ bits}$$

$$IG(Type) = 1 - [\frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4})] = 0 \text{ bits}$$

Patrons has the highest IG of all attributes and so is chosen by the DTL algorithm as the root

Example contd.

Decision tree learned from the 12 examples:



Substantially simpler than "true" tree---a more complex hypothesis isn't justified by small amount of data

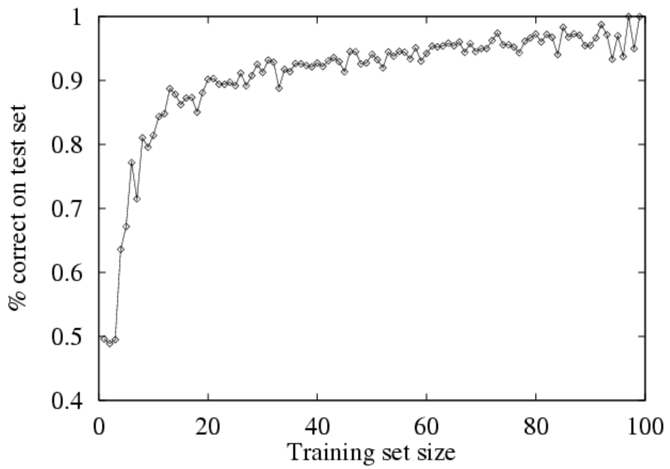
Performance measurement

How do we know that $h \approx f$?

1. Use theorems of computational/statistical learning theory
2. Try h on a new test set of examples

   (use same distribution over example space as training set)

Learning curve = % correct on test set as a function of training set size



Summary 1

Learning needed for unknown environments, lazy designers

Learning agent = performance element + learning element

For supervised learning, the aim is to find a simple hypothesis approximately consistent with training examples

Decision tree learning using information gain

Learning performance = prediction accuracy measured on test set

Statistical Learning

Parameter Estimation:

- Maximum Likelihood (ML)
- Maximum A Posteriori (MAP)
- Bayesian
- Continuous case

Learning Parameters for a Bayesian Network

Naive Bayes

- Maximum Likelihood estimates
- Priors

Learning Structure of Bayesian Networks

Coin Flip

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

Which coin will I use?

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

Prior: Probability of a hypothesis before we make any observations

Coin Flip

C_1 C_2 C_3

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

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

$P(H|C_1) = 0.1$

$P(C_1) = 1/3$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Terminology

Prior:
- Probability of a hypothesis before we see any data

Uniform Prior:
- A prior that makes all hypothesis equaly likely

Posterior:
- Probability of a hypothesis after we saw some data

Likelihood:
- Probability of data given hypothesis

Experiment 2: Tails

Which coin did I use?

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

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 1/3$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Experiment 2: Tails

Which coin did I use?

$P(C_1|HT)$ = 0.21 $P(C_2|HT)$ = 0.58 $P(C_3|HT)$ = 0.21

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 1/3$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Experiment 2: Tails

Which coin did I use?

$P(C_1|HT) = 0.21$ $P(C_2|HT) = 0.58$ $P(C_3|HT) = 0.21$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 1/3$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Your Estimate?

What is the probability of heads after two experiments?

Most likely coin:

C_2 

Best estimate for P(H)

$P(H|C_2) = 0.5$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 1/3$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Your Estimate?

Maximum Likelihood Estimate: The best hypothesis that fits observed data assuming uniform prior

Most likely coin:

C_2 

Best estimate for P(H)

$P(H|C_2) = 0.5$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

Using Prior Knowledge

Should we always use a ***Uniform Prior*** ?

Background knowledge:

- Heads => we have take-home midterm
- Dan doesn't like take-homes…
- => 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 Prior 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 <u>did</u> I use?

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

$$P(C_1|H) = \alpha P(H|C_1)P(C_1)$$

C_1 C_2 C_3

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

$\boxed{P(C_1) = 0.05 \quad P(C_2) = 0.25 \quad P(C_3) = 0.70}$

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 C_2 C_3

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

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

Experiment 2: Tails

Which coin did I use?

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

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

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

$P(H|C_3) = 0.9$

$P(C_3) = 0.70$

Experiment 2: Tails

Which coin did I use?

$P(C_1|HT) = 0.035$ $P(C_2|HT) = 0.481$ $P(C_3|HT) = 0.485$

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1	C_2	C_3
		
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$
$P(C_1) = 0.05$	$P(C_2) = 0.25$	$P(C_3) = 0.70$

Experiment 2: Tails

Which coin <u>did</u> I use?

$P(C_1|HT) = 0.035$ $P(C_2|HT)=0.481$ $P(C_3|HT) = 0.485$

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

$P(H|C_3) = 0.9$

$P(C_3) = 0.70$

Your Estimate?

What is the probability of heads after two experiments?

Most likely coin:

C_3

Best estimate for P(H)

$P(H|C_3) = 0.9$

C_1	C_2	C_3
		
$P(H\|C_1) = 0.1$	$P(H\|C_2) = 0.5$	$P(H\|C_3) = 0.9$
$P(C_1) = 0.05$	$P(C_2) = 0.25$	$P(C_3) = 0.70$

Your Estimate?

Maximum A Posteriori (MAP) Estimate:
The best hypothesis that fits observed data assuming a non-uniform prior

Most likely coin:

C_3

Best estimate for P(H)

$P(H|C_3) = 0.9$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 0.70$

Did We Do The Right Thing?

$P(C_1|HT)=0.035$ $P(C_2|HT)=0.481$ $P(C_3|HT)=0.485$

C_1

$P(H|C_1) = 0.1$

C_2

$P(H|C_2) = 0.5$

C_3

$P(H|C_3) = 0.9$

Did We Do The Right Thing?

$P(C_1|HT) = 0.035$ $P(C_2|HT) = 0.481$ $P(C_3|HT) = 0.485$

C_2 and C_3 are almost equally likely

C_1 C_2 C_3

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

A Better Estimate

Recall: $P(H) = \sum_{i=1}^{3} P(H|C_i)P(C_i)$ = 0.680

$P(C_1|HT)=0.035$ $P(C_2|HT)=0.481$ $P(C_3|HT)=0.485$

C_1

$P(H|C_1) = 0.1$

C_2

$P(H|C_2) = 0.5$

C_3

$P(H|C_3) = 0.9$

Bayesian Estimate

Bayesian Estimate: Minimizes prediction error, given data and (generally) assuming a non-uniform prior

$$P(H) = \sum_{i=1}^{3} P(H|C_i)P(C_i) = 0.680$$

$P(C_1|HT)=0.035$ $P(C_2|HT)=0.481$ $P(C_3|HT)=0.485$

C_1 C_2 C_3

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

Comparison
After more experiments: HTH^8

ML (Maximum Likelihood):

P(H) = 0.5
after 10 experiments: P(H) = 0.9

MAP (Maximum A Posteriori):

P(H) = 0.9
after 10 experiments: P(H) = 0.9

Bayesian:

P(H) = 0.68
after 10 experiments: P(H) = 0.9

Comparison

ML (Maximum Likelihood):

Easy to compute

MAP (Maximum A Posteriori):

Still easy to compute

Incorporates prior knowledge

Bayesian:

Minimizes error => great when data is scarce

Potentially much harder to compute

Summary For Now

	Prior	Hypothesis
Maximum Likelihood Estimate	Uniform	The most likely
Maximum A Posteriori Estimate	Any	The most likely
Bayesian Estimate	Any	Weighted combination

Continuous Case

In the previous example,

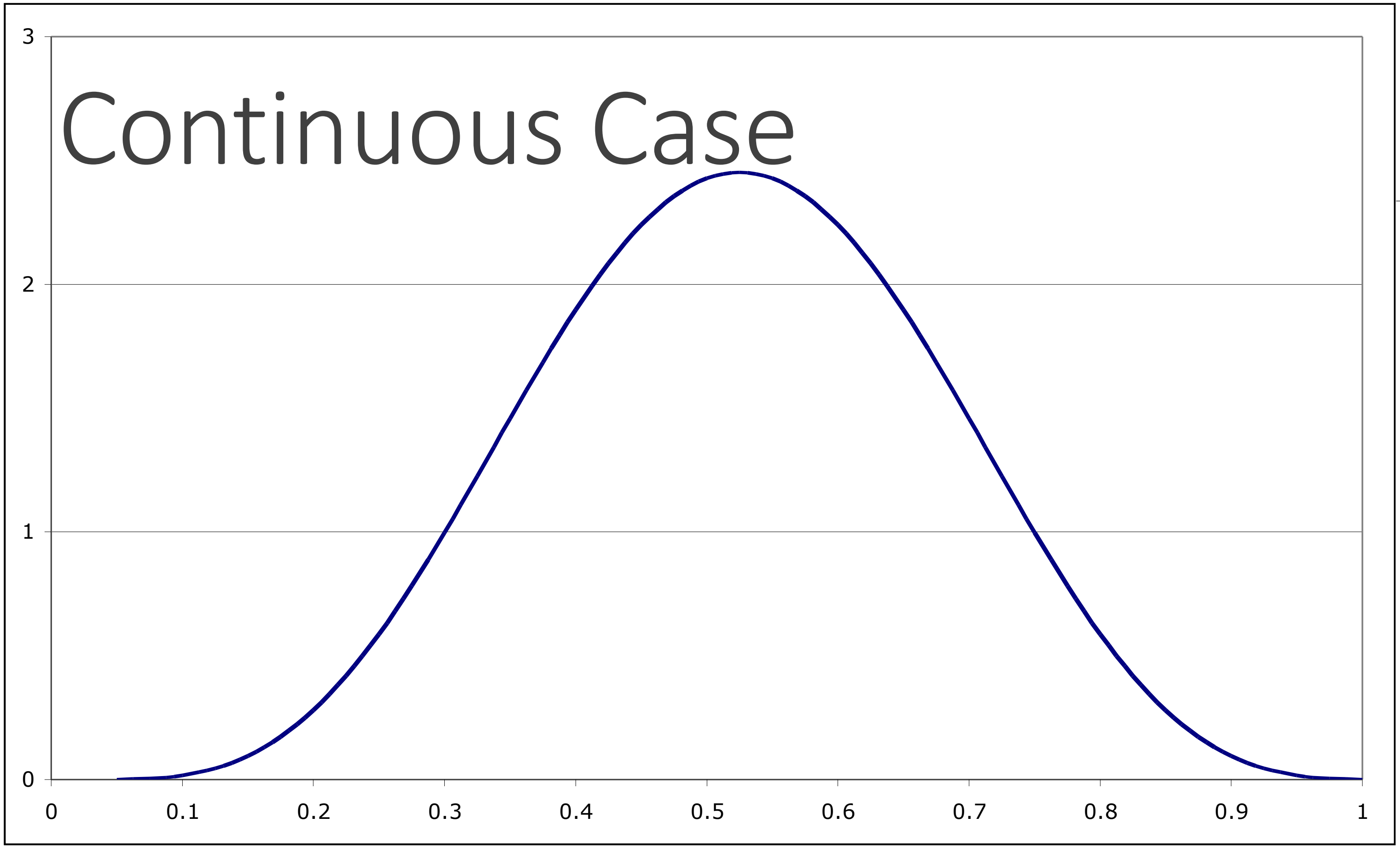
- we chose from a discrete set of three coins

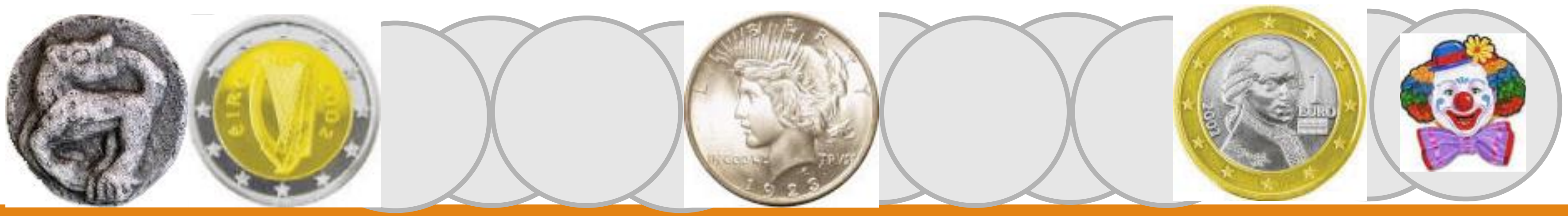
In general,

- we have to pick from a continuous distribution
- of biased coins

Continuous Case

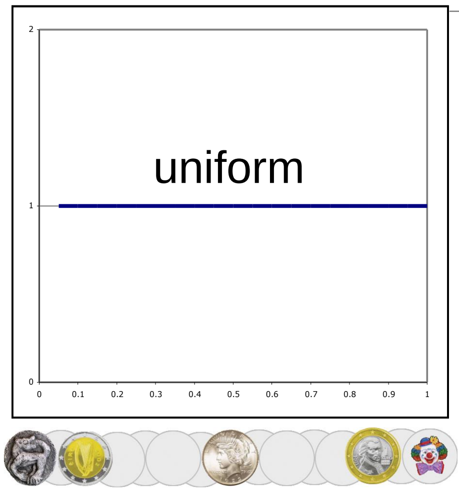
Continuous Case

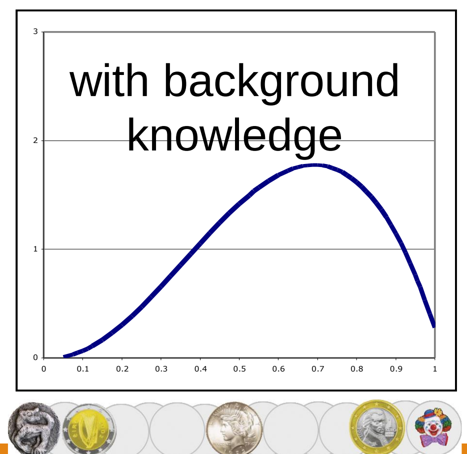




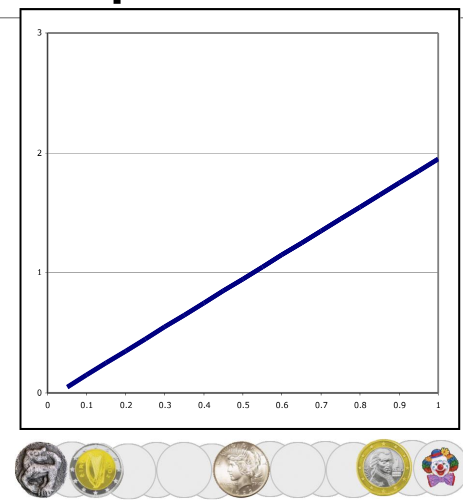
Continuous Case

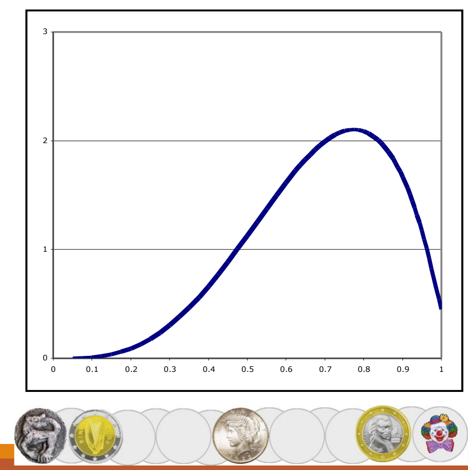
Prior



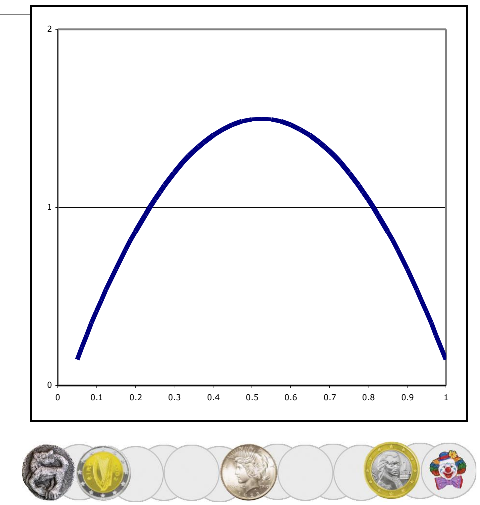


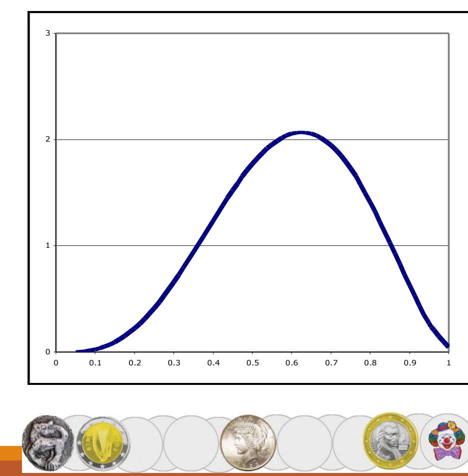
Exp 1: Heads





Exp 2: Tails





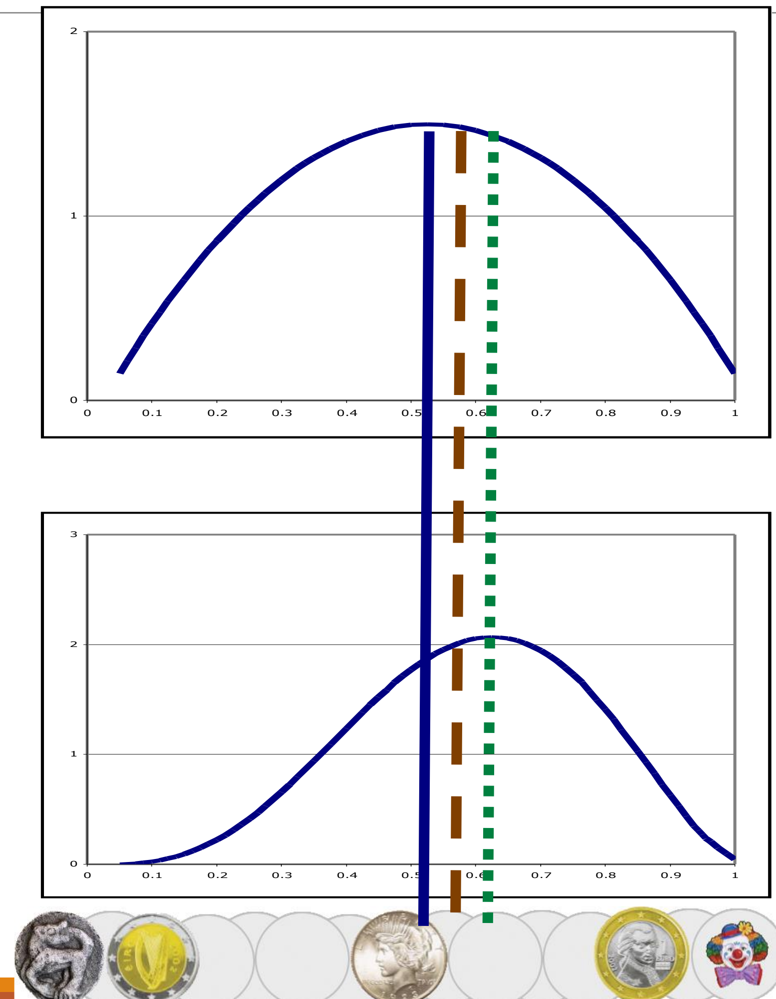
Continuous Case

Posterior after 2 experiments:

ML Estimate

MAP Estimate

Bayesian Estimate



w/ uniform prior

with background knowledge

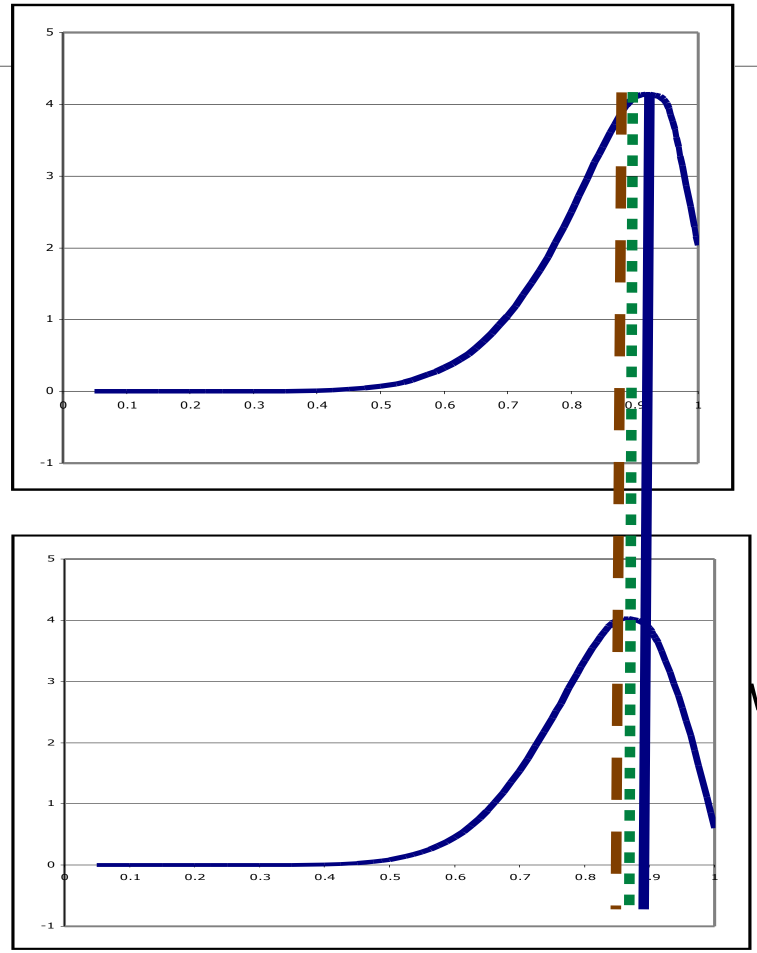
After 10 Experiments...

Posterior:

ML Estimate

MAP Estimate

Bayesian Estimate



w/ uniform prior

with background knowledge

After 100 Experiments...

Topics

Parameter Estimation:
- Maximum Likelihood (ML)
- Maximum A Posteriori (MAP)
- Bayesian
- Continuous case

Learning Parameters for a Bayesian Network

Naive Bayes
- Maximum Likelihood estimates
- Priors

Learning Structure of Bayesian Networks

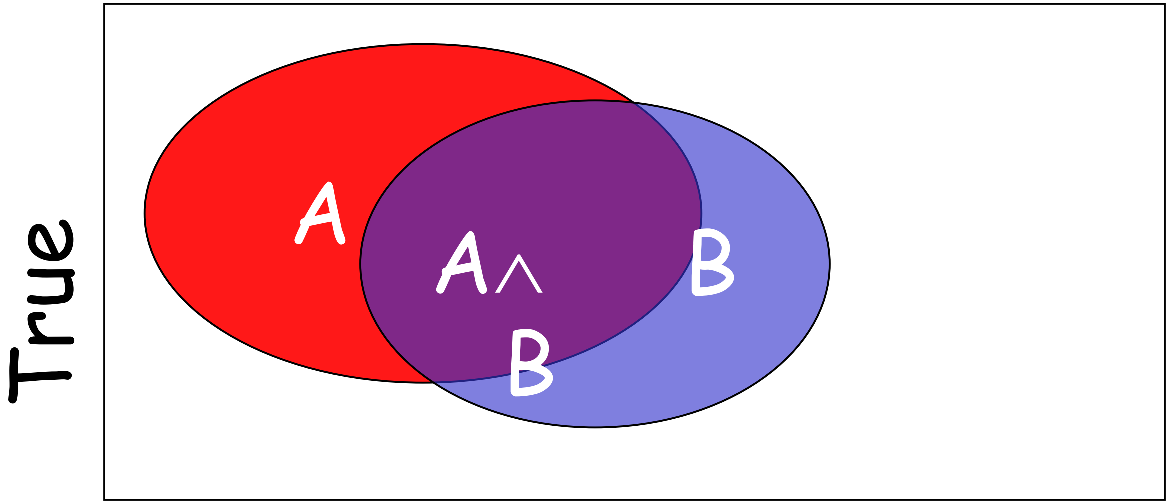
Review: Conditional Probability

P(A | B) is the probability of A given B

Assumes that B is the only info known.

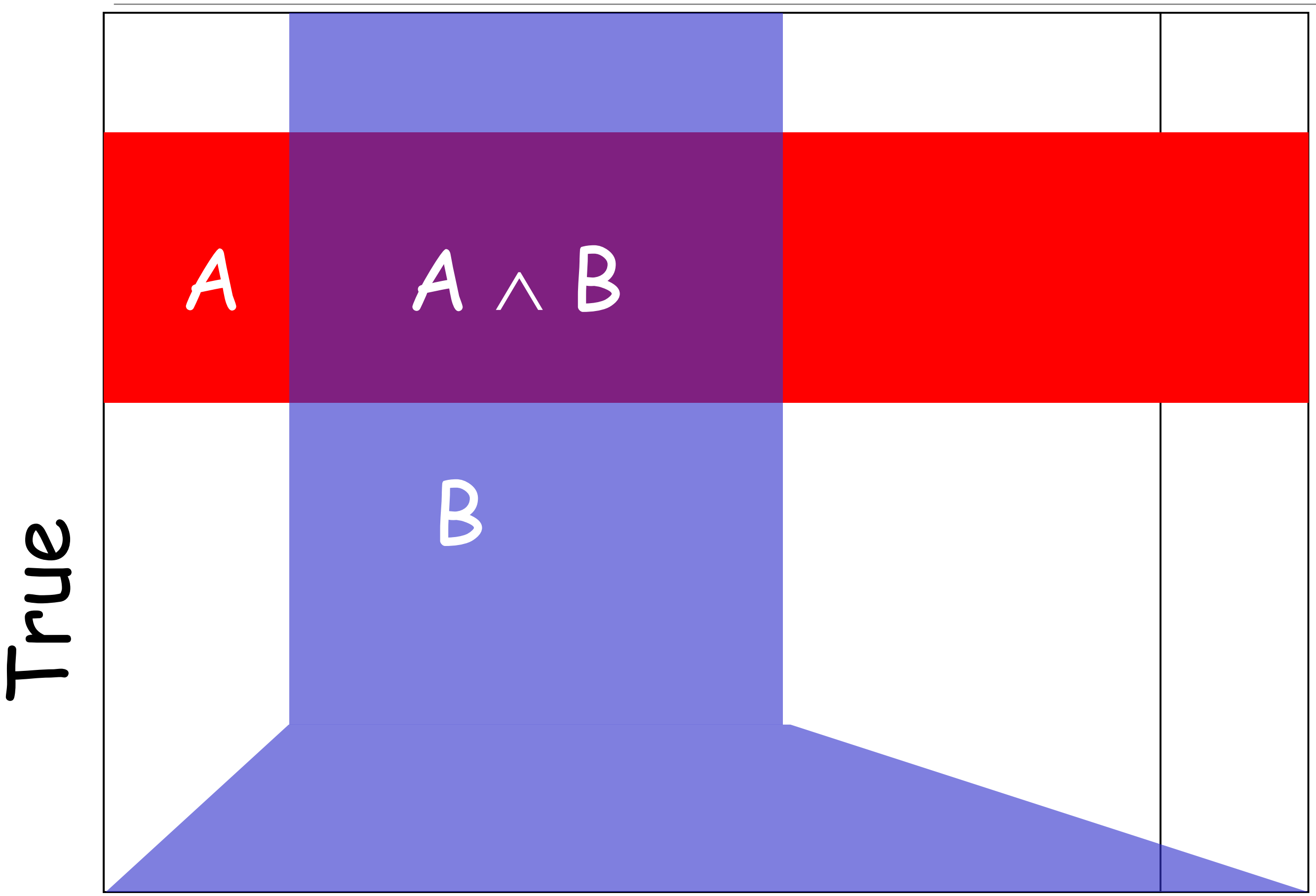
Defined by:

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



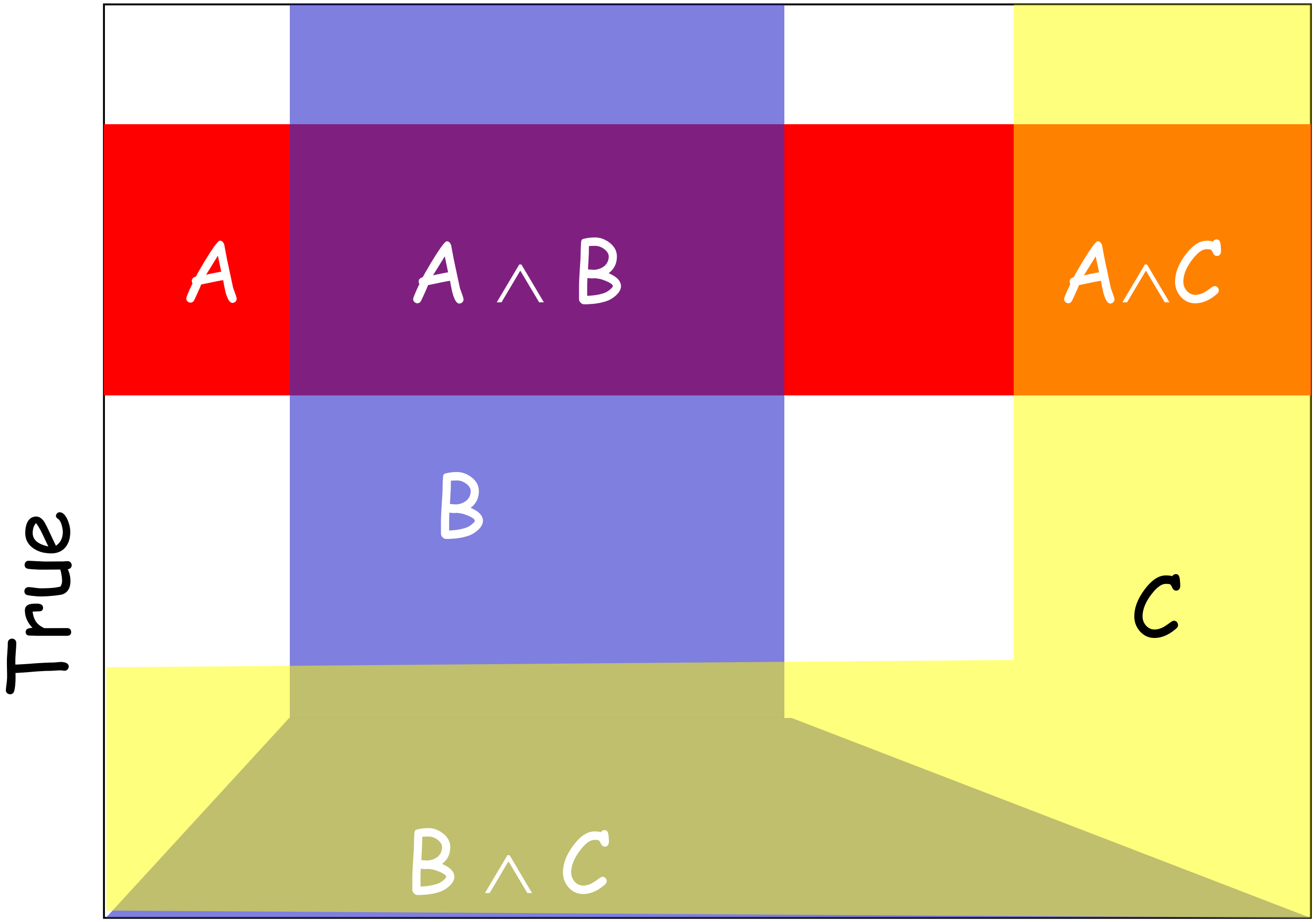
Conditional Independence

A&B not independent, since P(A|B) < P(A)



Conditional Independence

But: A&B are made independent by $\neg C$



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

Bayes Rule

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$

Simple proof from def of conditional probability:

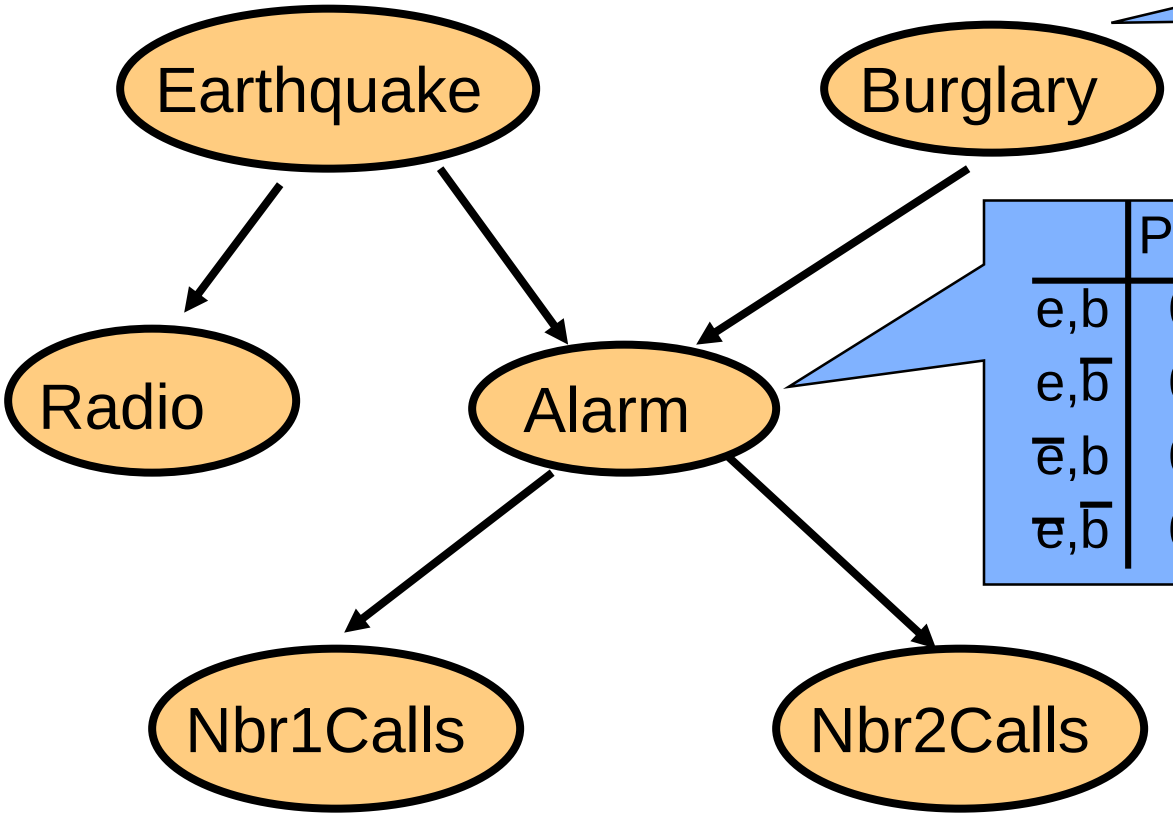
$$P(H \mid E) = \frac{P(H \wedge E)}{P(E)}$$ (Def. cond. prob.)

$$P(E \mid H) = \frac{P(H \wedge E)}{P(H)}$$ (Def. cond. prob.)

$$P(H \wedge E) = P(E \mid H)P(H)$$ (Mult by P(H) in line 1)

QED: $$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$ (Substitute #3 in #2)

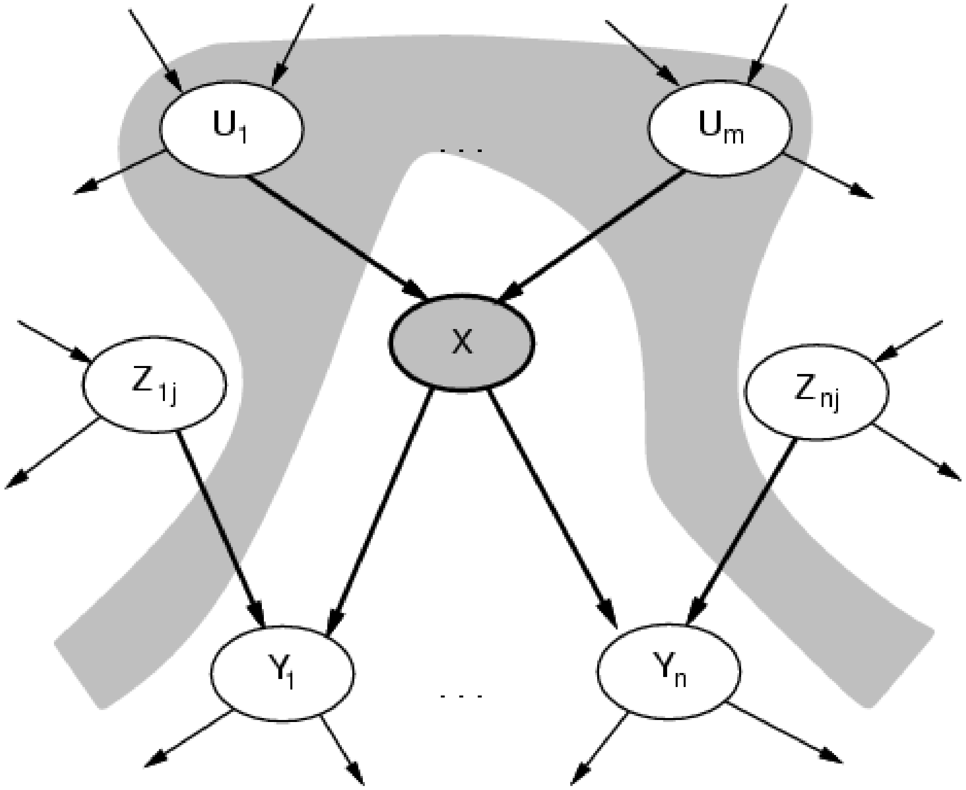
An Example Bayes Net



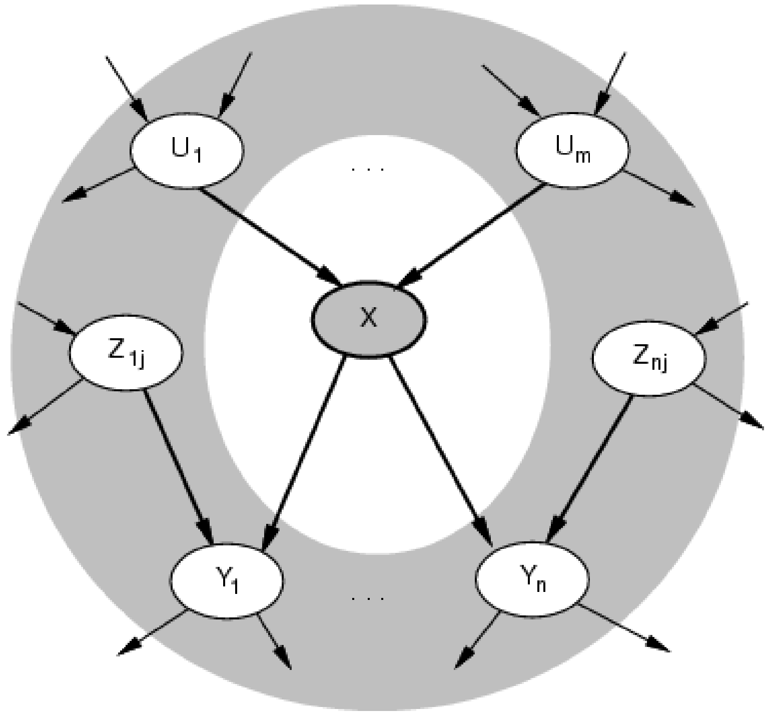
Pr(B=t)	Pr(B=f)
0.05	0.95

	Pr(A|E,B)
e,b	0.9 (0.1)
e,$\bar{b}$	0.2 (0.8)
$\bar{e}$,b	0.85 (0.15)
$\bar{e}$,$\bar{b}$	0.01 (0.99)

Given Parents, X is Independent of Non-Descendants

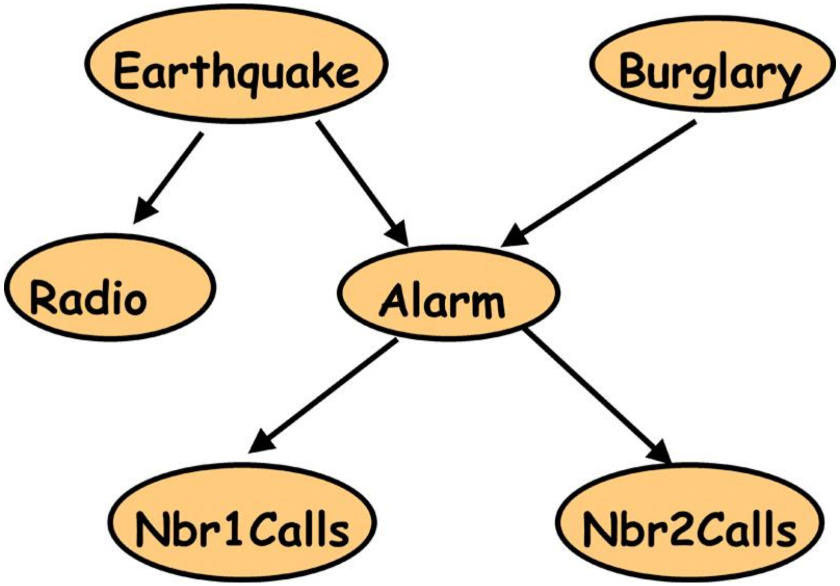


Given Markov Blanket, X is Independent of All Other Nodes



$$MB(X) = Par(X) \cup Childs(X) \cup Par(Childs(X))$$

Parameter Estimation and Bayesian Networks

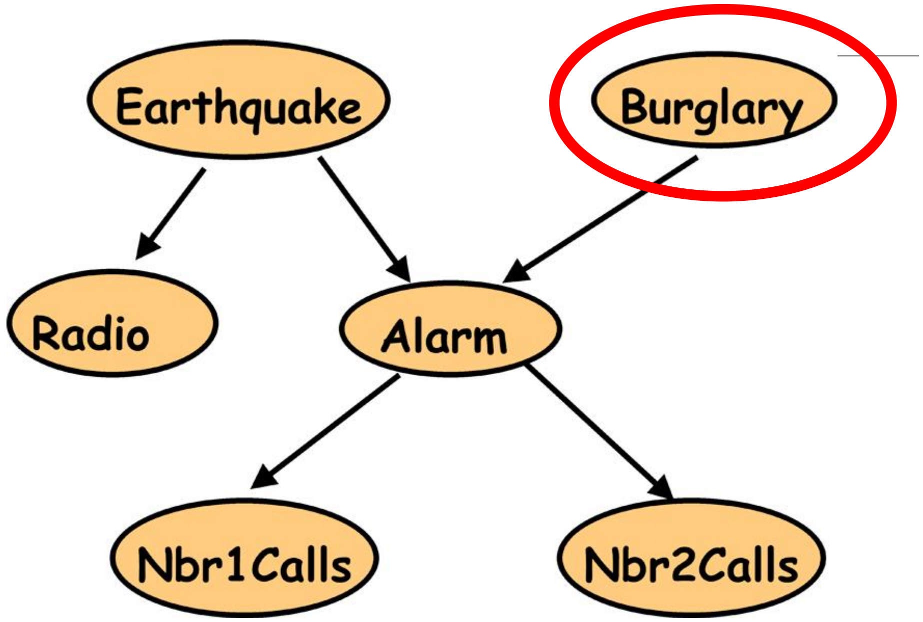


E	B	R	A	J	M
T	F	T	T	F	T
F	F	F	F	F	T
F	T	F	T	T	T
F	F	F	T	T	T
F	T	F	F	F	F
...					

We have:

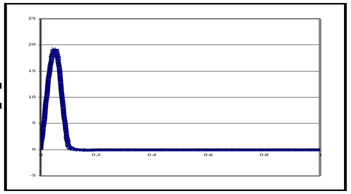
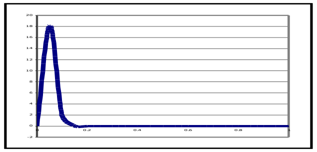
- Bayes Net structure and observations
- We need: Bayes Net parameters

Parameter Estimation and Bayesian Networks



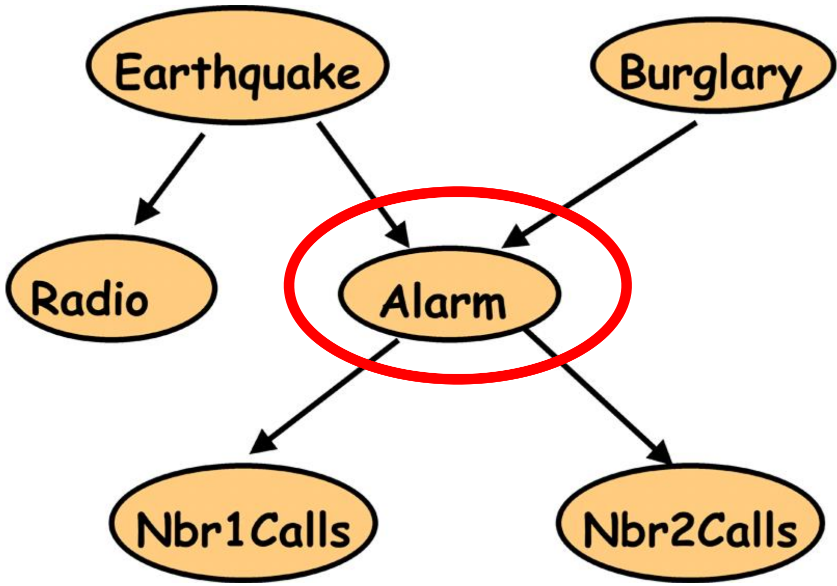
B
F
F
T
F
T

Prior

P(B) =  + data = 

Now compute either MAP or Bayesian estimate

Parameter Estimation and Bayesian Networks

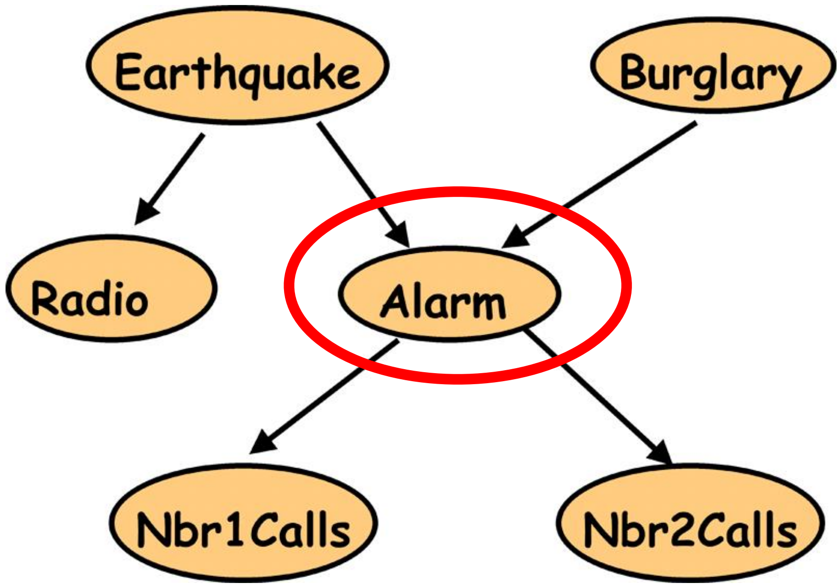


E	B
T	F
F	F
F	T
F	F
F	T
...	

A
T
F
T
T
F

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

Parameter Estimation and Bayesian Networks



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

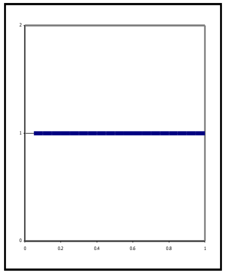
P(A|E,B) = ?

P(A|E,¬B) = ?

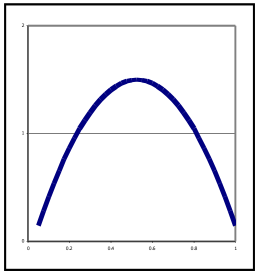
P(A|¬E,B) = ?

P(A|¬E,¬B) = ?

Prior



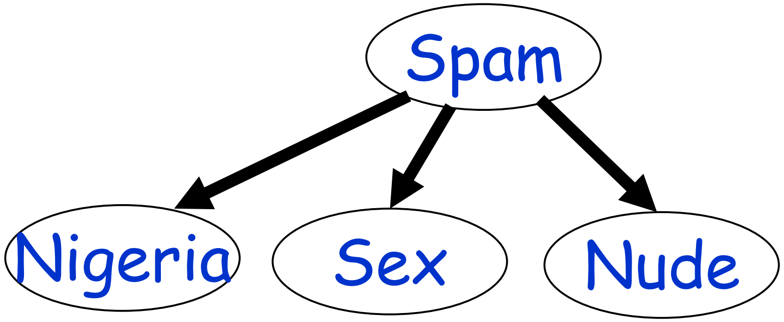
+ data=

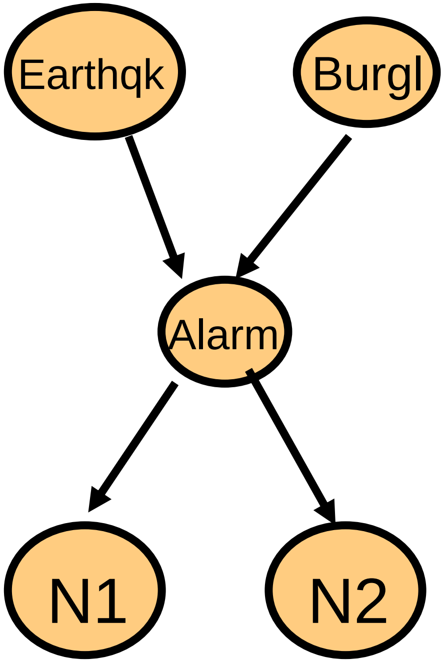


Now compute either MAP or Bayesian estimate

Recap

Given a BN structure (with discrete or continuous variables), we can learn the parameters of the conditional prop tables.





What if we *don't* know structure?

Learning The Structure of Bayesian Networks

Search thru the space…
- of possible network structures!
- (for now, assume we observe all variables)

For each structure, learn parameters

Pick the one that fits observed data best
- Caveat – won’t we end up fully connected????

Problem !?!?

When scoring, add a penalty

$\propto$ model complexity

Learning The Structure of Bayesian Networks

Search thru the space

For each structure, learn parameters

Pick the one that fits observed data best

Problem?

Exponential number of networks!
And we need to learn parameters for each!
Exhaustive search out of the question!

So what now?

Learning The Structure of Bayesian Networks

Local search!

- Start with some network structure
- Try to make a change
- (add or delete or reverse edge)
- See if the new network is any better

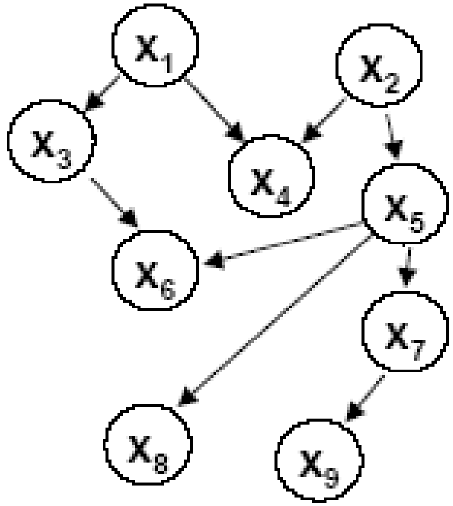
- What should be the initial state?

Initial Network Structure?

Uniform prior over random networks?

Network which reflects expert knowledge?

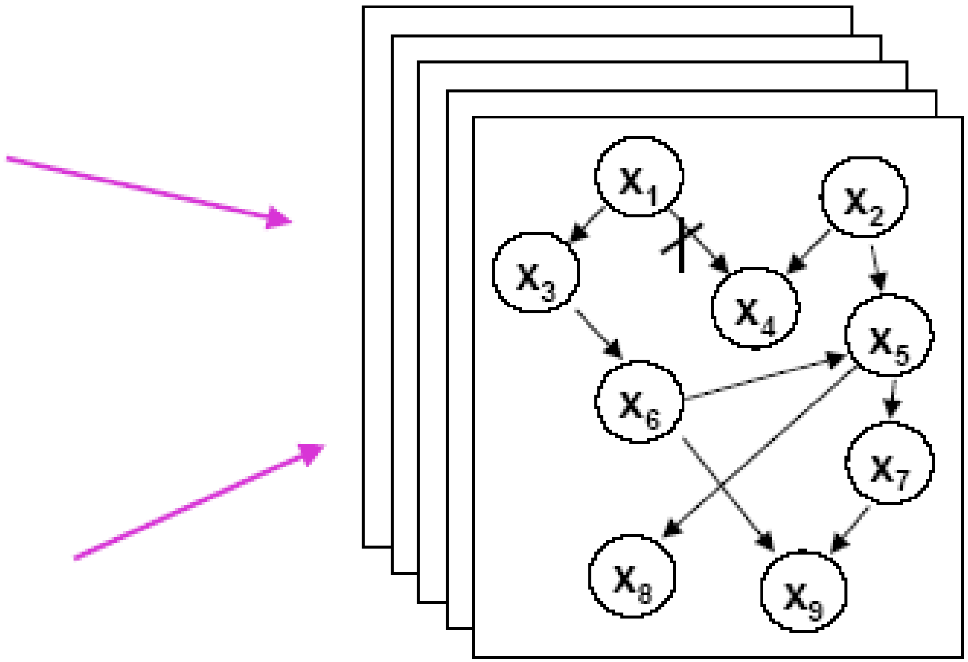
prior network+equivalent sample size



data

X_1	X_2	X_3	
true	false	true	
false	false	true	
false	false	false	...
true	true	false	
	⋮		⋱

improved network(s)



The Big Picture

We described how to do MAP (and ML) learning of a Bayes net (including structure)

How would Bayesian learning (of BNs) differ?

Find all possible networks

Calculate their posteriors

When doing inference, return weighed combination of predictions from all networks!