Non-Standard Databases and Data Mining

Dynamic Bayesian Networks

Prof. Dr. Ralf Möller
Universität zu Lübeck
Institut für Informationssystem

Time and Uncertainty

- The world changes, we need to track and predict it
- Examples: diabetes management, traffic monitoring
- Uncertainty is everywhere
- Need temporal probabilistic graphical models
- 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., $\text{MeasuredBloodSugar}_t$, PulseRate_t, FoodEaten_t
- Assumes discrete time steps

States and Observations

- Process of change viewed as series of snapshots, each describing the state of the world at a particular time
- 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 $\mathbf{e}_t$
- 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?

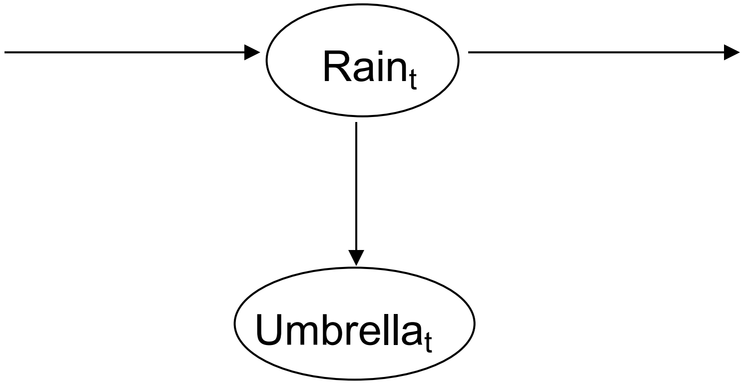
$$X_t = \{R_t\}$$

$$E_t = \{U_t\}$$

⇨ 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.

Example



DBN - Representation

- Problem: all previous random variables could have an influence on those of the the current timestamp

  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

Stationary Process/Markov Assumption

- Markov Assumption: X_t depends on some parent X_is
- First-order Markov process:

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

Transition Model

  - kth order: depends on previous k time steps
- Sensor Markov assumption:

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

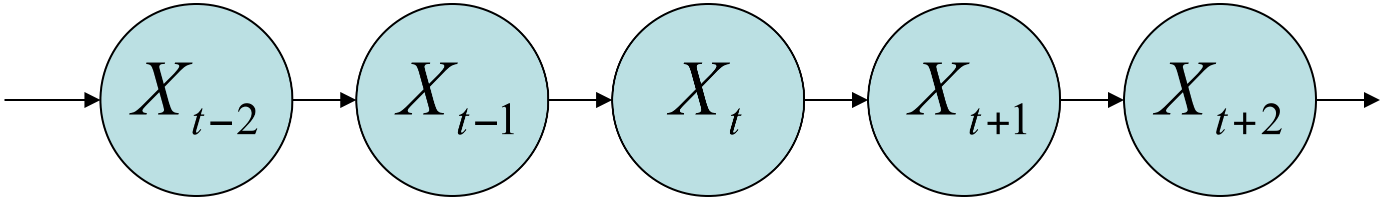
Sensor Model

- 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
  - Changes in the world state governed by laws not changing over time

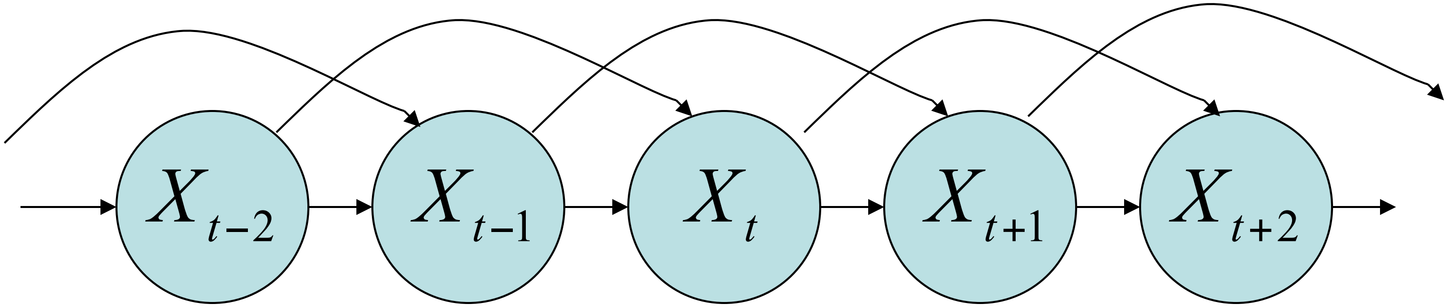
Dynamic Bayesian Networks

- 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.

Dynamic Bayesian Network



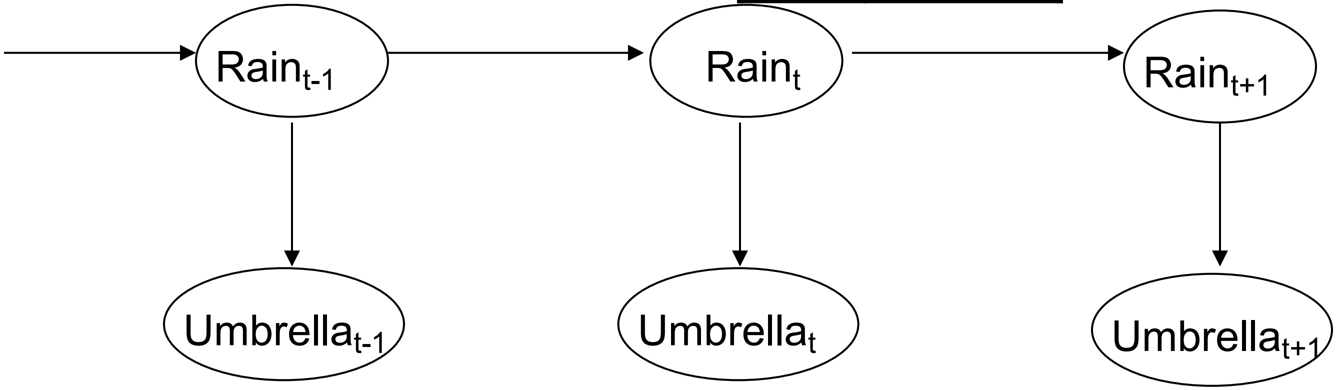
Bayesian network structure corresponding to a first-order of Markov process with state defined by the variables Xt.



A second order of Markov process

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

Complete Joint Distribution: Markov-1

- 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:

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

Inference Tasks

- 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 / most probable 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.

Bayes Rule

$$P(A \mid B) = P(A, B) \,/\, P(B)$$

$$P(A,B) = P(A \mid B)\, P(B) = P(B \mid A)\, P(A) = P(B, A)$$

Application of Bayes Rule

$$
\begin{aligned}
P(A \mid B, C) &= P(A, B, C) \,/\, P(B, C) \\
&= P(C, A, B) \,/\, P(B, C) \\
&= P(C \mid A, B)\, P(A, B) \,/\, P(B, C) \\
&= P(C \mid A, B)\, P(A \mid B)\, P(B) \,/\, (P(C \mid B)\, P(B)) \\
&= \alpha\, P(C \mid A, B)\, P(A \mid B)
\end{aligned}
$$

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

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.

Application of Bayes Rule

$$\begin{aligned}
P(A \mid B) &= \Sigma_c \, P(A, c \mid B) \\
&= \Sigma_c \, P(A, c, B) \,/\, P(B) \\
&= \Sigma_c \, P(A \mid c, B) \, P(c, B) \,/\, P(B) \\
&= \Sigma_c \, P(A \mid c, B) \, P(c \mid B) \, P(B) \,/\, P(B) \\
&= \Sigma_c \, P(A \mid c, B) \, P(c \mid B)
\end{aligned}$$

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

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:

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

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

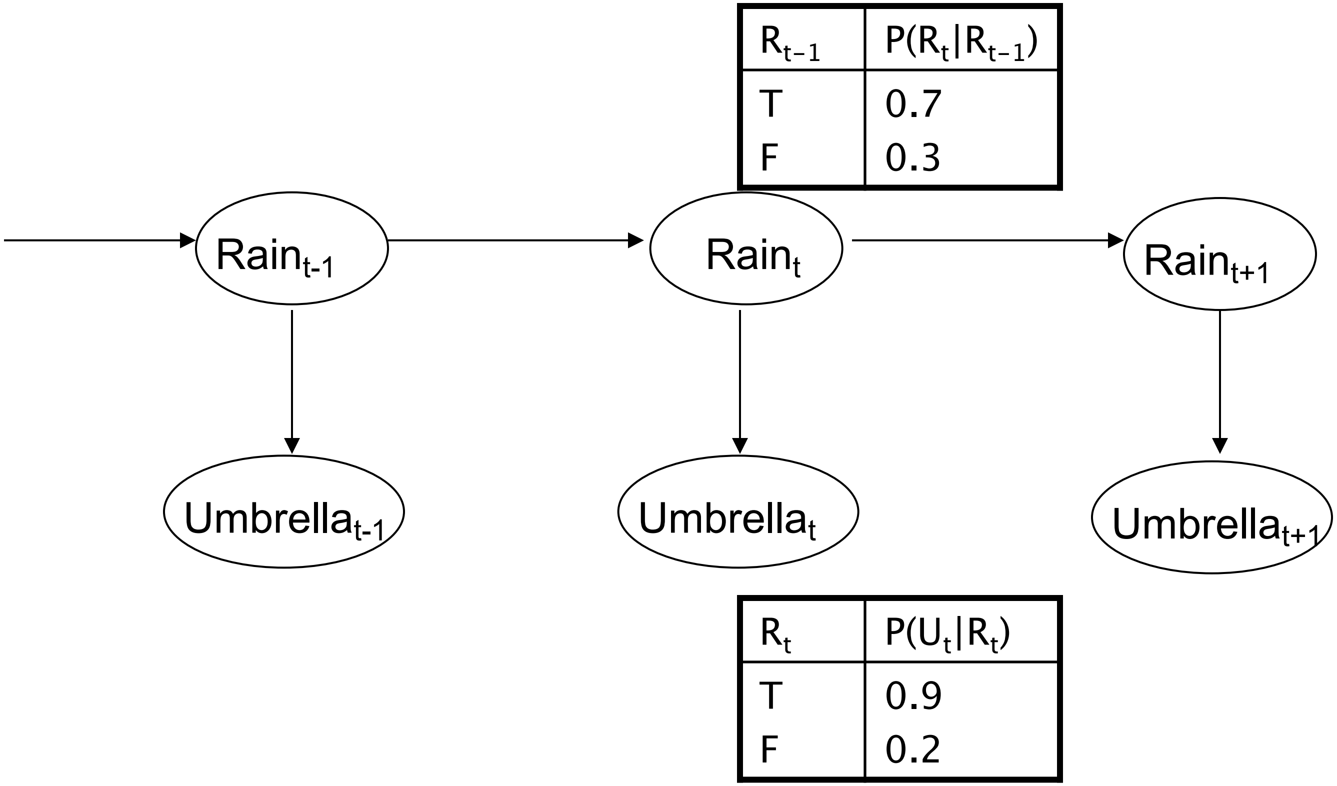
(using the Markov property)

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)

Example $P(Rain_0) = (0.5\ 0.5)^T$



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

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

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

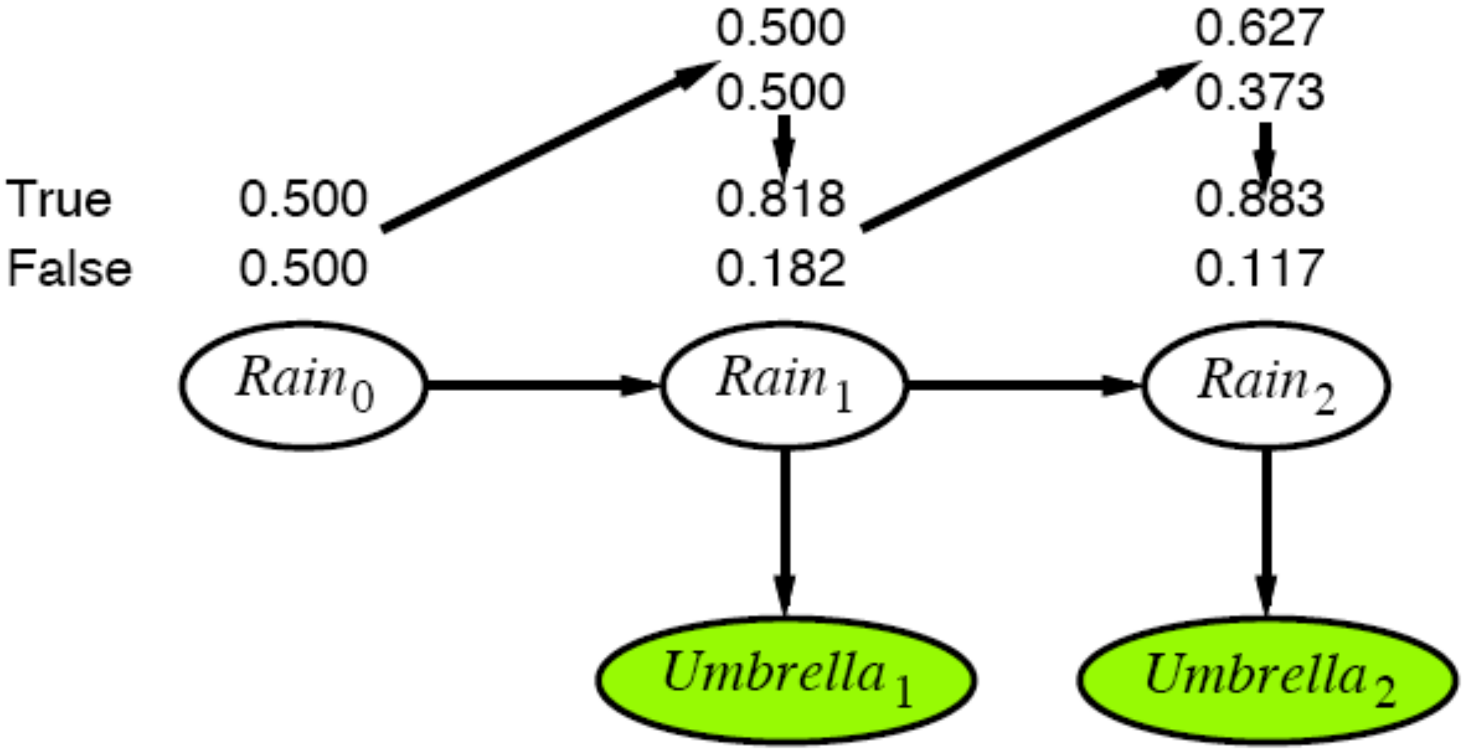
- On day 2, the umbrella appears, so U2=true. The prediction from t=1 to t=2 is

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

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 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 $0 \leq k < t$.

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 $O(t|\mathbf{f}|)$

Application of Bayes Rule

$$
\begin{aligned}
P(A \mid B, C) &= P(A, B, C) \,/\, P(B, C) \\
&= P(C, A, B) \,/\, P(B, C) \\
&= P(C \mid A, B)\, P(A, B) \,/\, P(B, C) \\
&= P(C \mid A, B)\, P(A \mid B)\, P(B) \,/\, (P(C \mid B)\, P(B)) \\
&= \alpha\, P(C \mid A, B)\, P(A \mid B)
\end{aligned}
$$

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t}) \;=\; \alpha \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{e}_{1:k})$$

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 $O(t|\mathbf{f}|)$

Application of Bayes Rule

$$
\begin{aligned}
P(A \mid B) &= \Sigma_c\, P(A, c \mid B) \\
&= \Sigma_c\, P(A, c, B) / P(B) \\
&= \Sigma_c\, P(A \mid c, B)\, P(c, B) / P(B) \\
&= \Sigma_c\, P(A \mid c, B)\, P(c \mid B)\, P(B) / P(B) \\
&= \Sigma_c\, P(A \mid c, B)\, P(c \mid B)
\end{aligned}
$$

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

Smoothing

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

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

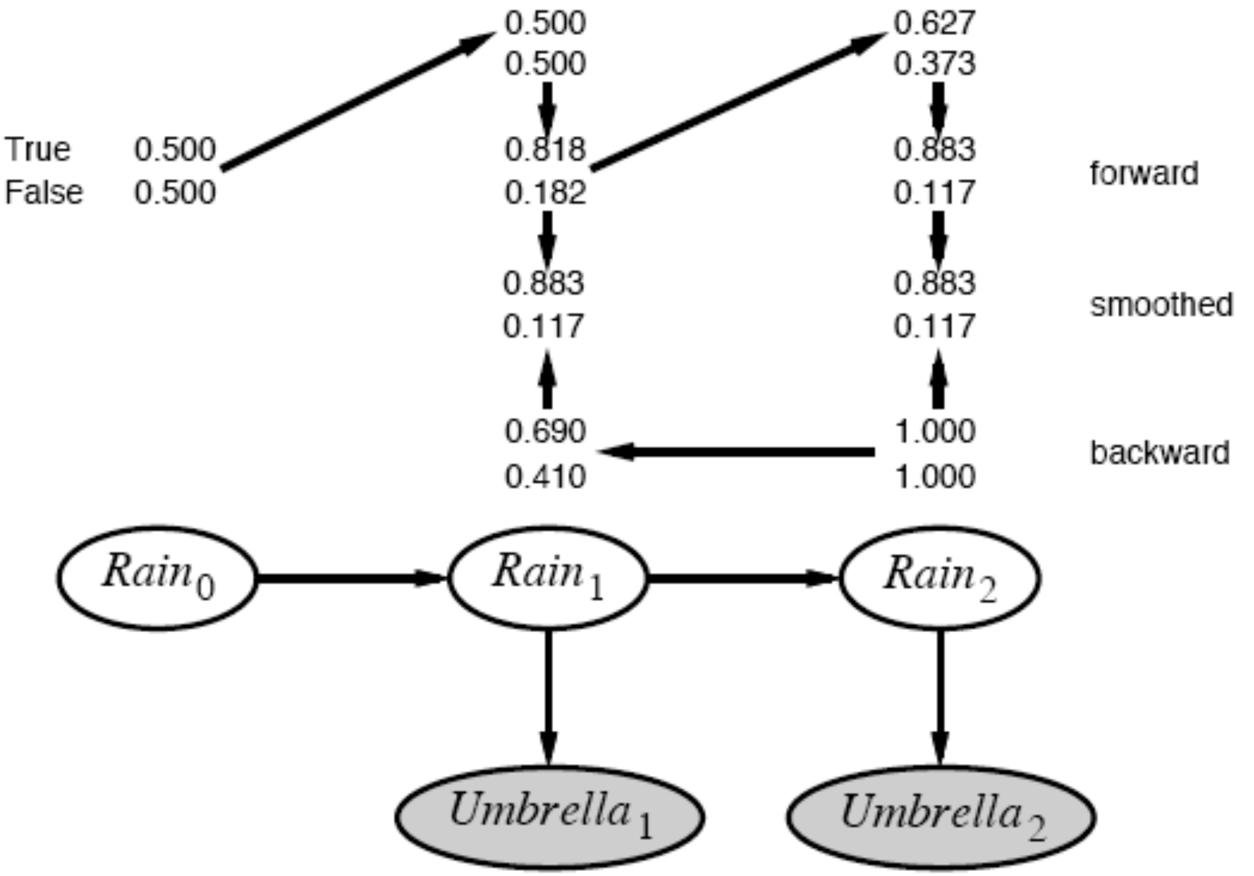
Backward message computed by a backwards recursion:

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

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

- 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:

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

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

(using the Markov property)

DBN – Basic Inference

- Most likely explanation:

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

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

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

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

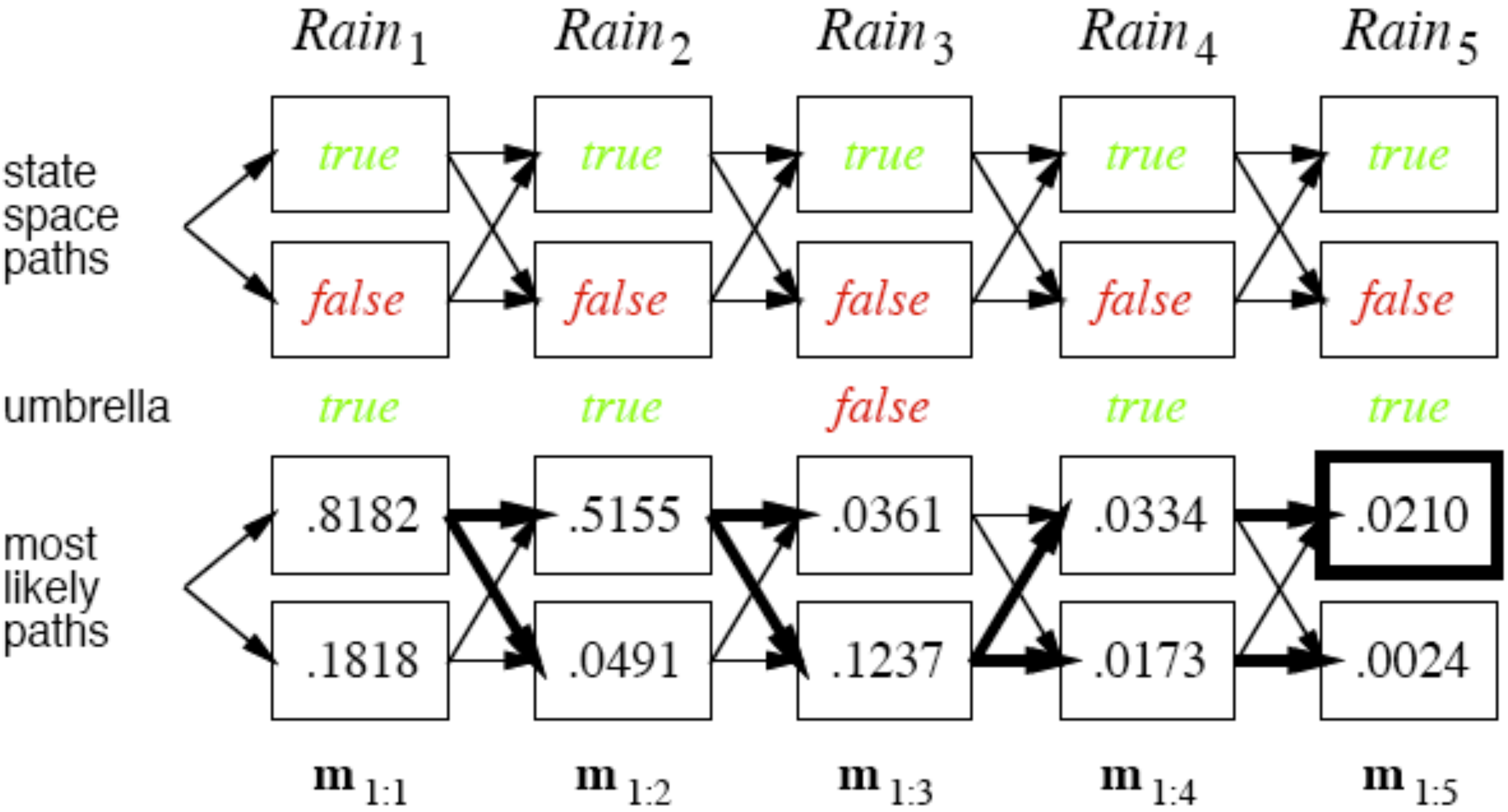
Identical to filtering, except $\mathbf{f}_{1:t}$ replaced by

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

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:

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

Rain/Umbrella Example



Hidden Markov Model (HMM)

Consider special case of a dynamic Bayesian Network:

- Use vector of independent state variables $\mathbf{X}_t$
- Use vector of independent evidence variables $\mathbf{E}_t$
- This was already used in the rain-umbrella example
- For high-dimensional vectors the transition and sensor models become quite complex: $O(d^2)$ space

NB:

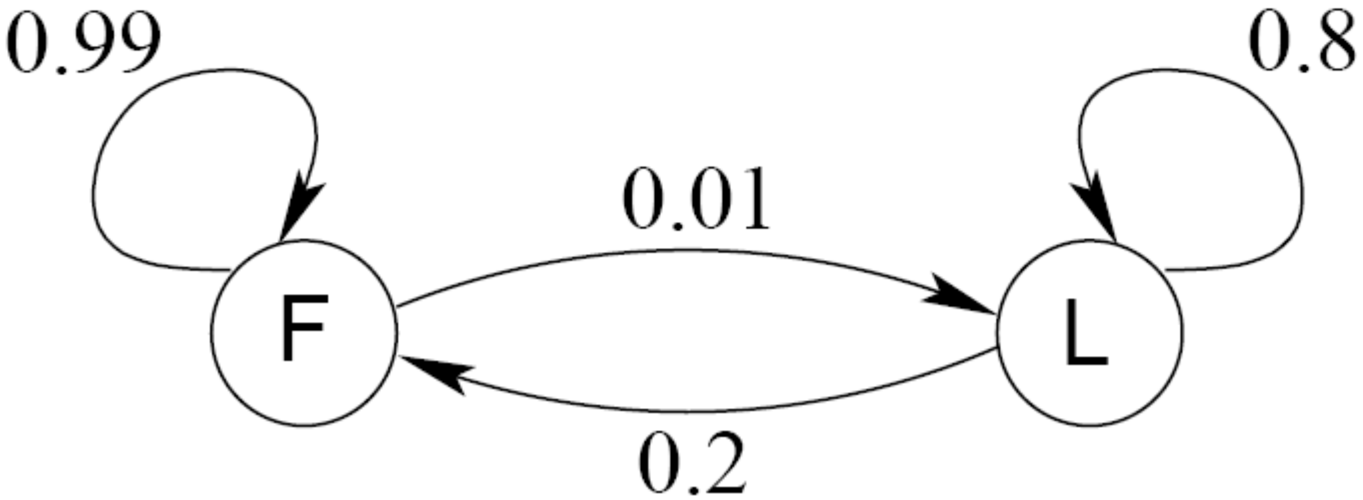
- In a general dynamic Bayesian network,
  state variables are not necessarily independent
- Even evidence variables might be dependent on one another (naïve Bayes does not work)

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 modeled by a Markov process

Slide by Changui Yan

Transition model for the 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 model
  - 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*:

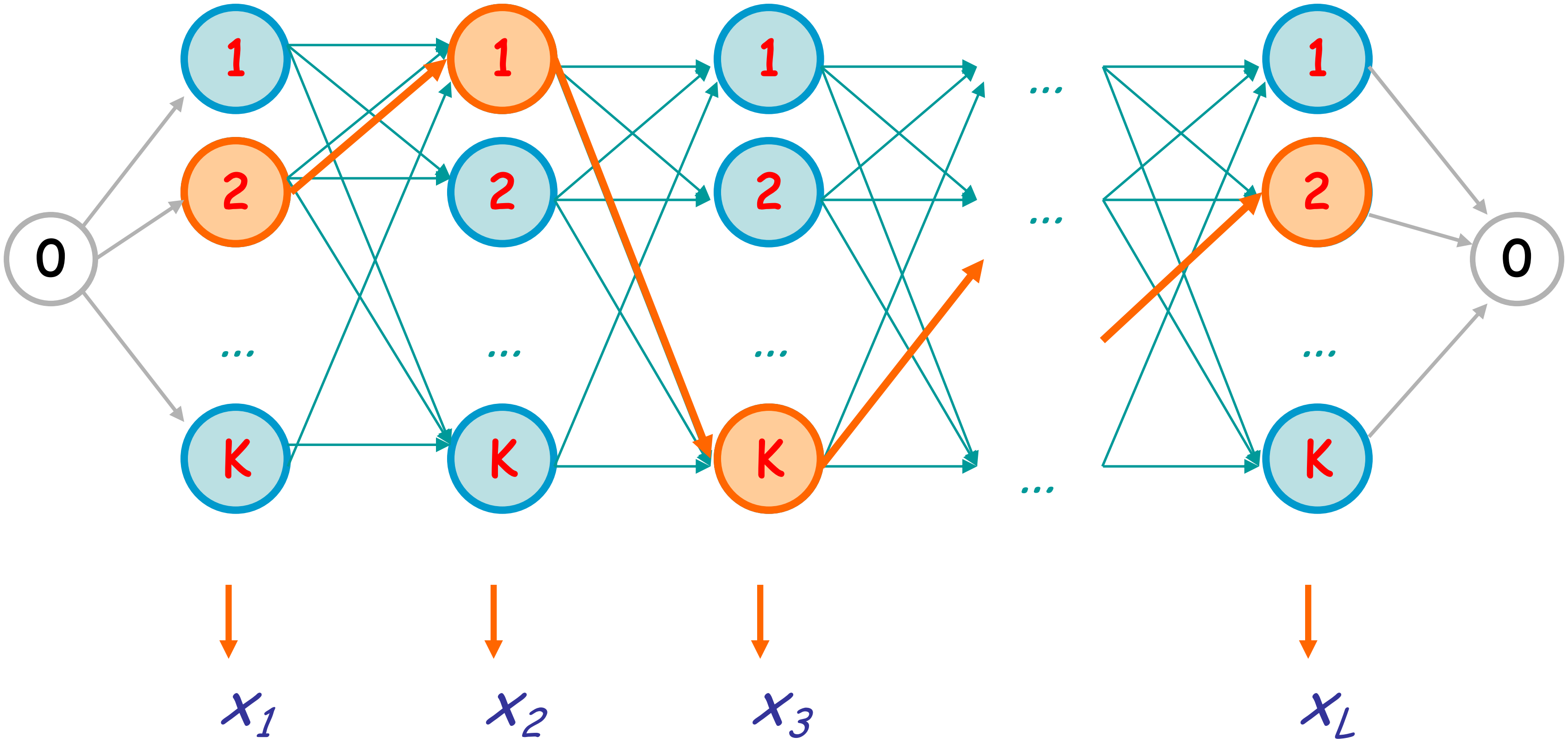
$$a_{kr} = \Pr(\pi_i = r \mid \pi_{i-1} = k)$$

- $e_k(b)$ 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

$$v_k(i) = \text{Prob. of path } \langle \pi_1, \cdots, \pi_i \rangle \text{ most likely}$$

$$\text{to emit } \langle x_1, \ldots, x_i \rangle \text{ such that } \pi_i = k$$

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 ($i = 1, \ldots, L$): For each state k

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

- Termination:

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

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

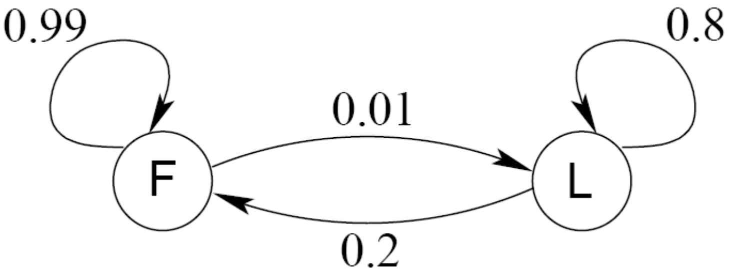
Slide by Changui Yan

Viterbi: Example

x

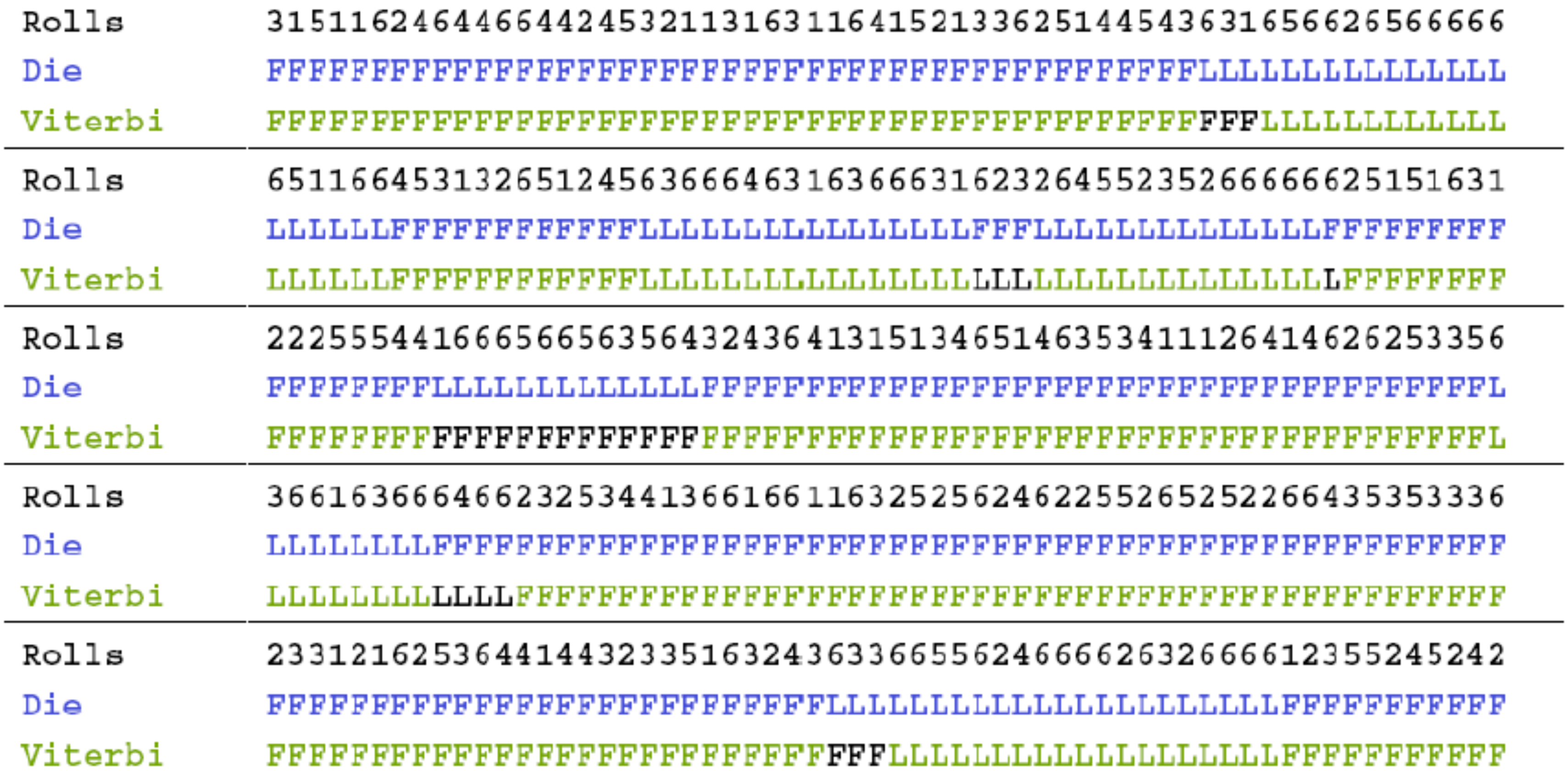
		ε	6	2	6
	B	**1**	**0**	**0**	**0**
π	F	**0**	$(1/6)\times(1/2)$ = **1/12**	$(1/6)\times\max\{(1/12)\times0.99,$ $(1/4)\times0.2\}$ = **0.01375**	$(1/6)\times\max\{0.01375\times0.99,$ $0.02\times0.2\}$ = **0.00226875**
	L	**0**	$(1/2)\times(1/2)$ = **1/4**	$(1/10)\times\max\{(1/12)\times0.01,$ $(1/4)\times0.8\}$ = **0.02**	$(1/2)\times\max\{0.01375\times0.01,$ $0.02\times0.8\}$ = **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	LLLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLFFFLLLLLLLLLLLLLLFFFFFFFFF
Viterbi	LLLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFF
Rolls	222555441666566563564324364131513465146353411126414626253356
Die	FFFFFFFFLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
Viterbi	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
Rolls	366163666466232534413661661163252562462255265252266435353336
Die	LLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Viterbi	LLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Rolls	233121625364414432335163243633665562466662632666612355245242
Die	FFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
Viterbi	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF

Slide by Changui Yan