Artificial Intelligence

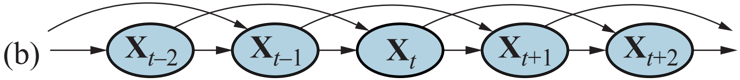
Probabilistic Temporal Reasoning

Christopher Simpkins

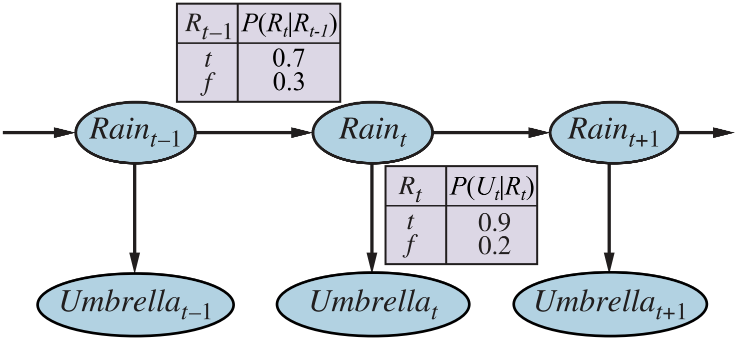
Kennesaw State University

Probabilistic Temporal Reasoning

First- and Second-Order Markov Processes



Umbrella World



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 in Temporal Models

- **Filtering**, a.k.a., **state estimation** is
- **Prediction**:
- **Smoothing**:
- **Most likely explanation**:

Learning Temporal Models

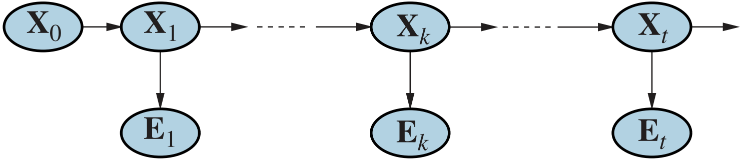
Unknown transition and sensor models can be learned from observations.

- As with static Bayesian networks, dynamic Bayes net learning can be done as a by-product of inference.
- Inference provides an estimate of transitions that actually occurred and the states that generated the sensor readings, and these estimates can be used to learn the models.
- Learning via iterative update algorithm, expectation–maximization or EM, or Bayesian updating of the model parameters given the evidence.

We'll return to these ideas in our lesson on statistical learning.

Filtering and Prediction

Smoothing

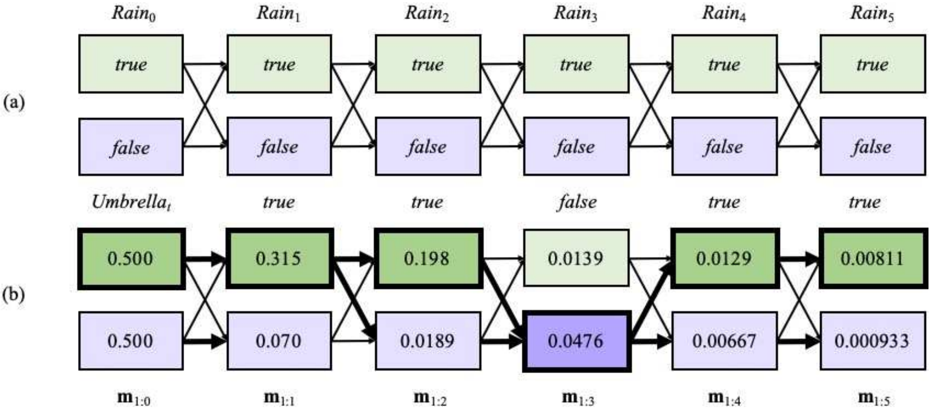

$\mathbf{X}_0$
$\mathbf{X}_1$
$\mathbf{X}_k$
$\mathbf{X}_t$
$\mathbf{E}_1$
$\mathbf{E}_k$
$\mathbf{E}_t$

Forward-Backward Smoothing Algorithm

```
function FORWARD-BACKWARD(ev, prior) returns a vector of probability distributions
    inputs: ev, a vector of evidence values for steps $1,\dots,t$
            prior, the prior distribution on the initial state, $\mathbf{P}(\mathbf{X}_0)$
    local variables: fv, a vector of forward messages for steps $0,\dots,t$
                     b, a representation of the backward message, initially all 1s
                     sv, a vector of smoothed estimates for steps $1,\dots,t$

    $\mathbf{fv}[0] \leftarrow prior$
    for $i = 1$ to $t$ do
        $\mathbf{fv}[i] \leftarrow$ FORWARD($\mathbf{fv}[i-1], \mathbf{ev}[i]$)
    for $i = t$ down to 1 do
        $\mathbf{sv}[i] \leftarrow$ NORMALIZE($\mathbf{fv}[i] \times \mathbf{b}$)
        $\mathbf{b} \leftarrow$ BACKWARD($\mathbf{b}, \mathbf{ev}[i]$)
    return sv
```

Finding the Most Likely Sequence

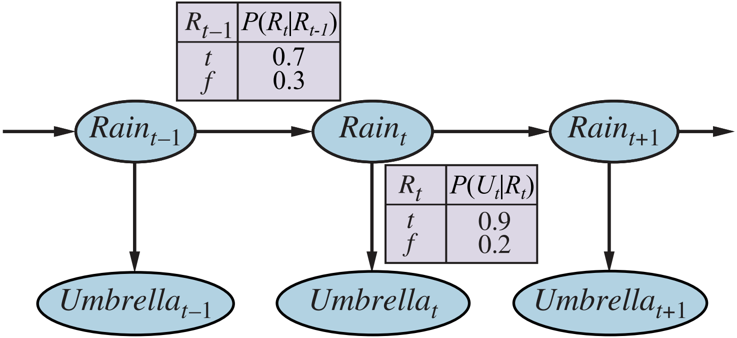


Hidden Markov Models (HMMs)

An HMM is a temporal probabilis- tic model in which the state of the process is described by a single, discrete random variable.

HMM Matrix Formulation

$$\boldsymbol{T}_{ij} = Pr(X_t = j \mid X_{t-1} = i)$$



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

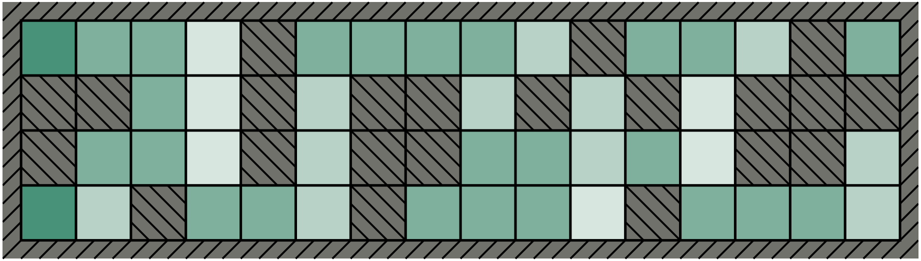
$$\boldsymbol{T}_{ij} = Pr(X_t \mid X_{t-1}) = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$$

Fixed Lag Smoothing Algorithm

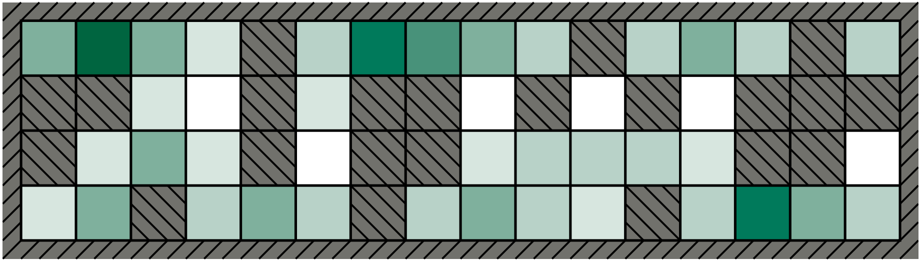
```
function FIXED-LAG-SMOOTHING($e_t, hmm, d$) returns a distribution over $\mathbf{X}_{t-d}$
  inputs: $e_t$, the current evidence for time step $t$
          $hmm$, a hidden Markov model with $S \times S$ transition matrix $\mathbf{T}$
          $d$, the length of the lag for smoothing
  persistent: $t$, the current time, initially 1
              $\mathbf{f}$, the forward message $\mathbf{P}(X_t \mid e_{1:t})$, initially $hmm$.PRIOR
              $\mathbf{B}$, the $d$-step backward transformation matrix, initially the identity matrix
              $e_{t-d:t}$, double-ended list of evidence from $t-d$ to $t$, initially empty
  local variables: $\mathbf{O}_{t-d}, \mathbf{O}_t$, diagonal matrices containing the sensor model information

  add $e_t$ to the end of $e_{t-d:t}$
  $\mathbf{O}_t \leftarrow$ diagonal matrix containing $\mathbf{P}(e_t \mid X_t)$
  if $t > d$ then
     $\mathbf{f} \leftarrow$ FORWARD($\mathbf{f}, e_{t-d}$)
     remove $e_{t-d-1}$ from the beginning of $e_{t-d:t}$
     $\mathbf{O}_{t-d} \leftarrow$ diagonal matrix containing $\mathbf{P}(e_{t-d} \mid X_{t-d})$
     $\mathbf{B} \leftarrow \mathbf{O}_{t-d}^{-1} \mathbf{T}^{-1} \mathbf{B} \mathbf{T} \mathbf{O}_t$
  else $\mathbf{B} \leftarrow \mathbf{B} \mathbf{T} \mathbf{O}_t$
  $t \leftarrow t + 1$
  if $t > d + 1$ then return NORMALIZE($\mathbf{f} \times \mathbf{B1}$) else return null
```

Localization with HMMs

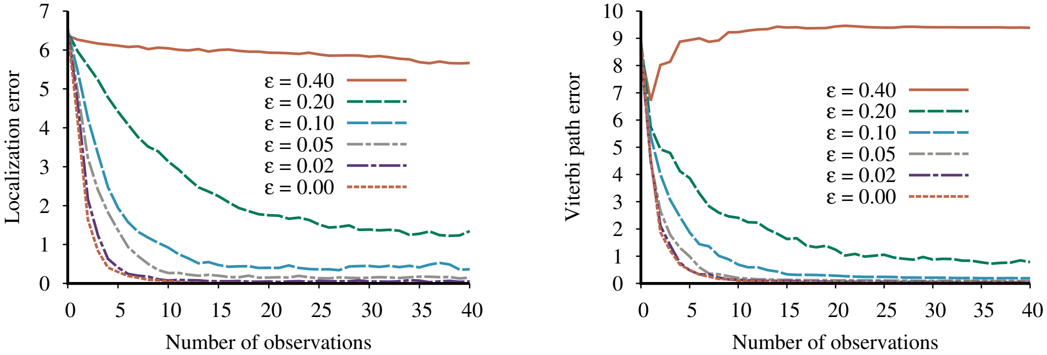


(a) Posterior distribution over robot location after $E_1 = 1011$



(b) Posterior distribution over robot location after $E_1 = 1011$, $E_2 = 1010$

HMM Performance



(a)

(b)