Dynamic Bayes Nets

An Approach to Temporal Reasoning and Planning under Uncertainty

Motivation

- Bayes Nets worked in the context of *static* worlds.
- Imagine treating a diabetic patient. Dynamic aspects of the problem, such as blood sugar levels and measurements thereof, change rapidly over time.
- How can we model dynamic processes?

States and Observations

- View the process of change as a series of snapshots, or time slices.
- Each time slice contains random variables, some of which are observable and some of which are not.
- Evidence (observable) variables at time t are denoted by $\mathbf{E}_t$ and state (unobservable) variables are denoted by $\mathbf{X}_t$.

Example

- You are a security guard at a secret underground installation. You want to know if it is raining today, but your only access to the outside world occurs each morning when you see the directory coming in with or without an umbrella.
- For each day t, the set $\mathbf{E}_t$ contains a single evidence variable U_t (whether or not the

Example (Continued)

- (Continued)… umbrella appears), and the set $\mathbf{X_t}$ contains the single state variable R_t (whether or not it is raining).
- For simplicity, we often assume everything starts at t=0.

General Approach and Problems

- Now that we have random variables, we might now reason about conditional independencies and *construct a Bayes Net*.
- **Problem**: the set of variables is unbounded, since we may have arbitrarily many time slices.
- Number of CPTs is therefore unbounded, and the number of parents for a single CPT might be unbounded.

Stationary Process Assumption

- We assume the changes in world state are caused by a *stationary process:* change is governed by laws that themselves do not change. (Different from a static process, in which no change occurs at all).
- Therefore, we need only specify CPTs for a *representative* time slice -- solves the problem of infinitely many CPTs to specify.

Markov Assumption

- *The current state depends only on a **finite** history of previous states.*
- Solves the problem of infinitely many parents for a CPT.
- A process obeying the Markov assumption is called a Markov process or Markov chain.

Order of a Markov Process

- First-order Markov process: a state depends only on the previous state.
- Second-order: a state depends only on the previous two states.
- Etc.

Transition Model

- A transition model describes how the state evolves over time.
- In a first-order Markov Process, the transition model is simply the distribution $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$. In a second-order process, it is $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2},\mathbf{X}_{t-1})$. Etc.

The Evidence Variables

- The Markov Assumption actually is a restriction only on the state variables.
- We assume that the evidence variables for a time slice depend only on (some subset of) the state variables at that time slice.
- Thus we specify a distribution $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$. This conditional distribution is called the *sensor model*.

Types of Inference

- **Filtering (monitoring):** computing the posterior distribution over the current state given all the evidence to date **--** $\mathbf{P}(\mathbf{X}_t|\mathbf{E}_{1:t})$.
- **Prediction:** computing the posterior distribution over a *future* state given all the evidence to date **--** $P(\mathbf{X}_{t+k}|\mathbf{E}_{1:t})$.
- **Smoothing (hindsight):** computing the posterior distribution over a *past* state given

Types of Inference (Continued)

- (Continued)… all the evidence to date (to the present) **--** $\mathbf{P}(\mathbf{X_k}|\mathbf{E_{1:t}})$.
- **Most likely explanation:** Given a sequence of observations, what is the most likely sequence of states that generated those observations -- $\mathrm{ARGMAX}_{X_{1:t}}\, P(\mathbf{X_{1:t}}|\mathbf{E_{1:t}})$.

Examples of the Inference Types

- Filtering: What is the probability that it is raining today, given all the umbrella observations up through today?
- Prediction: What is the probability that it will rain the day after tomorrow, given all the umbrella observations up through today?
- Smoothing: What is the probability that it

Examples (Continued)

- (Continued)… rained yesterday, given all the umbrella observations through today?
- Most likely explanation: if the umbrella appeared the first three days but not on the fourth, the most likely explanation is that it rained the first three days but not on the fourth.

Fifth Inference Type: Learning

- We may wish to learn transition and sensor models from data (either just the CPTs or the arcs as well).
- Use EM algorithm (we won’t cover this application).
- Use with smoothing rather than merely filtering -- otherwise, it won’t converge.

Contrasting Forward and Backward

- “Forward” and “Backward” are somewhat poor names.
- If we run the forward algorithm recursively rather than iteratively, it’s still the forward algorithm but it looks backward rather than forward.
- If we run the backward algorithm recursively rather than iteratively, it looks forward rather than backward.

Contrasting Forward and Backward (Continued)

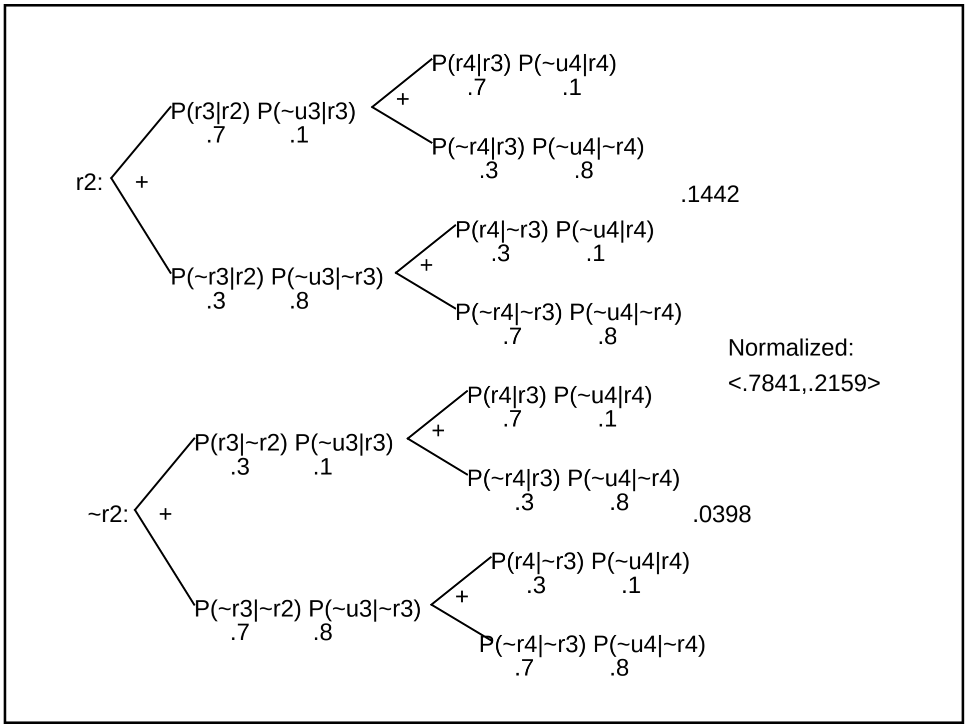
- The distinction is subtle because both algorithms take all the *same paths* through the possible state settings.
- But they group these paths differently and hence give different results.
- Key point: *forward* gives probability of a *state given the evidence* (up to the state), while *backward* gives probability of (subsequent) *evidence given a state*.

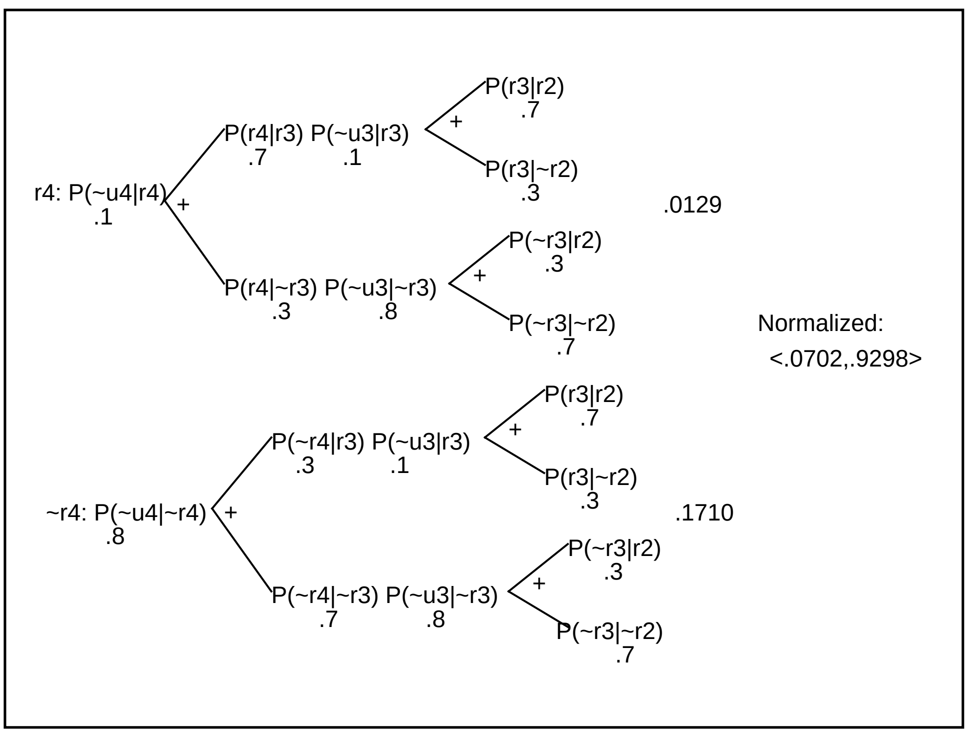
Contrasting Forward and Backward (Continued)

- Easiest to see with an example given the same evidence. Let's use our running example.
- Suppose our evidence is *u1, u2, ~u3, ~u4*. We've already examined filtering to predict rain at time 2 (based on *u1, u2*). This leaves us with a distribution over rain = <*true*,*false*> of <.883,.117>.

Forward and Backward (Continued)

- The next two slides compare the action of the forward and backward algorithms over the same span (evidence at times t=3 and t=4).
- The first slide is the backward algorithm, and the second is the forward algorithm (they appear reversed because we're using the recursive versions).


r2:
+
P(r3|r2) P(~u3|r3)
.7 .1
P(r4|r3) P(~u4|r4)
.7 .1
P(~r4|r3) P(~u4|~r4)
.3 .8
P(~r3|r2) P(~u3|~r3)
.3 .8
P(r4|~r3) P(~u4|r4)
.3 .1
P(~r4|~r3) P(~u4|~r4)
.7 .8
.1442
Normalized:
<.7841,.2159>
~r2:
P(r3|~r2) P(~u3|r3)
.3 .1
P(r4|r3) P(~u4|r4)
.7 .1
P(~r4|r3) P(~u4|~r4)
.3 .8
P(~r3|~r2) P(~u3|~r3)
.7 .8
P(r4|~r3) P(~u4|r4)
.3 .1
P(~r4|~r3) P(~u4|~r4)
.7 .8
.0398


r4: P(~u4|r4)
.1
+
P(r4|r3) P(~u3|r3)
.7 .1
P(r3|r2)
.7
P(r3|~r2)
.3
P(r4|~r3) P(~u3|~r3)
.3 .8
P(~r3|r2)
.3
P(~r3|~r2)
.7
.0129
Normalized:
<.0702,.9298>
~r4: P(~u4|~r4)
.8
P(~r4|r3) P(~u3|r3)
.3 .1
P(r3|r2)
.7
P(r3|~r2)
.3
P(~r4|~r3) P(~u3|~r3)
.7 .8
P(~r3|r2)
.3
P(~r3|~r2)
.7
.1710

Observations

- Both algorithms consider exactly the same "paths" (possible state settings). There is a one-to-one correspondence between the values at the leaves for the for the backward algorithm and those for the forward algorithm.
- Nevertheless, the results are different because the groupings are different.

Dynamic Programming

- Forward, Forward-Backward, and Viterbi algorithms all are instances of Dynamic Programming.
- For more details about Dynamic Programming, take Mark Craven's Bioinformatics Course.
- These algorithms are polynomial in the number of states, though exponential in the

Approximation

- (Continued)… number of state variables.
- We would like an algorithm that is polynomial in the number of state variables.
- This brings us to approximation algorithms.
- Much room for further research. Consider the filtering task in particular.
- Current best approach is *particle filtering*.

Particle Filtering

- Choose sample size N (higher N tends to yield better accuracy but more time.) Draw N initial states according to prior over state.
- For each time step, starting from t = 0:
  - For each state in the sample, draw a corresponding state in the next time step according to the transition probabilities. These new states become the sample at the next time step.

Particle Filtering (Continued)

– Weight each state for the new time slice by the probability it assigns to the evidence for the new time slice.
– Sampling with replacement according to these weights, draw N states from this set of N states to be the new (unweighted) sample.

- Answer query using the last sample. For example, if the query is the probability distribution over states, use the state frequencies in this sample.

AI & Statistics (Looking Back)

- *Probabilistic Reasoning*: Bayes Nets, Utility Theory, Accuracy estimates of learned rules in ILP, Dynamic Bayes Nets.
- *Stochastic Search and Approximation*: Markov Chain Monte Carlo, GSAT and WSAT, Particle Filtering.
- *Knowledge Discovery and Model Building*: ILP for drug design & mutagenicity, much more in the Machine Learning course.

Much More Available

- Bayesian inference: STAT 775
- Bioinformatics, dynamic programming: Bioinformatics (currently a CS 838), also a course by Ann Palmenberg in Biochem
- Machine learning: CS 760
- Neural nets and fuzzy logic: ECE 539
- Computer vision: CS 766
- Robotics: ME 439