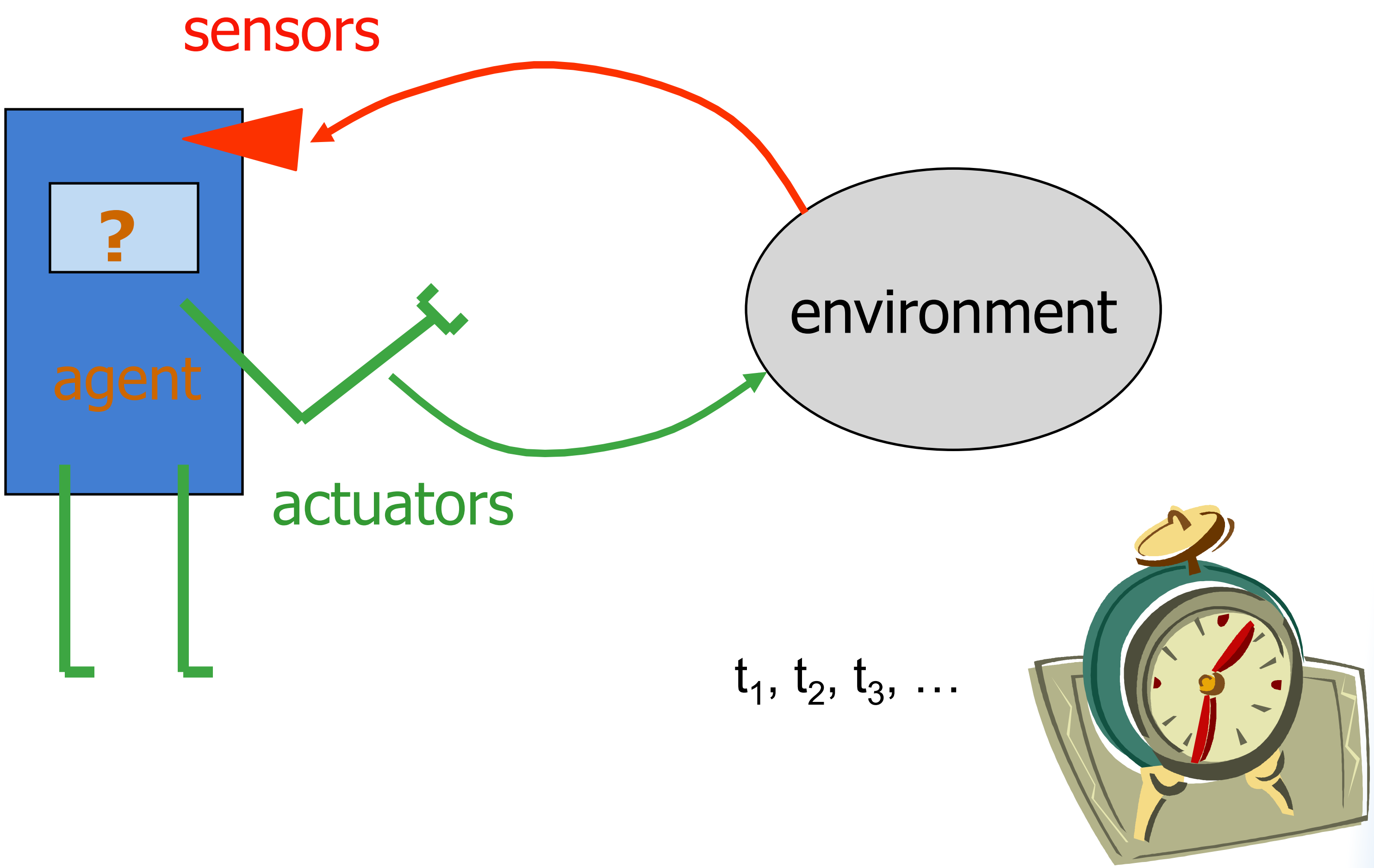
Intelligent Autonomous Agents

Probabilistic Reasoning over Time (Dynamic Bayesian Networks)

Ralf Möller, Rainer Marrone
Hamburg University of Technology

Temporal Probabilistic Agent



Probabilistic Temporal Models

- Dynamic Bayesian Networks (DBNs)
- Hidden Markov Models (HMMs)
- Kalman Filters

Time and Uncertainty

- The world changes, we need to track and predict it
- Examples: diabetes management, traffic monitoring
- Basic idea: copy state and evidence variables for each time step
- $\mathbf{X}_t$ – set of unobservable state variables at time t
  - e.g., $BloodSugar_t$, $StomachContents_t$
- $\mathbf{E}_t$ – set of evidence variables at time t
  - e.g., $MeasuredBloodSugar_t$, $PulseRate_t$, $FoodEaten_t$
- Assumes discrete time steps

States and Observations

- Process of change is viewed as series of snapshots, each describing the state of the world at a particular time
- Each time slice involves a set of random variables indexed by t:
  - the set of unobservable state variables $\mathbf{X}_t$
  - the set of observable evidence variable $\mathbf{E}_t$
- The observation at time t is $\mathbf{E}_t = \mathbf{e}_t$ for some set of values e_t
- The notation $\mathbf{X}_{a:b}$ denotes the set of variables from $\mathbf{X}_a$ to $\mathbf{X}_b$

Dynamic Bayesian Networks

- How can we model *dynamic* situations with a Bayesian network?

- Example: *Is it raining today?*

$$X_t = \{R_t\}$$

$$E_t = \{U_t\}$$

⇨ next step: specify dependencies among the variables.

The term "dynamic" means we are modeling a dynamic system, not that the network structure changes over time.

DBN – Representation

- Problem:

  1. Necessity to specify an unbounded number of conditional probability tables, one for each variable in each slice,
  2. Each one might involve an unbounded number of parents.

- Solution:

  1. Assume that changes in the world state are caused by a stationary process (unmoving process over time).

$$P(U_t / Parent(U_t))$$ is the same for all *t*

DBN – Representation

- Solution cont.:

2. Use **Markov assumption** - The current state depends on only in a finite history of previous states.

   Using the first-order Markov process:

$$P(X_t / X_{0:t-1}) = P(X_t / X_{t-1})$$

Transition Model

In addition to restricting the parents of the state variable *Xt*, we must restrict the parents of the evidence variable *Et*

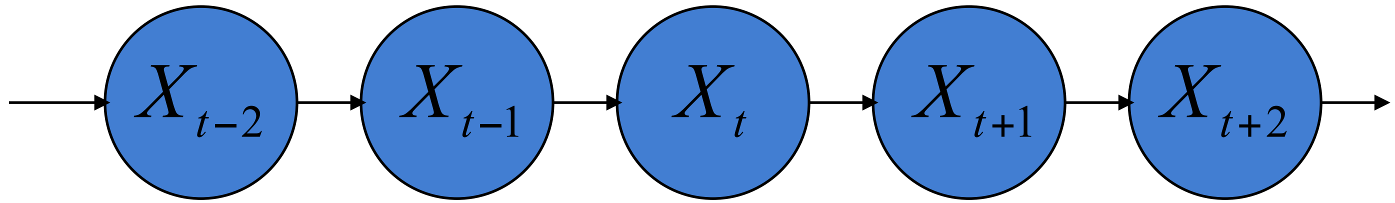
$$P(E_t / X_{0:t}, E_{0:t-1}) = P(E_t / X_t)$$

Sensor Model

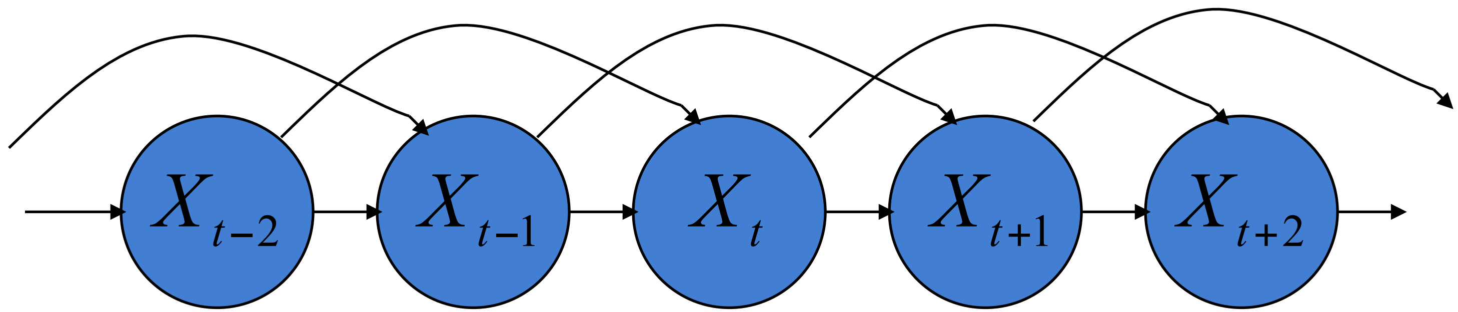
Stationary Process/Markov Assumption

- Markov Assumption: X_t depends on some previous X_is
- First-order Markov process:
  $P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$
- kth order: depends on previous k time steps
- Sensor Markov assumption:
  $P(E_t|X_{0:t}, E_{0:t-1}) = P(E_t|X_t)$
- Assume stationary process: transition model $P(X_t|X_{t-1})$ and sensor model $P(E_t|X_t)$ are the same for all t
- In a **stationary process**, the changes in the world state are governed by laws that do not themselves change over time

Dynamic Bayesian Network



Bayesian network structure corresponding to a first-order of Markov process with state defined by the variables Xt.



A second order of Markov process

Dynamic Bayesian Networks

- There are two possible fixes if the approximation is too inaccurate:
  - Increasing the order of the Markov process model. For example, adding $Rain_{t-2}$ as a parent of $Rain_t$, which might give slightly more accurate predictions.
  - Increasing the set of state variables. For example, adding $Season_t$ to allow to incorporate historical records of rainy seasons, or adding $Temperature_t$, $Humidity_t$ and $Pressure_t$ to allow to use a physical model of rainy conditions.

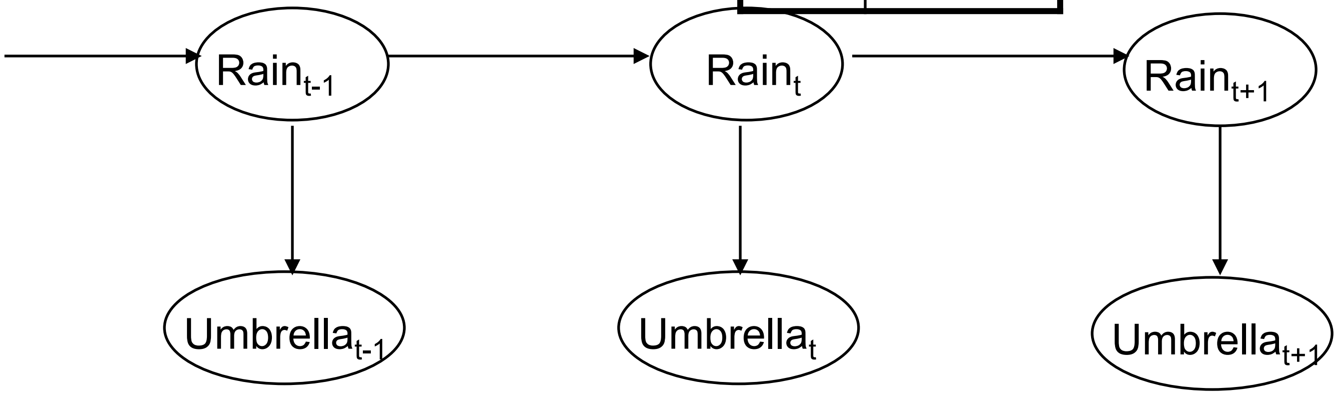
Complete Joint Distribution

- Given:
  - Transition model: $P(X_t|X_{t-1})$
  - Sensor model: $P(E_t|X_t)$
  - Prior probability: $P(X_0)$
- Then we can specify complete joint distribution:

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

Example

R_{t-1}	$P(R_t \mid R_{t-1})$
T	0.7
F	0.3



R_t	$P(U_t \mid R_t)$
T	0.9
F	0.2

Inference Tasks: Examples

- **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 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, what is the most likely weather sequence to produce these umbrella sightings?

DBN – Basic Inference

- Filtering or Monitoring:

Compute the belief state – the posterior distribution over the *current* state, given all evidence to date.

$$P(X_t / e_{1:t})$$

Filtering is what a rational agent needs to do in order to keep track of the current state so that the rational decisions can be made.

DBN – Basic Inference

- Filtering cont.

Given the results of filtering up to time *t*, one can easily compute the result for *t+1* from the new evidence e_{t+1}

$$P(X_{t+1} / e_{1:t+1}) = f(e_{t+1,} P(X_t / e_{1:t+1}))$$ (for some function *f*)

$$= P(X_{t+1} / e_{1:t,} e_{t+1})$$ (dividing up the evidence)

$$= \alpha P(e_{t+1} / X_{t+1,} e_{1:t}) P(X_{t+1} / e_{1:t})$$ (using Bayes' Theorem)

$$= \alpha P(e_{t+1} / X_{t+1}) P(X_{t+1} / e_{1:t})$$ (by the Markov property of evidence)

α is a normalizing constant used to make probabilities sum up to 1.

DBN – Basic Inference

- Filtering cont.

The second term $P(X_{t+1}/e_{1:t})$ represents a one-step prediction of the next step, and the first term $P(e_{t+1}/X_{t+1})$ updates this with the new evidence.

Now we obtain the one-step prediction for the next step by conditioning on the current state Xt:

$$P(X_{t+1}/e_{1:t+1}) = \alpha P(e_{t+1}/X_{t+1})\sum_{X_t} P(X_{t+1}/x_t, e_{1:t})P(x_t/e_{1:t})$$

$$= \alpha P(e_{t+1}/X_{t+1})\sum_{X_t} P(X_{t+1}/\boldsymbol{x_t})P(x_t/e_{1:t})$$

(using the Markov property)

Forward Messages

$\mathbf{f}_{1:t+1} = \textsc{Forward}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$ where $\mathbf{f}_{1:t} = \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$

Time and space **constant** (independent of t)

DBN – Basic Inference

Illustration for two steps in the Umbrella example:

- On day 1, the umbrella appears so U1=true. The prediction from t=0 to t=1 is

$$P(R_1) = \sum_{r_0} P(R_1 / r_0) P(r_0)$$

and updating it with the evidence for t=1 gives

$$P(R_1 / u_1) = \alpha P(u_1 / R_1) P(R_1)$$

- On day 2, the umbrella appears so U2=true. The prediction from t=1 to t=2 is

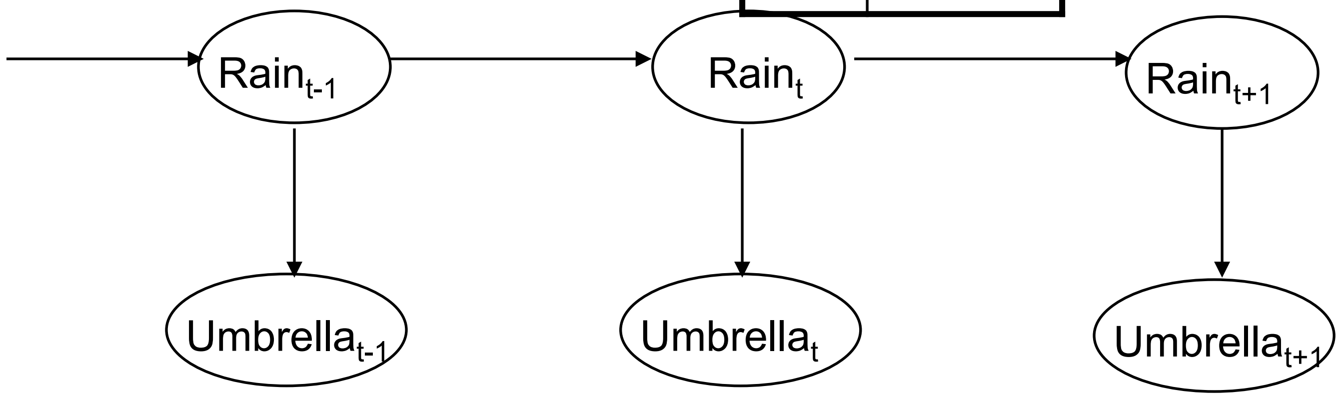
$$P(R_2 / u_1) = \sum_{r_1} P(R_2 / r_1) P(r_1 / u_1)$$

and updating it with the evidence for t=2 gives

$$P(R_2 / u_1, u_2) = \alpha P(u_2 / R_2) P(R_2 / u_1)$$

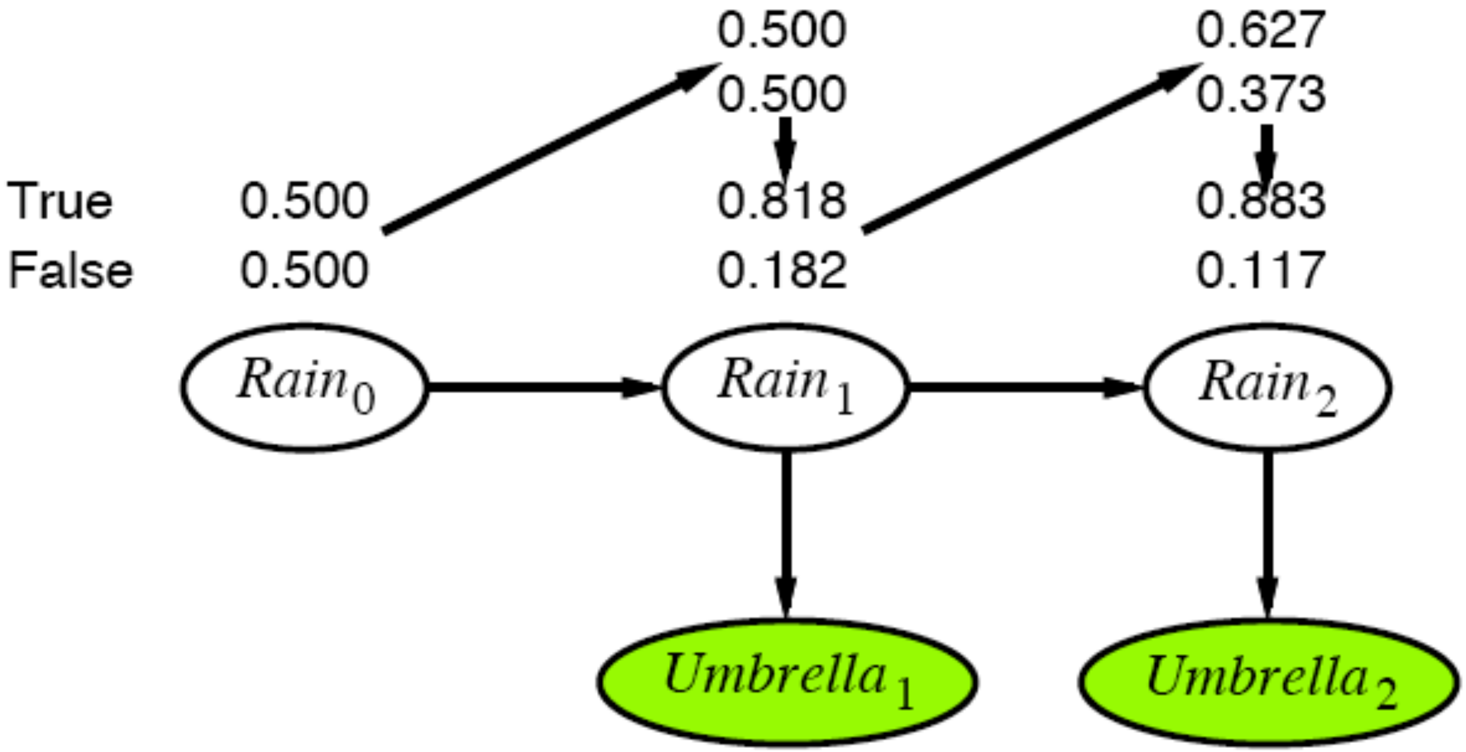
Example

R_{t-1}	$P(R_t \mid R_{t-1})$
T	0.7
F	0.3



R_t	$P(U_t \mid R_t)$
T	0.9
F	0.2

Example cntd.



DBN – Basic Inference

- Prediction:

Compute the posterior distribution over the *future* state, given all evidence to date.

$$P(X_{t+k} / e_{1:t})$$

for some k>0

The task of prediction can be seen simply as filtering without the addition of new evidence.

DBN – Basic Inference

- Smoothing or hindsight:

Compute the posterior distribution over the *past* state, given all evidence up to the present.

$$P(X_k / e_{1:t})$$

for some k such that $0 \le k < t$.

Hindsight provides a better estimate of the state than was available at the time, because it incorporates more evidence.

Smoothing

Divide evidence $\mathbf{e}_{1:t}$ into $\mathbf{e}_{1:k}$, $\mathbf{e}_{k+1:t}$:

$$\begin{aligned}
\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) &= \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k}, \mathbf{e}_{k+1:t}) \\
&= \alpha\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{e}_{1:k}) \\
&= \alpha\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) \\
&= \alpha\mathbf{f}_{1:k}\mathbf{b}_{k+1:t}
\end{aligned}$$

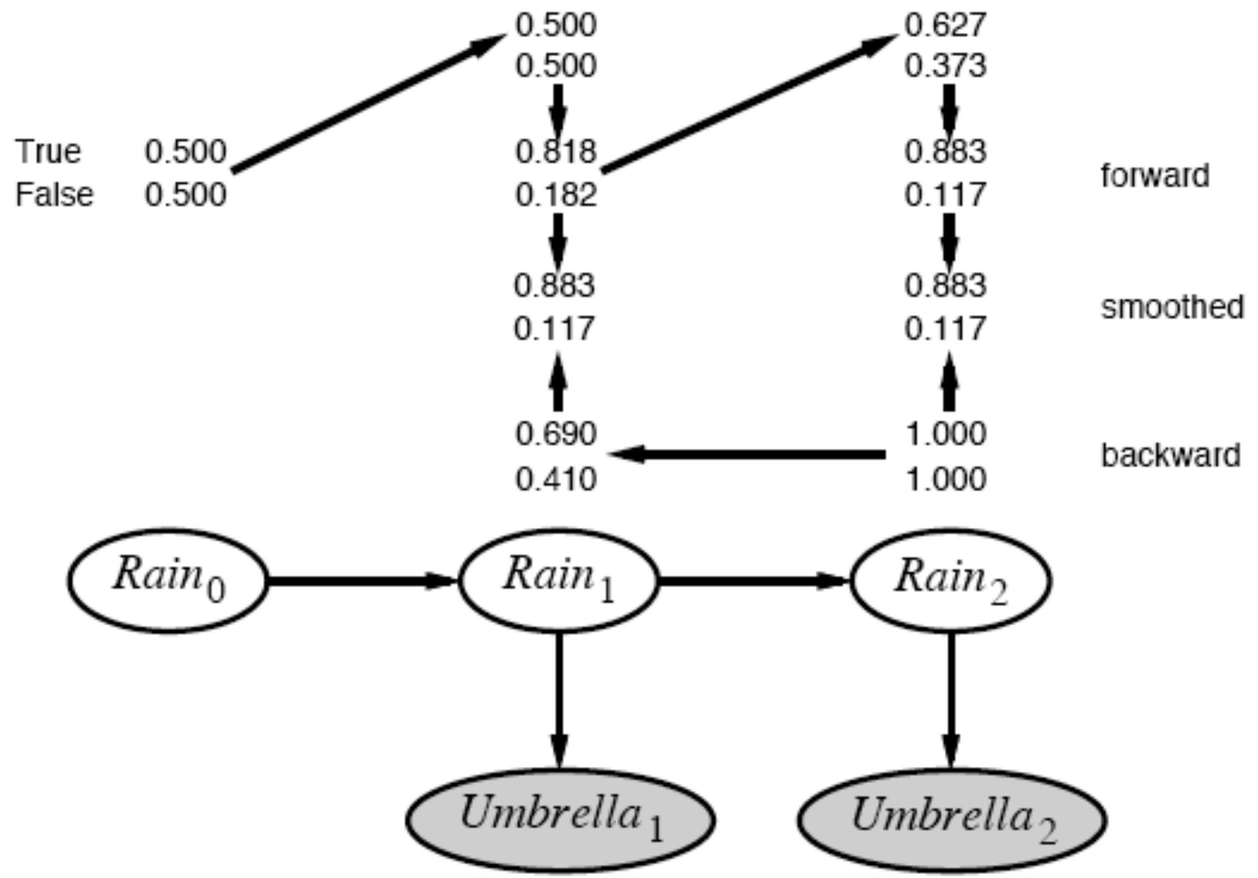
Backward message computed by a backwards recursion:

$$\begin{aligned}
\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) &= \Sigma_{\mathbf{x}_{k+1}}\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\
&= \Sigma_{\mathbf{x}_{k+1}}P(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\
&= \Sigma_{\mathbf{x}_{k+1}}P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)
\end{aligned}$$

Forward–backward algorithm: cache forward messages along the way

Time linear in t (polytree inference), space $O(t|\mathbf{f}|)$

Example contd.



DBN – Basic Inference

- Most likely explanation:

Compute the sequence of states that is most likely to have generated a given sequence of observation.

$$\arg\max_{x_{1:t}} P(X_{1:t} / e_{1:t})$$

Algorithms for this task are useful in many applications, including, e.g., speech recognition.

Most-likely explanation

Most likely sequence $\neq$ sequence of most likely states!!!!

Most likely path to each $\mathbf{x}_{t+1}$
= most likely path to **some** $\mathbf{x}_t$ plus one more step

$$\max_{\mathbf{x}_1 \ldots \mathbf{x}_t} \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{X}_{t+1} | \mathbf{e}_{1:t+1})$$
$$= \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \left(\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) \max_{\mathbf{x}_1 \ldots \mathbf{x}_{t-1}} P(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{e}_{1:t}) \right)$$

Identical to filtering, except $\mathbf{f}_{1:t}$ replaced by

$$\mathbf{m}_{1:t} = \max_{\mathbf{x}_1 \ldots \mathbf{x}_{t-1}} \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{X}_t | \mathbf{e}_{1:t}),$$

I.e., $\mathbf{m}_{1:t}(i)$ gives the probability of the most likely path to state i.
Update has sum replaced by max, giving the Viterbi algorithm:

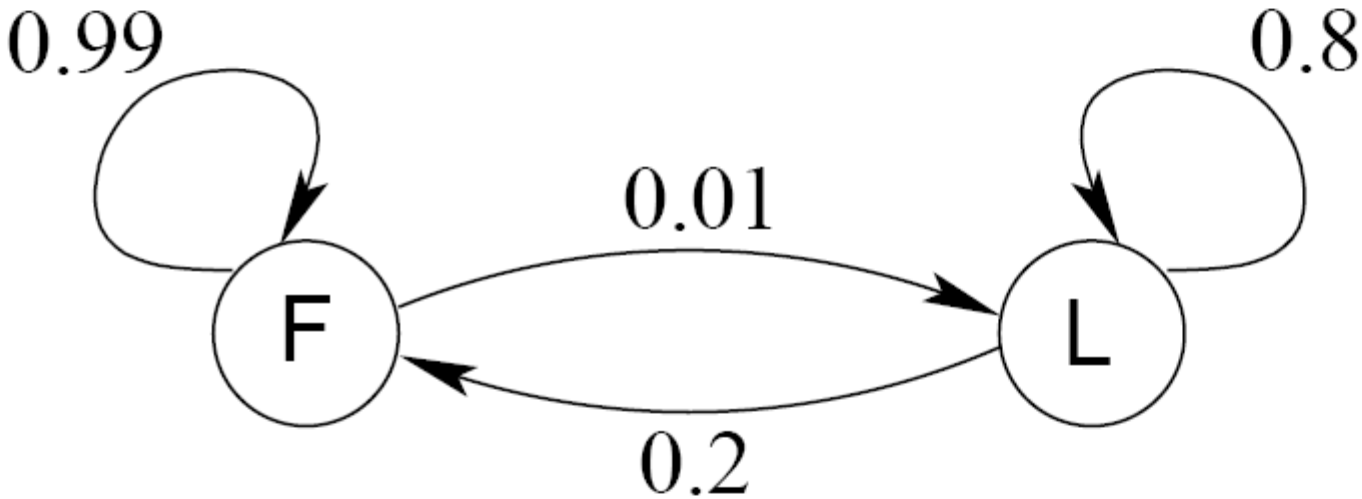
$$\mathbf{m}_{1:t+1} = \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \left(\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) \mathbf{m}_{1:t} \right)$$

The occasionally dishonest casino

- A casino uses a fair die most of the time, but occasionally switches to a loaded one
  - Fair die: Prob(1) = Prob(2) = . . . = Prob(6) = 1/6
  - Loaded die: Prob(1) = Prob(2) = . . . = Prob(5) = 1/10, Prob(6) = ½
  - These are the ***emission*** probabilities
- ***Transition probabilities***
  - Prob(Fair → Loaded) = 0.01
  - Prob(Loaded → Fair) = 0.2
  - Transitions between states obey a Markov process

Slide by Changui Yan

An HMM for the occasionally dishonest casino



Slide by Changui Yan

The occasionally dishonest casino

- Known:
  - The structure of the model
  - The transition probabilities
- Hidden: What the casino did
  - FFFFFLLLLLLLFFFF...
- Observable: The series of die tosses
  - 3415256664666153...
- What we must infer:
  - When was a fair die used?
  - When was a loaded one used?
    - The answer is a sequence FFFFFFFLLLLLLFFF...

Slide by Changui Yan

Making the inference

- Model assigns a probability to each explanation of the observation:

  $P(326|FFL)$
  $= P(3|F) \cdot P(F \rightarrow F) \cdot P(2|F) \cdot P(F \rightarrow L) \cdot P(6|L)$
  $= 1/6 \cdot 0.99 \cdot 1/6 \cdot 0.01 \cdot ½$
- **Maximum Likelihood:** Determine which explanation is most likely
  - Find the path *most likely* to have produced the observed sequence
- **Total probability:** Determine probability that observed sequence was produced by the HMM
  - Consider *all* paths that could have produced the observed sequence

Slide by Changui Yan

Notation

- x is the sequence of symbols emitted by model
  - x_i is the symbol emitted at time i
- A ***path***, π, is a sequence of states
  - The i–th state in π is π_i
- a_{kr} is the probability of making a transition from state k to state r:

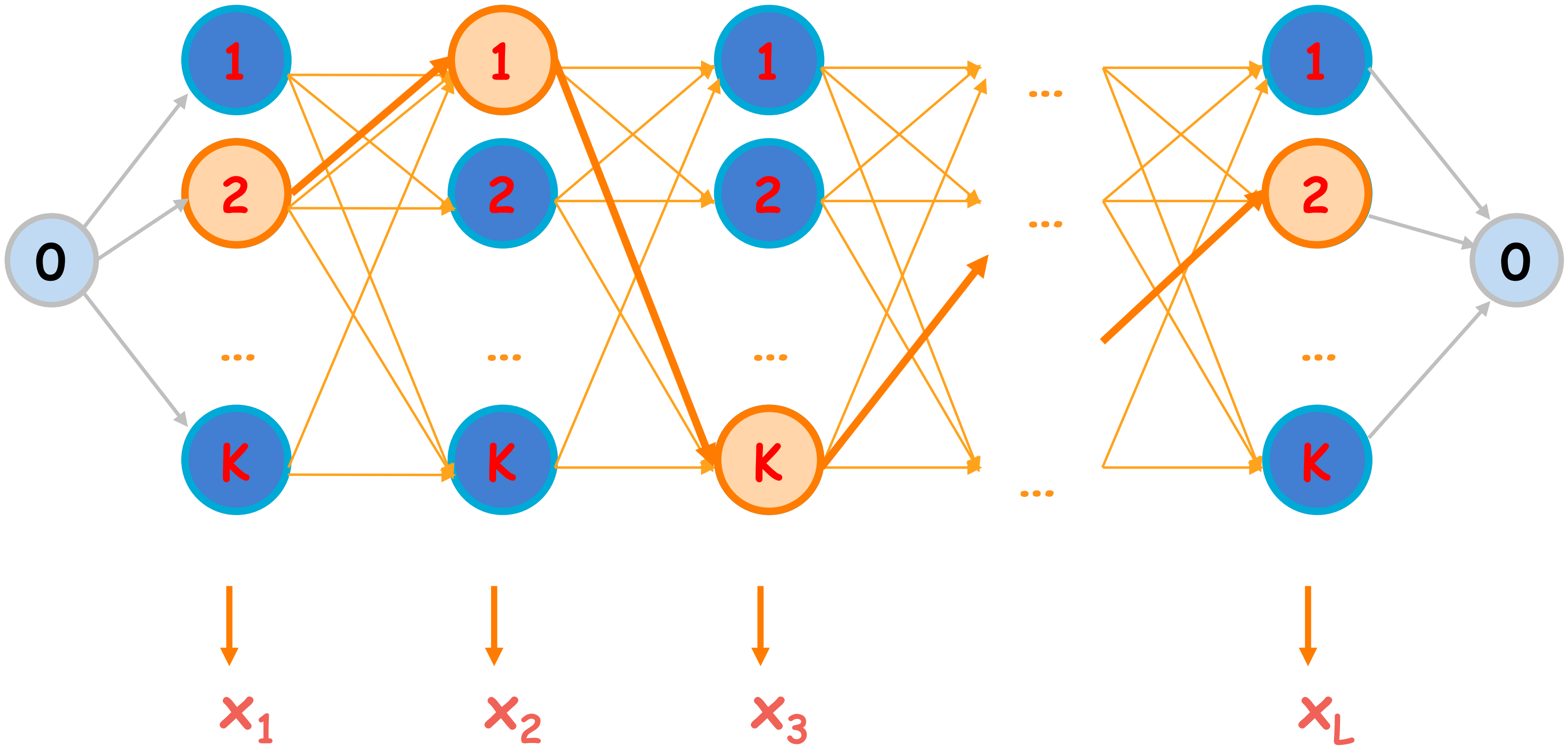
$$a_{kr} = \Pr(\pi_i = r \mid \pi_{i-1} = k)$$

- $e_k(b)$ is the probability that symbol b is emitted when in state k

$$e_k(b) = \Pr(x_i = b \mid \pi_i = k)$$

Slide by Changui Yan

A "parse" of a sequence



$$\Pr(x,\pi) = a_{0\pi_1} \prod_{i=1}^{L} e_{\pi_i}(x_i) \cdot a_{\pi_i \pi_{i+1}}$$

Slide by Changui Yan

The occasionally dishonest casino

$$x = \langle x_1, x_2, x_3 \rangle = \langle 6,2,6 \rangle$$

$$\pi^{(1)} = FFF$$

$$\Pr(x, \pi^{(1)}) = a_{0F} e_F(6) a_{FF} e_F(2) a_{FF} e_F(6)$$
$$= 0.5 \times \frac{1}{6} \times 0.99 \times \frac{1}{6} \times 0.99 \times \frac{1}{6}$$
$$\approx 0.00227$$

$$\pi^{(2)} = LLL$$

$$\Pr(x, \pi^{(2)}) = a_{0L} e_L(6) a_{LL} e_L(2) a_{LL} e_L(6)$$
$$= 0.5 \times 0.5 \times 0.8 \times 0.1 \times 0.8 \times 0.5$$
$$= 0.008$$

$$\pi^{(3)} = LFL$$

$$\Pr(x, \pi^{(3)}) = a_{0L} e_L(6) a_{LF} e_F(2) a_{FL} e_L(6) a_{L0}$$
$$= 0.5 \times 0.5 \times 0.2 \times \frac{1}{6} \times 0.01 \times 0.5$$
$$\approx 0.0000417$$

Slide by Changui Yan

The most probable path

The most likely path π^* satisfies

$$\pi^* = \arg\max_{\pi} \Pr(x, \pi)$$

To find π^*, consider all possible ways the last symbol of **x** could have been emitted

Let

$$v_k(i) = \text{Prob. of path } \langle \pi_1, \cdots, \pi_i \rangle \text{ most likely to emit } \langle x_1, \ldots, x_i \rangle \text{ such that } \pi_i = k$$

Then

$$v_k(i) = e_k(x_i) \max_r \left(v_r(i-1) a_{rk} \right)$$

Slide by Changui Yan

The Viterbi Algorithm

- Initialization ($i = 0$)

$$v_0(0) = 1, \quad v_k(0) = 0 \text{ for } k > 0$$

- Recursion ($i = 1, . . . , L$): For each state k

$$v_k(i) = e_k(x_i)\max_r\left(v_r(i-1)a_{rk}\right)$$

- Termination:

$$\Pr(x,\pi^*) = \max_k\left(v_k(L)a_{k0}\right)$$

To find π^*, use trace-back, as in dynamic programming

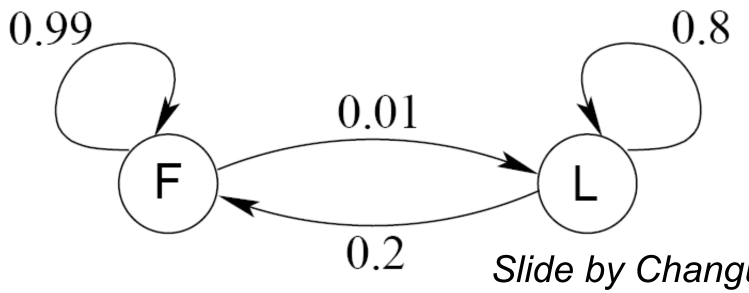
Slide by Changui Yan

Viterbi: Example

x

π		ε	6	2	6
	B	1	0	0	0
	F	0	$(1/6)\times(1/2) = 1/12$	$(1/6)\times\max\{(1/12)\times 0.99, (1/4)\times 0.2\} = 0.01375$	$(1/6)\times\max\{0.01375\times 0.99, 0.02\times 0.2\} = 0.00226875$
	L	0	$(1/2)\times(1/2) = 1/4$	$(1/10)\times\max\{(1/12)\times 0.01, (1/4)\times 0.8\} = 0.02$	$(1/2)\times\max\{0.01375\times 0.01, 0.02\times 0.8\} = 0.08$

$$v_k(i) = e_k(x_i)\max_r\left(v_r(i-1)a_{rk}\right)$$



Slide by Changui Yan

Viterbi gets it right more often than not

Rolls	315116246446644245321131631164152133625144543631656626566666
Die	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLL
Viterbi	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLL
Rolls	651166453132651245636664631636663162326455235266666625151631
Die	LLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLLFFFLLLLLLLLLLLLLLFFFFFFFFF
Viterbi	LLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFF
Rolls	222555441666566563564324364131513465146353411126414626253356
Die	FFFFFFFFLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
Viterbi	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
Rolls	366163666466232534413661661163252562462255265252266435353336
Die	LLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Viterbi	LLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Rolls	233121625364414432335163243633665562466662632666612355245242
Die	FFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
Viterbi	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF

Slide by Changui Yan

Rain/Umbrella Example