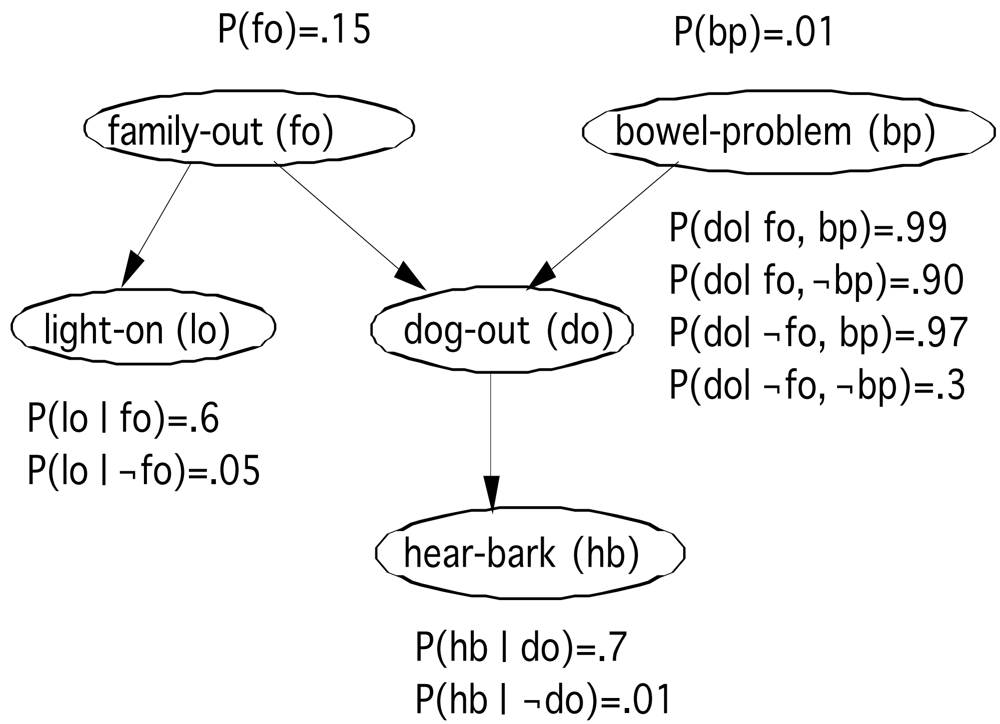
Inference in Bayesian Networks

Bayesian Network

- Independence assumptions
  - Seems to be necessary for probabilistic inference to be practical.
- Naïve Bayes Method
  - Makes independence assumptions that are often not true
  - Also called Idiot Bayes Method for this reason.
- Bayesian Network
  - Explicitly models the independence relationships in the data.
  - Use these independence relationships to make probabilistic inferences.
  - Also known as: Belief Net, Bayes Net, Causal Net, …

Probabilistic Inference

Network represents the joint probability over all the variables



$$
\begin{aligned}
P(hb,do,lo,fo,bp) &= P(hb|do,lo,fo,bp)*P(do,lo,fo,bp)\\
&= P(hb|do,lo,fo,bp)*P(do|lo,fo,bp)*P(lo,fo,bp)\\
&= P(hb|do,lo,fo,bp)*P(do|lo,fo,bp)*P(lo|fo,bp)P(fo,bp)\\
&= P(hb|do,lo,fo,bp)*P(do|lo,fo,bp)*P(lo|fo,bp)P(fo|bp)P(bp)
\end{aligned}
$$

Compactness

- How many numbers are required to build a Bayes Net?
  - For a Boolean variable X_i with k Boolean parents, how many rows in the CPT?
  - If each variable has no more than k parents and there are n nodes in the network, how many numbers required?
- How many numbers required to specify the full joint distribution?

Inference Overview

- Exact Inference
  - Enumeration
  - Variable Elimination
  - Belief Propagation
- Approximate Inference

Inference in Bayesian Networks

- The inputs to a Bayesian Network evaluation algorithm is a set of evidences: e.g.,

  E = { hear-bark=true, lights-on=true }

- The outputs of Bayesian Network evaluation algorithm are
  - Simple queries

    $$P(X_i \mid E)$$

    where X_i is a variable in the network.
  - conjunctive queries:

    $P(X_i, X_j \mid E)$

Inference by Enumeration

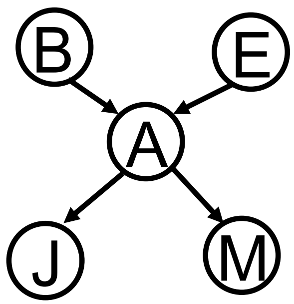
- Bayes Nets represent a joint probability

Simple query on the burglary network:

$\mathrm{P}(B|j,m)$

$= \mathrm{P}(B,j,m) / \mathrm{P}(j,m)$

$= \alpha\mathrm{P}(B,j,m)$



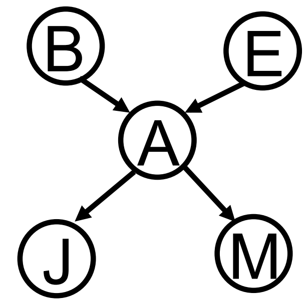
Inference By Variable Elimination

- Carry out sums from right to left, storing intermediate results to avoid recomputation

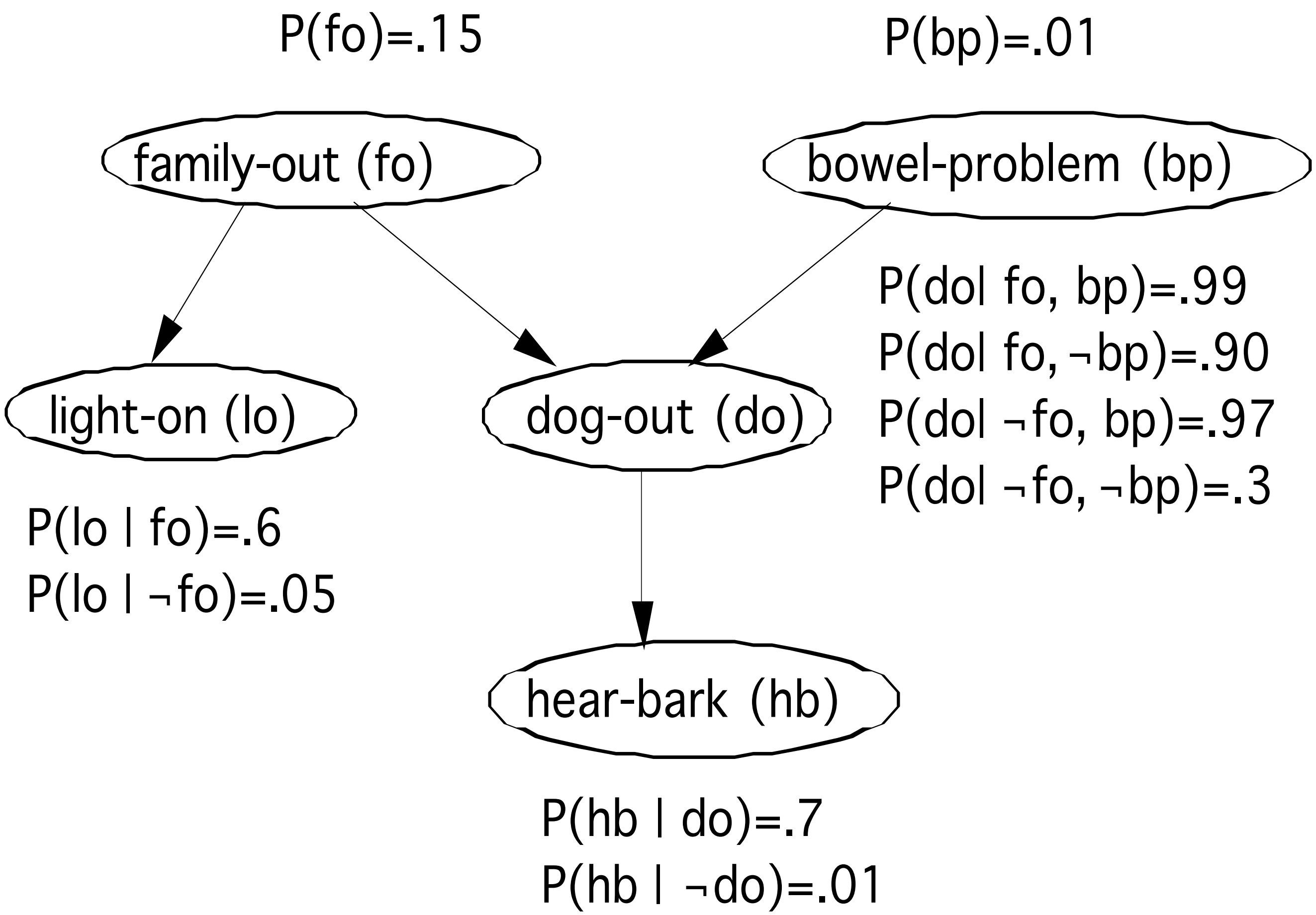
$$
\begin{aligned}
\mathbf{P}(B|j,m) & \\
&= \alpha \underbrace{\mathbf{P}(B)}_{B} \textstyle\sum_e \underbrace{P(e)}_{E} \textstyle\sum_a \underbrace{\mathbf{P}(a|B,e)}_{A} \underbrace{P(j|a)}_{J} \underbrace{P(m|a)}_{M} \\
&= \alpha \mathbf{P}(B) \textstyle\sum_e P(e) \textstyle\sum_a \mathbf{P}(a|B,e) P(j|a) f_M(a) \\
&= \alpha \mathbf{P}(B) \textstyle\sum_e P(e) \textstyle\sum_a \mathbf{P}(a|B,e) f_J(a) f_M(a) \\
&= \alpha \mathbf{P}(B) \textstyle\sum_e P(e) \textstyle\sum_a f_A(a,b,e) f_J(a) f_M(a) \\
&= \alpha \mathbf{P}(B) \textstyle\sum_e P(e) f_{\bar{A}JM}(b,e) \text{ (sum out } A) \\
&= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\
&= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
\end{aligned}
$$

Irrelevant variables

- What if you want to know P(J|b)?



$$P(J \mid b) = \alpha P(b) \sum_e P(e) \sum_a P(a \mid b, e) P(J \mid a) \sum_m P(m \mid a)$$

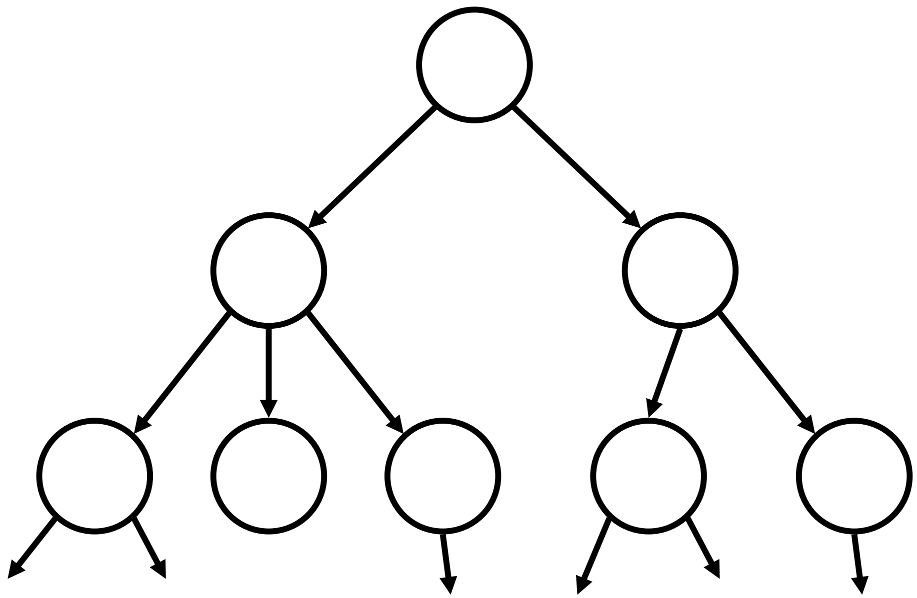

P(fo)=.15
P(bp)=.01
family-out (fo)
bowel-problem (bp)
light-on (lo)
dog-out (do)
hear-bark (hb)
P(lo | fo)=.6
P(lo | ¬fo)=.05
P(do| fo, bp)=.99
P(do| fo, ¬bp)=.90
P(do| ¬fo, bp)=.97
P(do| ¬fo, ¬bp)=.3
P(hb | do)=.7
P(hb | ¬do)=.01

So is VE any better than Enumeration?

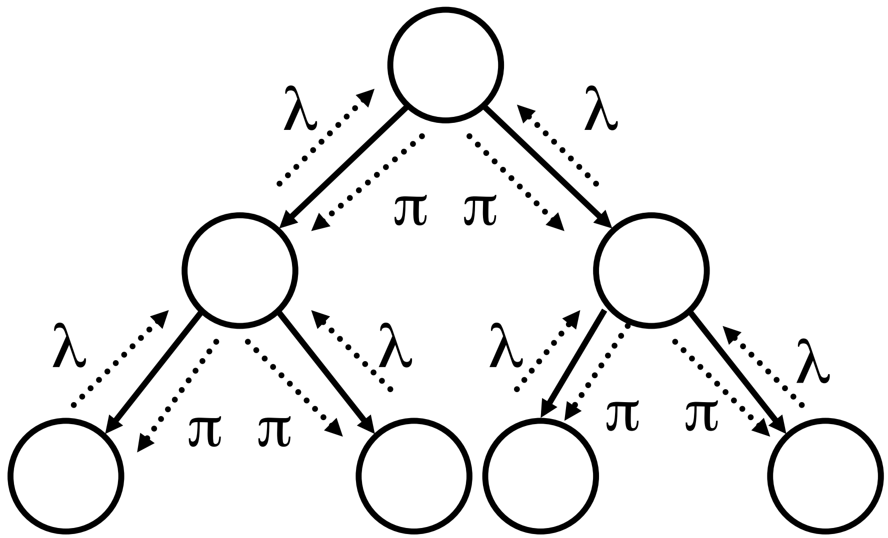
- Yes and No…
  - For singly-connected networks (poly-trees), YES
  - In general, NO…

Bayesian Network Inference

- **But**…inference is still tractable in some cases.
- Special case: trees (each node has one parent)
- VE is LINEAR in this case



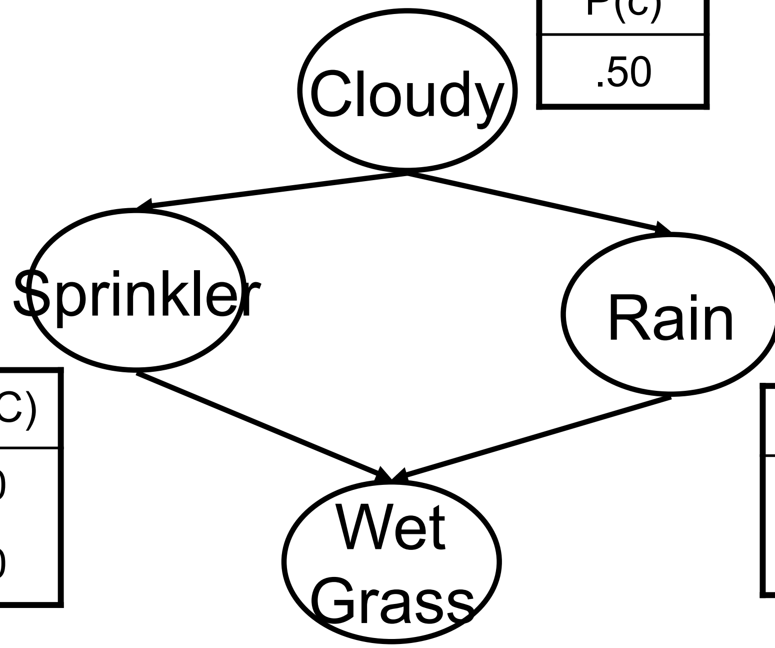
Another Algorithm: Belief Propagation



So far…

- Bayesian networks encode independence assumptions
- Inference in Bayes Nets is inherently difficult, unless the network has special structure
- Variable Elimination and Belief Propagation save time by:
  - Exploiting independence
  - Storing intermediate results
  - Eliminating irrelevant variables

So, what about multiply connected graphs?



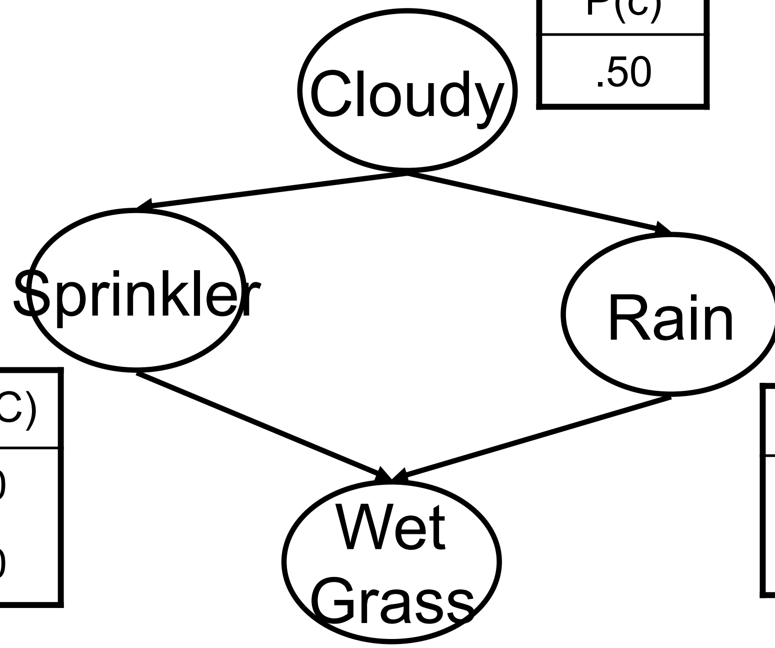
P(c)
.50

C	P(s|C)
T	.10
F	.50

C	P(r|C)
T	.80
F	.20

S	R	P(w|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

So, what about multiply connected graphs?



P(c)
.50

C	P(s|C)
T	.10
F	.50

C	P(r|C)
T	.80
F	.20

S	R	P(w|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

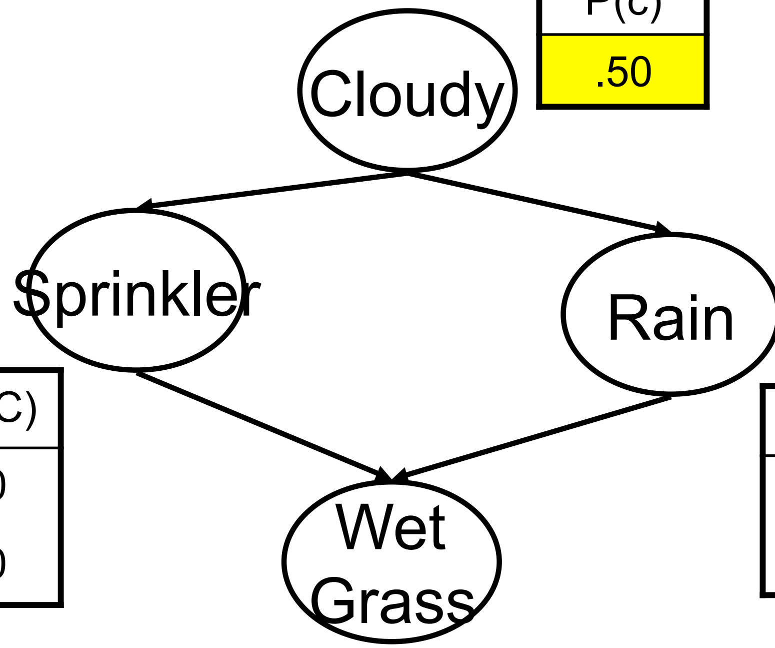
Approximate Inference to the rescue!

Approximate Inference by Stochastic Simulation

- Basic Idea:
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior (conditional) probability P
  - Show this converges to the true probability P
- Four techniques
  - Direct Sampling
  - Rejection Sampling
  - Likelihood weighting
  - Markov chain Monte Carlo (MCMC)

Sampling Basics: Sampling from an empty network

P(c)
.50

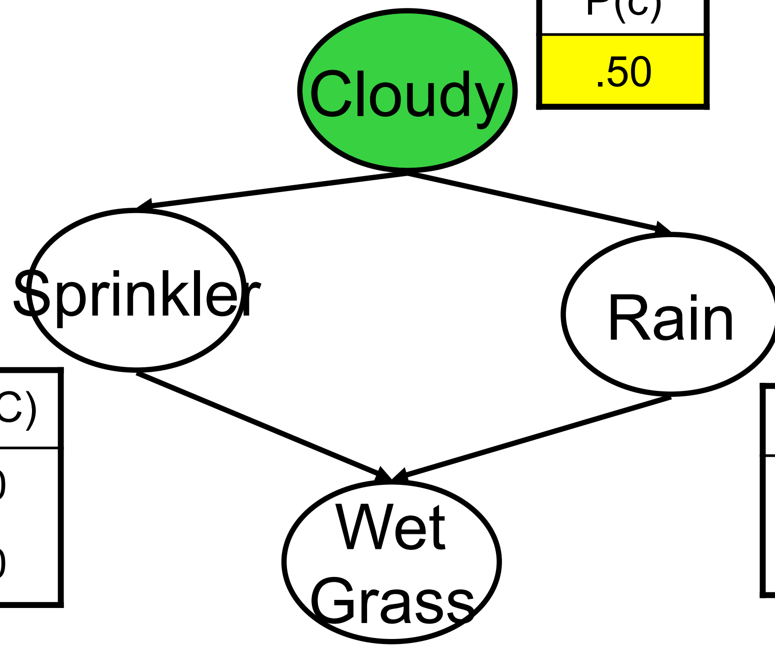


C	P(s|C)
T	.10
F	.50

C	P(r|C)
T	.80
F	.20

S	R	P(w|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



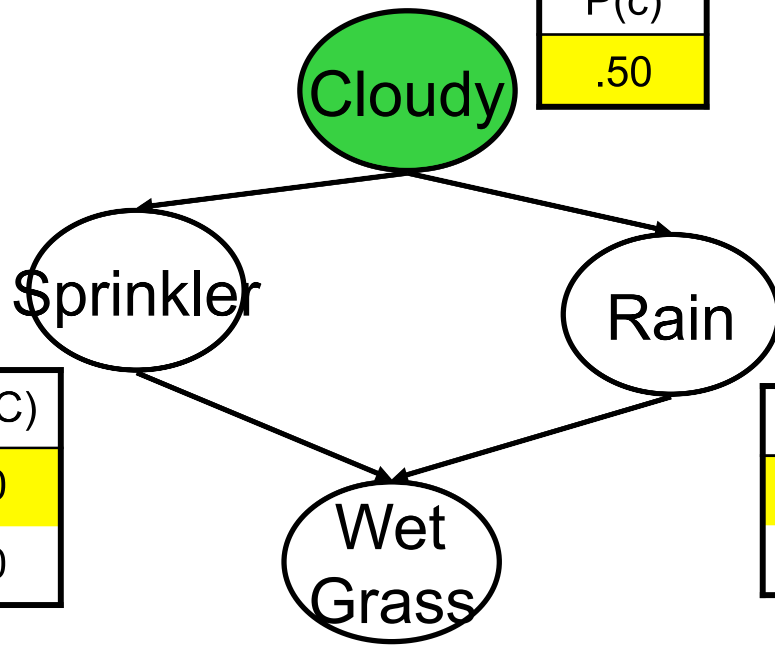
P(c)
.50

C	P(s|C)
T	.10
F	.50

C	P(r|C)
T	.80
F	.20

S	R	P(w|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



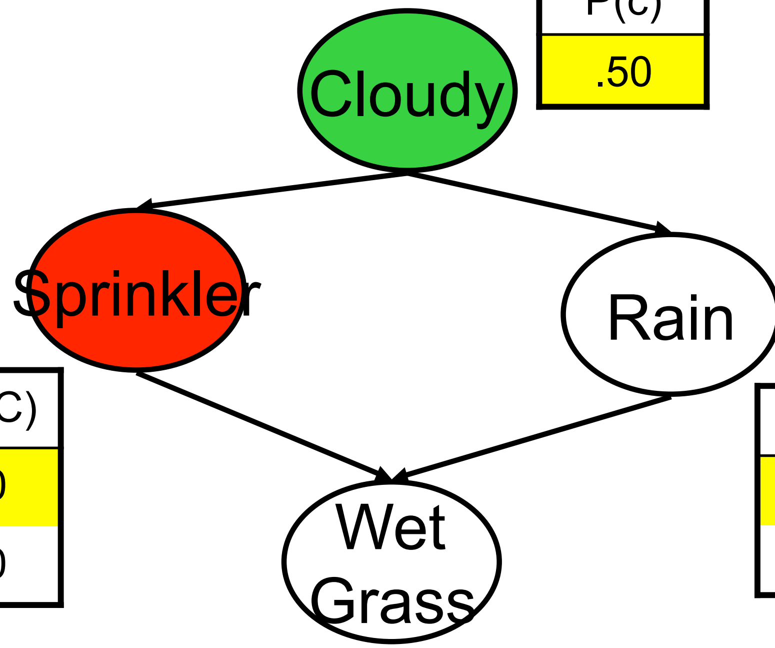
P(c)
.50

C	P(s|C)
T	.10
F	.50

C	P(r|C)
T	.80
F	.20

S	R	P(w|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



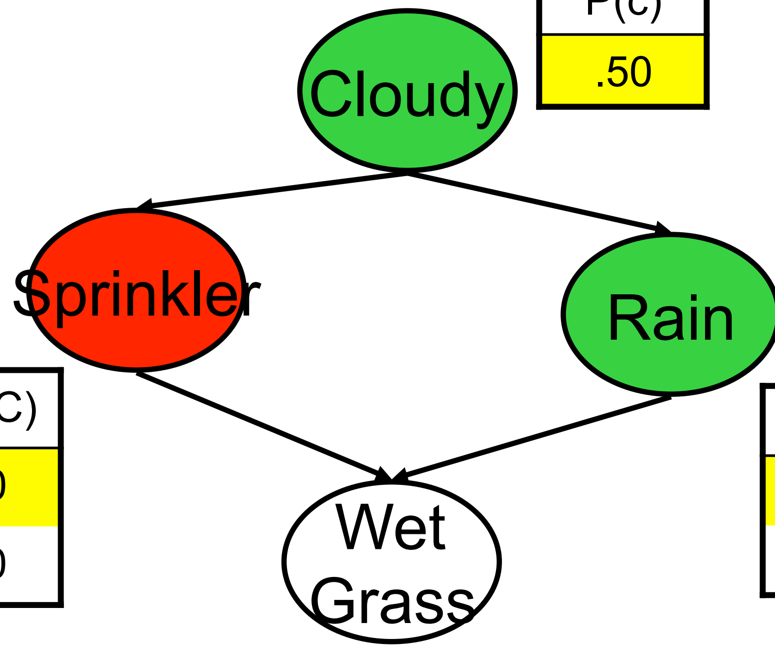
P(c)
.50

C	P(s|C)
T	.10
F	.50

C	P(r|C)
T	.80
F	.20

S	R	P(w|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



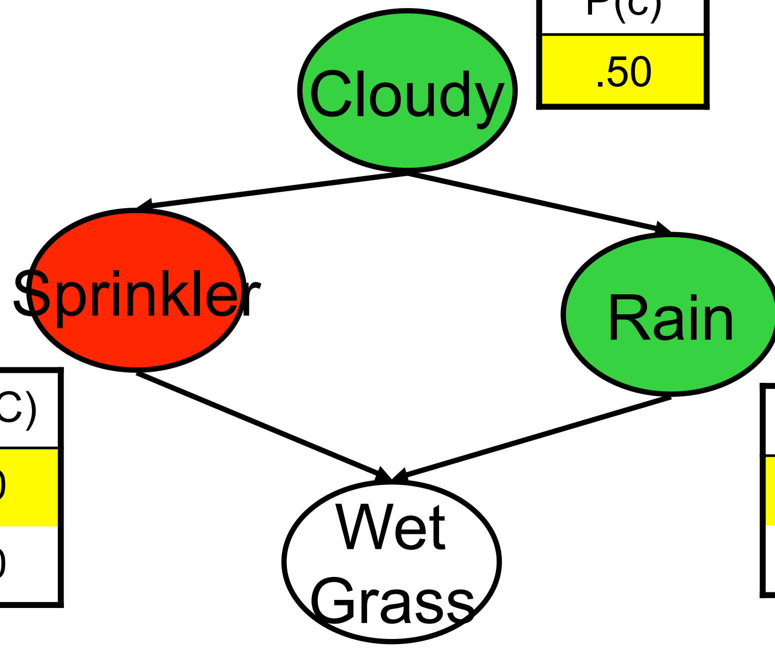
P(c)
.50

C	P(s|C)
T	.10
F	.50

C	P(r|C)
T	.80
F	.20

S	R	P(w|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



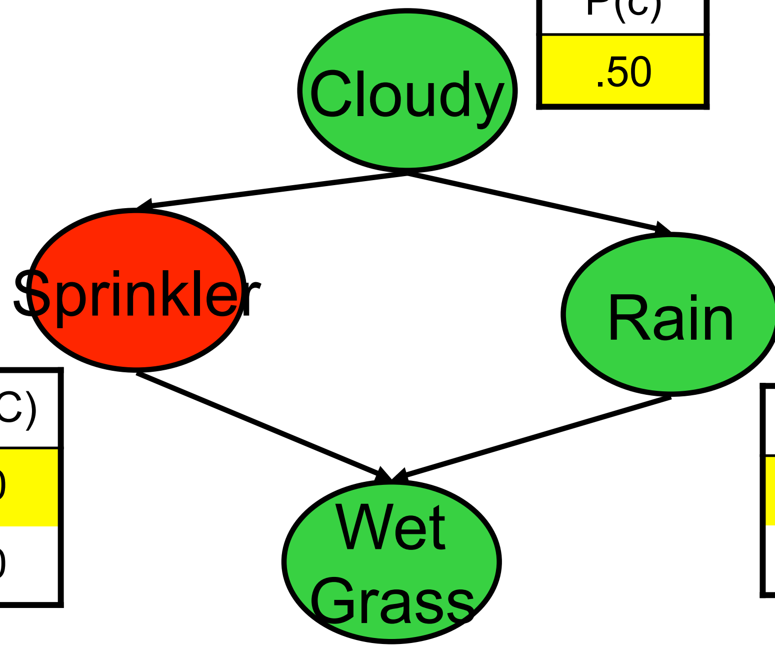
P(c)
.50

C	P(s|C)
T	.10
F	.50

C	P(r|C)
T	.80
F	.20

S	R	P(w|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



P(c)
.50

C	P(s|C)
T	.10
F	.50

C	P(r|C)
T	.80
F	.20

S	R	P(w|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sample: [T, F, T, T]

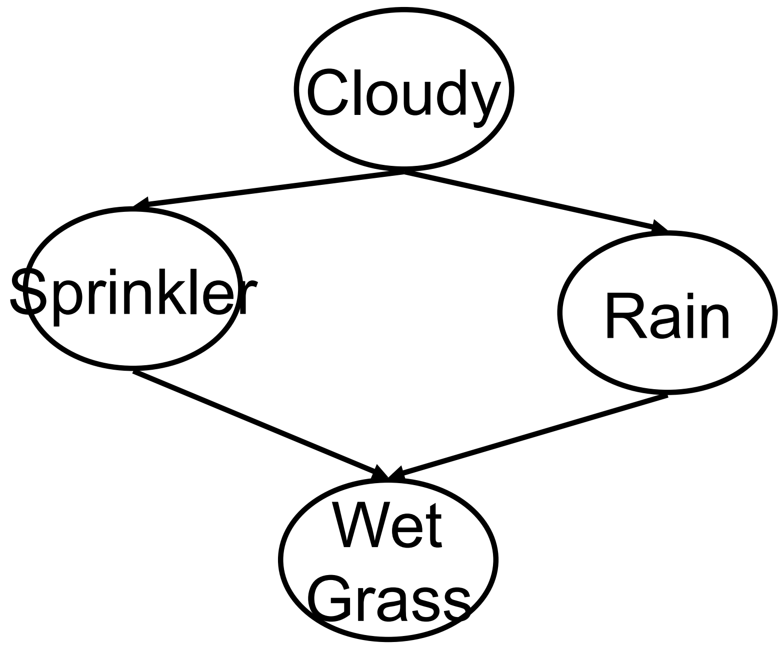
Adding Evidence: Rejection Sampling

$\hat{P}(X \mid \mathbf{e})$ estimated from samples agreeing with e

E.g. Estimate P(R|s)

Samples ([C, S, R, W]):

[T, T, F, T]
[F, F, F, F]
[F, T, F, T]
[F, F, T, T]
[T, F, F, F]
[T, T, F, T]
[F, T, F, T]
[T, F, F, F]
[F, T, T, F]
[T, T, F, F]



Problem with Rejection Sampling?

Likelihood weighting

1) Sample from P(Cloudy) = <0.5, 0.5>; supposed this returns true.
2) Sprinkler is an evidence variable with value true:

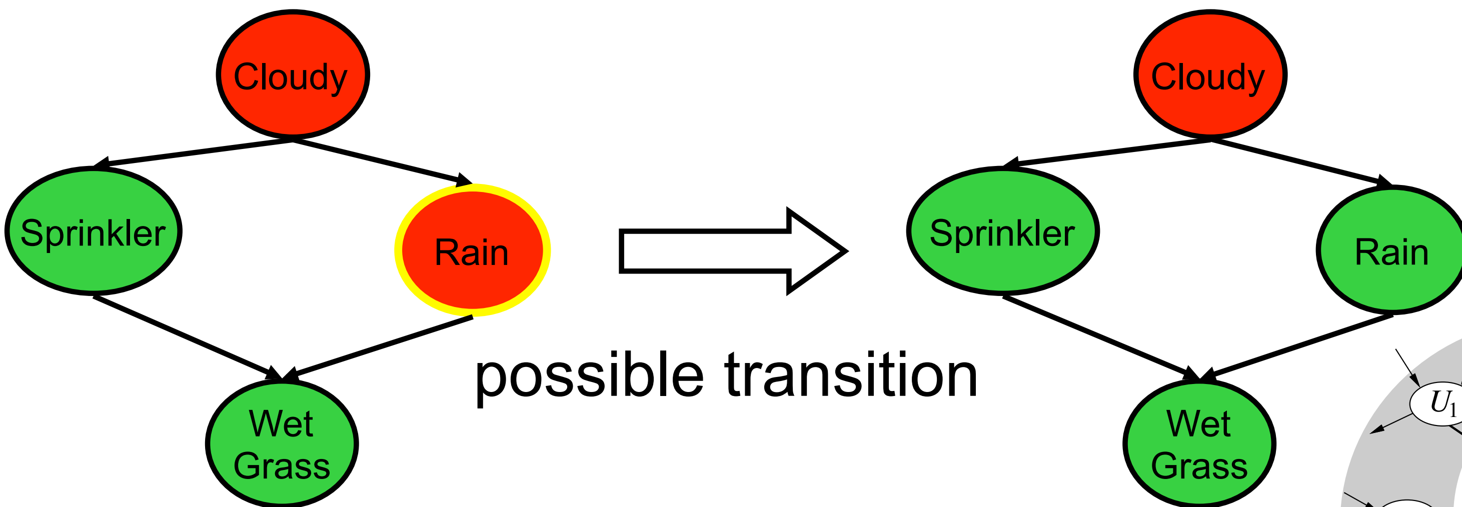
   w = w x P(Sprinkler=true | Cloudy=true) = 0.1

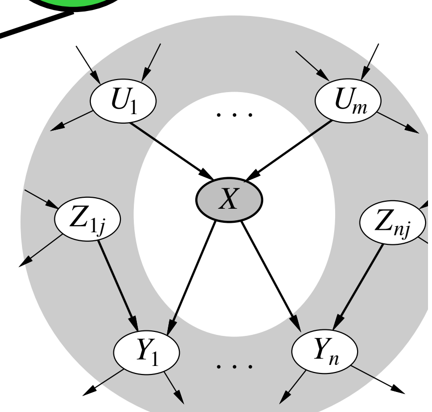
3) Sample from P(Rain|Cloudy=true) = <0.8,0.2>; suppose this returns true
4) WetGrass is an evidence variable with value true,

   w = w x P(WetGrass=true|Sprinkler=true,Rain=true) = 0.099

Approximate Inference using MCMC

- MCMC = Markov chain Monte Carlo
- Idea: Rather than generate individual samples, transition between "states" of the network

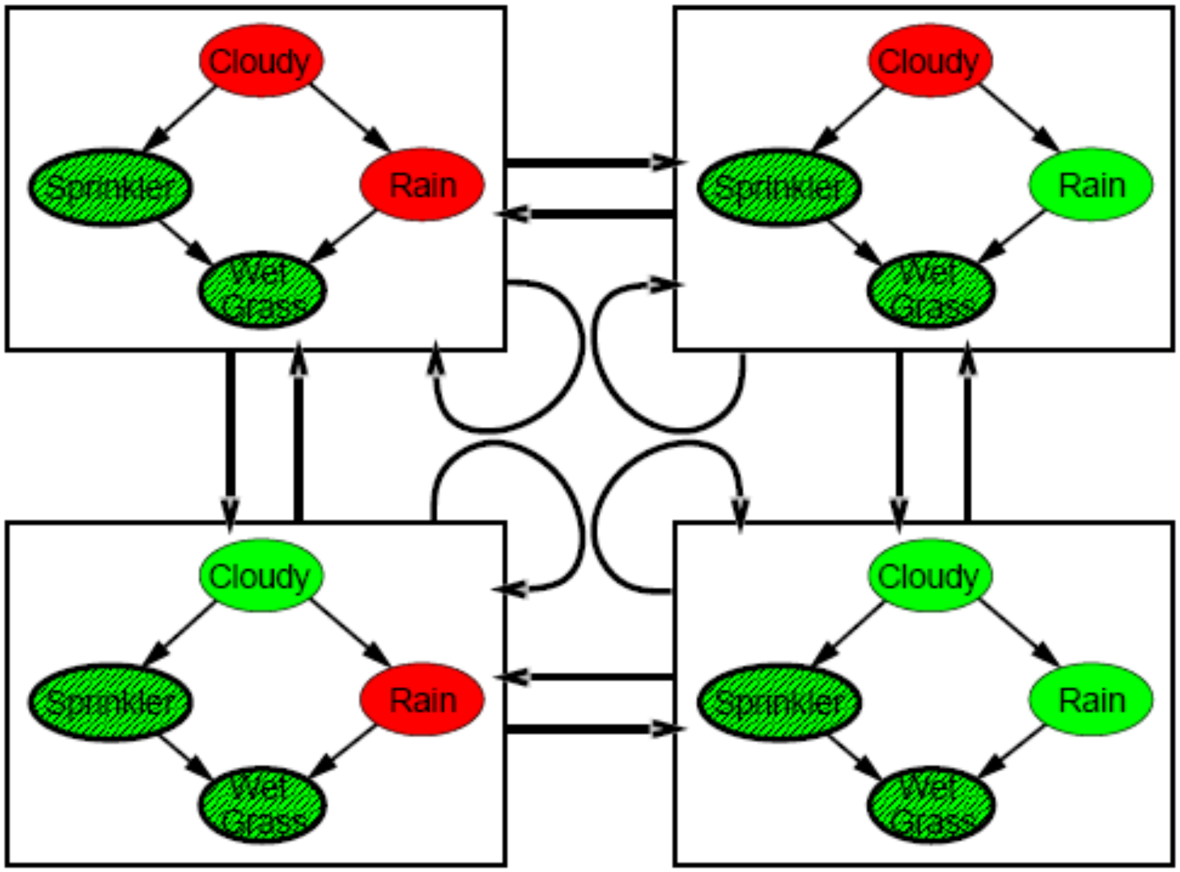




Choose one variable and sample it given its Markov Blanket

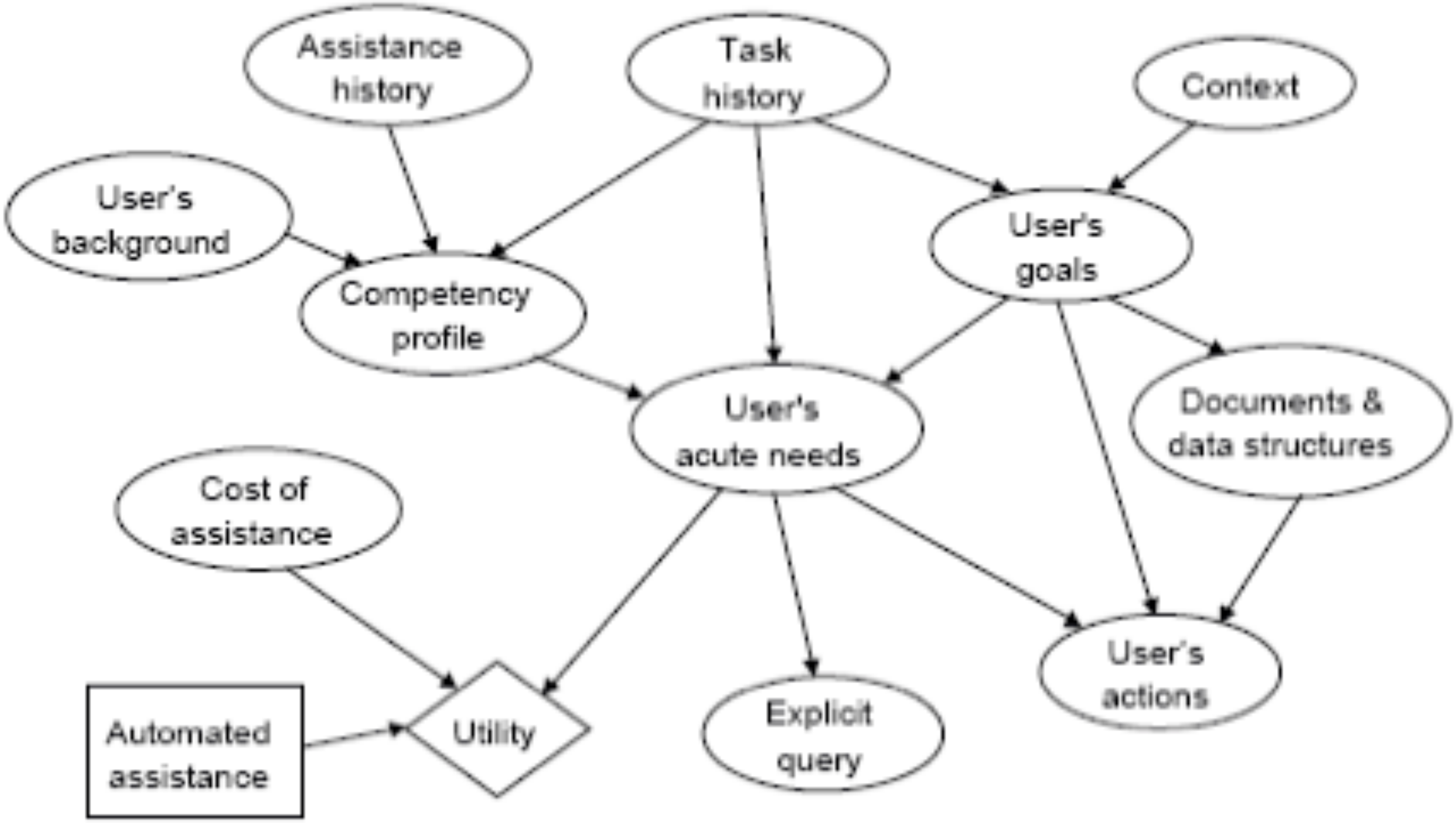
MCMC

If you know Sprinkler=T and Wet Grass=T, there are 4 network state



Wander for awhile, average what you see

Modeling the User's Intentions: The Lumiere Project



Building the Lumiere Bayes Net

- Goal: Build a Bayes net that will model:
  - Whether the user needs assistance
  - What assistance the user needs
  - How useful will this assistance be?

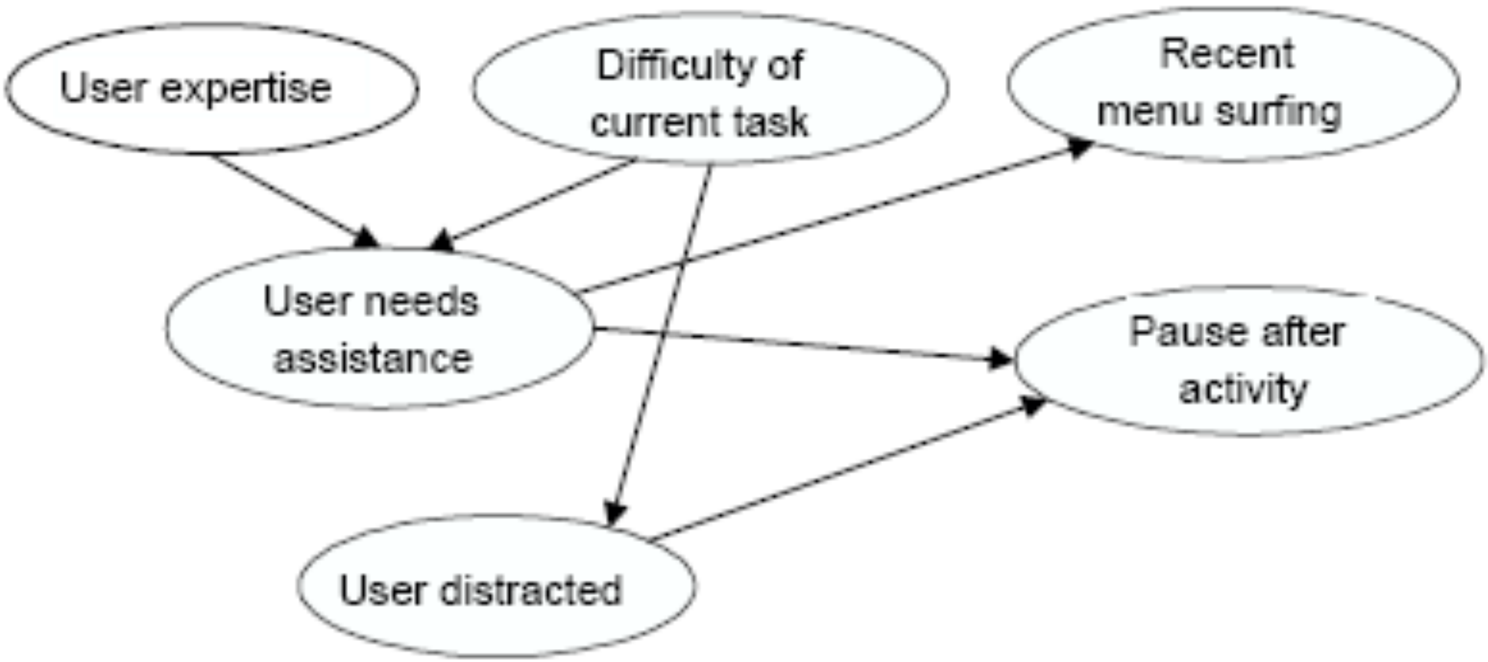
Building the Lumiere Bayes Net

- Approach: Wizard of Oz user study
  - "If experts can't tell when user needs help, how can a computer possibly tell!"
  - Experts watch the user, offer help
  - Results:
    - Experts could do it, but it was hard
    - Poor advice is costly—people took this advice seriously (leading to bad feedback loop)

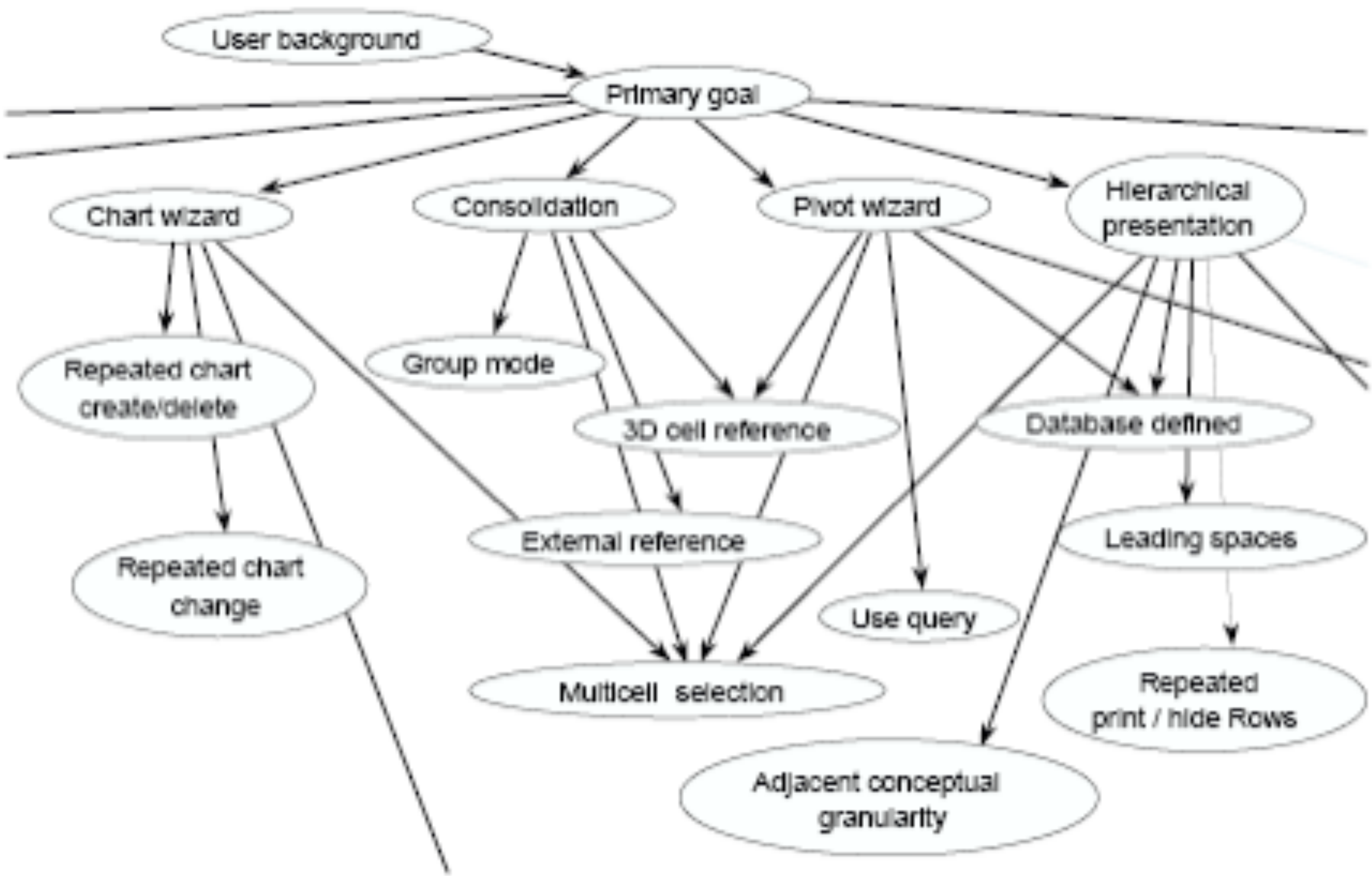
Actions with Relevance to Needs

- Search:
  - Exploring menus, scrolling through text, mousing over regions
- Focus of attention:
  - Selection and dwelling on text or graphical objects
- Introspection:
  - Sudden pause after period of activity
- Undesired effects:
  - Undo, closing a dialog box, undoing an action by hand
- Inefficient control sequences
- Domain-specific syntactic and semantic content

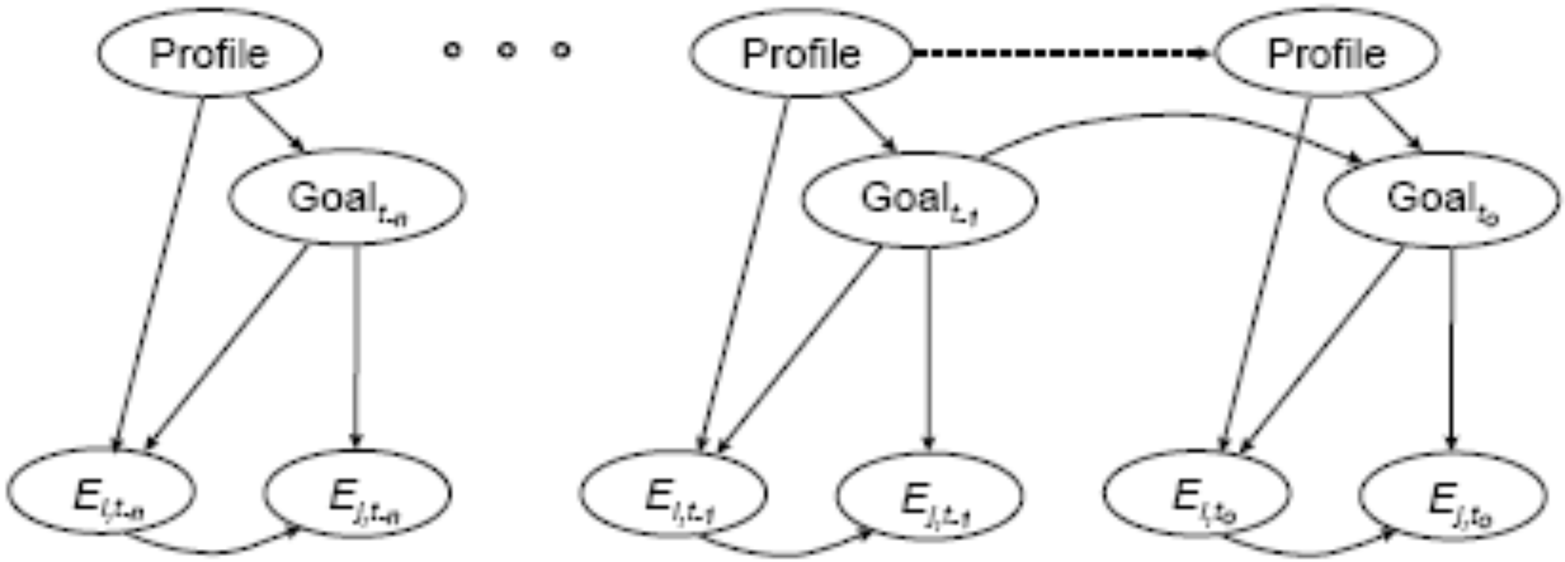
Building the Bayesian network



Building the Bayesian network

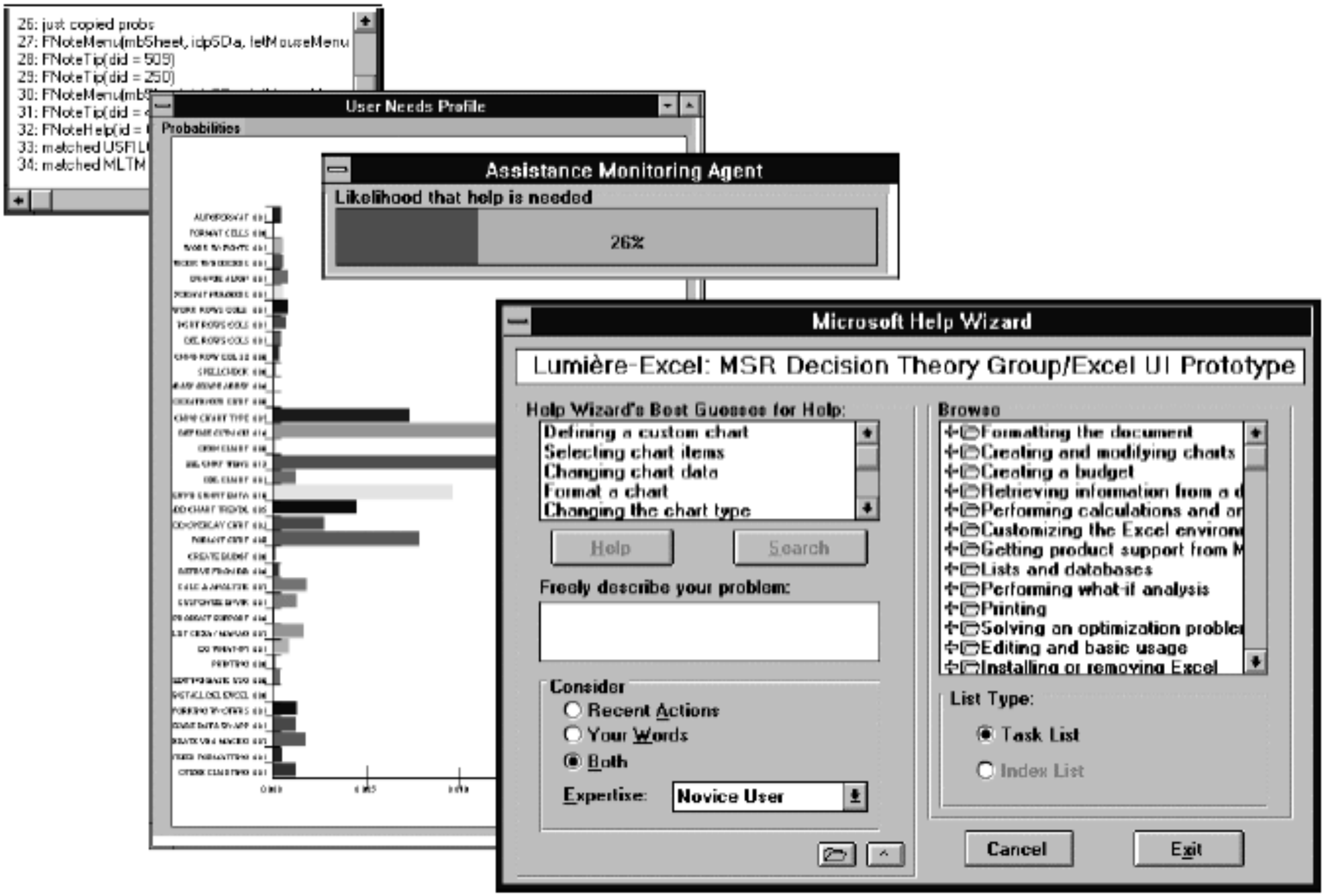


Representing time…



More on this soon…

Lumiere + Excel



Issues

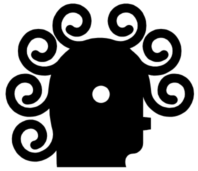
- How to get user model right for a given user?
- How to adapt the model in response to user actions?
- How to display information to the user?

Temporal Reasoning Problem

- Static Reasoning – diagnosis of a car
- Temporal Reasoning –
  - Treating a diabetic patient
  - Speech Recognition

Question: Where is Tracy spending the night?

- Tracy has a new boyfriend, and she has been known to spend the night at his place
- When she spends the night at his place, I often observe that her hair is messy (he doesn't have a blow dryer) but sometimes she oversleeps when she is at home and her hair is mess anyway
- How can we model this as a graphical model (Bayes net)?



Question: Where is Tracy spending the night?

- Suppose I also know that where Tracy spent the night last night affects where she will spend the night tonight.
  - How does this change the model?
  - What problems start to arise?

States and Evidence

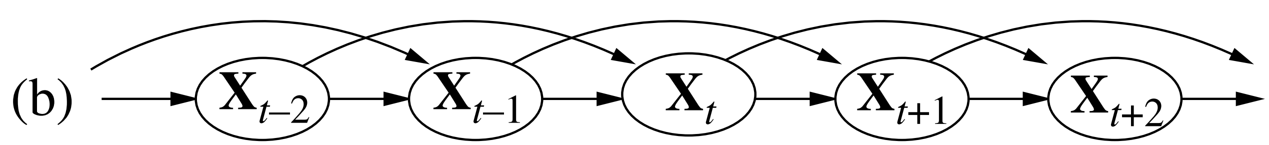
- Reason about a variable X_t given the history of the variable (at times 0:t-1)
  - e.g. Where did Tracy spend the night last (*time 3*) night given she spent Friday (*time 0*) at home and Sat (*time 1*) and Sun (*time 2*) with her boyfriend?

Concept: Markov Chains

- In a Markov chain a variable X_t depends on a bounded subset of $X_{0:t-1}$
- First order Markov Process:

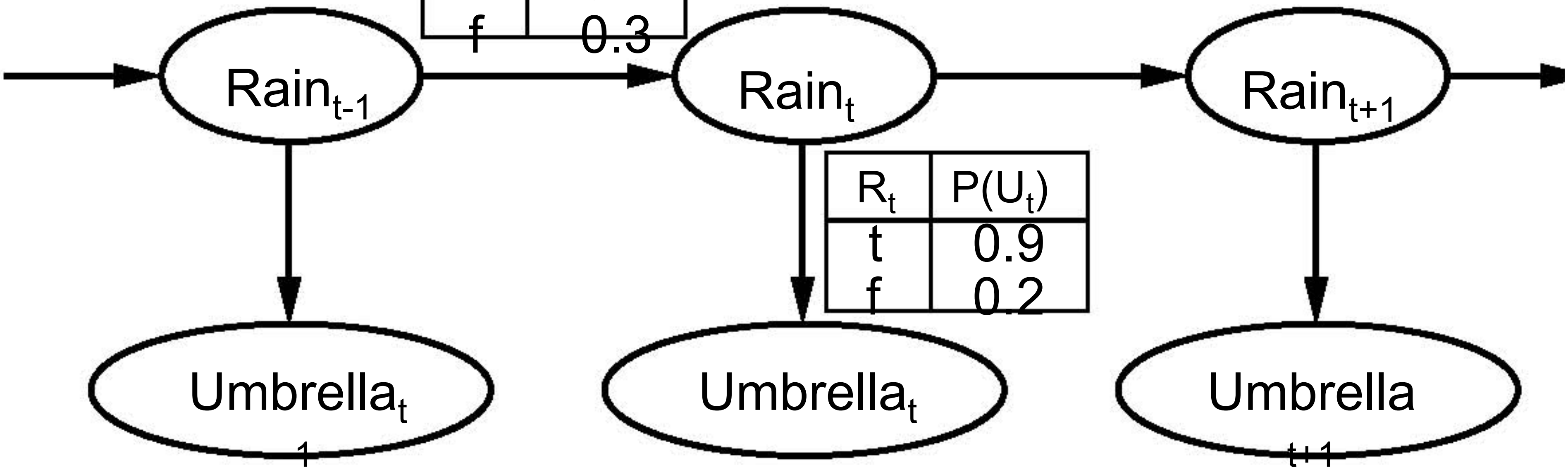
  $P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$
- Second order Markov Process:

  $P(X_t|X_{0:t-1}) = P(X_t|X_{t-1},X_{t-2})$



Markov chains: Adding Evidence

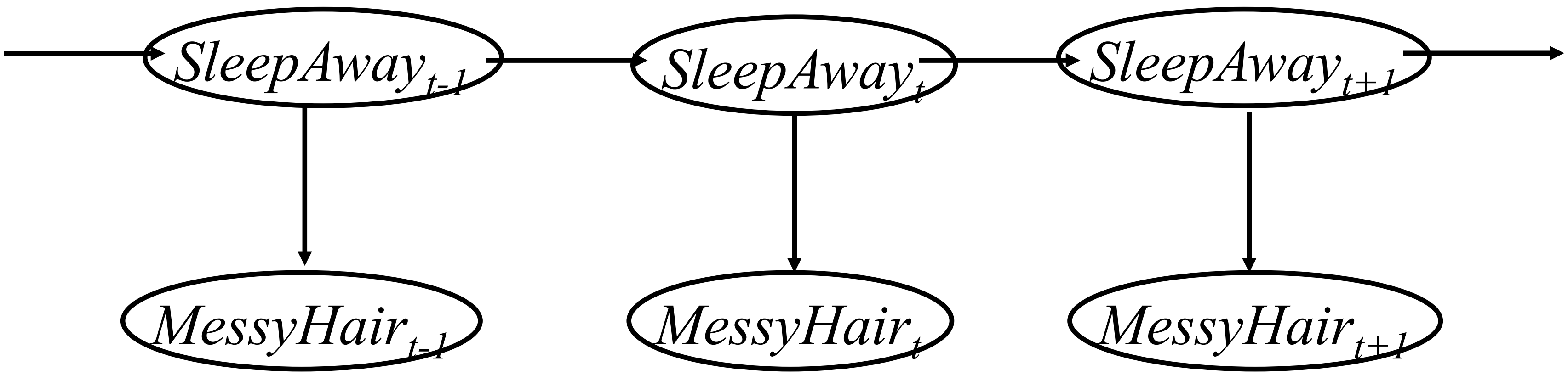
- Markov Models involve two things:
  - transition model: $P(X_t|X_{t-1})$
  - evidence model: $P(E_t|X_t)$
- Sensor Markov assumption: $P(E_t|X_{0:t},E_{0:t-1})=P(E_t|X_t)$
  - e.g., if I know Tracy spent the night at home last night, then the state of her hair does not depend on what her hair looked like on Saturday



R_{t-1}	$P(R_t)$
t	0.7
f	0.3

R_t	$P(U_t)$
t	0.9
f	0.2

$$P(X_0, X_1, \ldots, X_t, E_1, E_2, \ldots E_t) = P(X_0) \prod_{i=1}^{t} P(X_i \mid X_{i-1}) P(E_i \mid X_i)$$



$SleepAway_{t-1}$
$SleepAway_t$
$SleepAway_{t+1}$
$MessyHair_{t-1}$
$MessyHair_t$
$MessyHair_{t+1}$

Modeling a dynamic world

- We need to track and predict signals that change
- Real-world example:
  - Speech recognition
- Issue: what is a "step"?

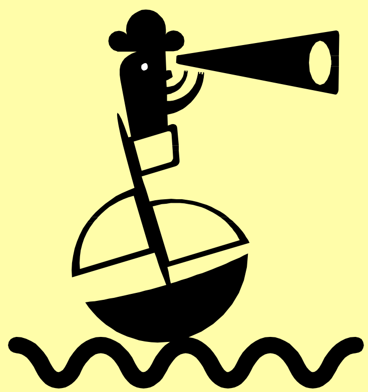
Inference Tasks

- What might we want to do with this model?

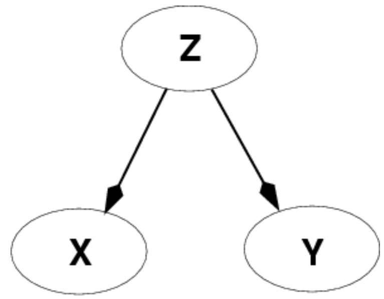
Inference Tasks

- Filtering: $P(X_t|e_{0:t})$
  - Decision making in the here and now
- Prediction: $P(X_{t+k}|e_{0:t})$
  - Trying to plan the future
- Smoothing: $P(X_k|e_{0:t})$ for $0<=k<t$
  - "Revisionist history" (essential for learning)
- Most Likely Explanation (MLE): $\text{argmax}_{x1:t}P(x_{1:t}|e_{1:t})$
  - e.g., speech recognition

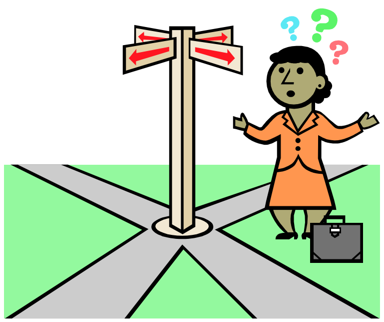
Techniques for Implementing Policies



Search

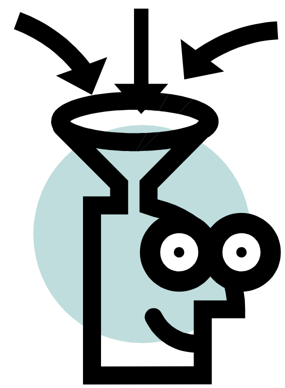


Reasoning with knowledge and uncertainty



Reasoning with Utility

At the core of this class will be several techniques for Policy design and implementation



Learning