Markov Decision Processes

Philipp Koehn

4 April 2024

Outline

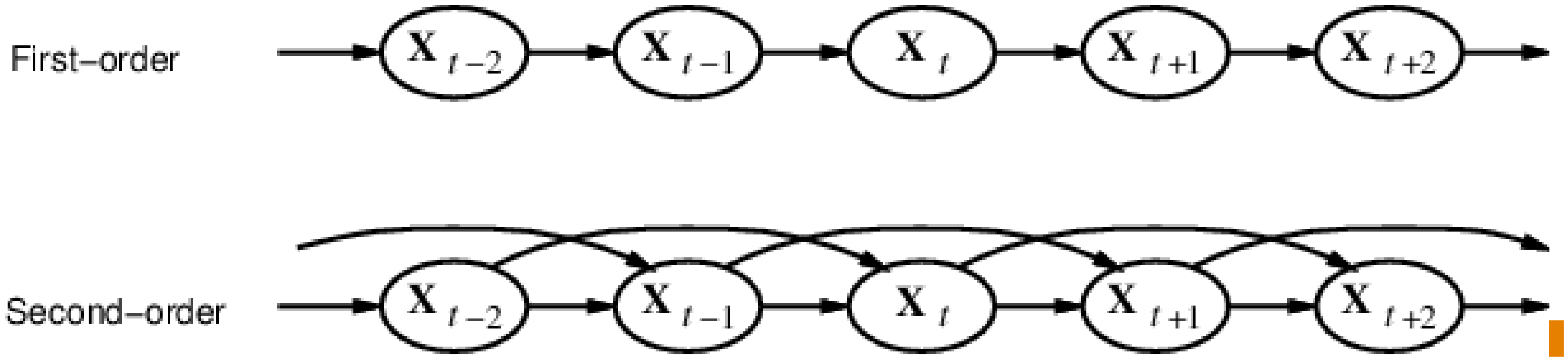
- Hidden Markov models
- Inference: filtering, smoothing, best sequence
- Dynamic Bayesian networks
- Speech recognition

Time and Uncertainty

- The world changes; we need to track and predict it
- Diabetes management vs vehicle diagnosis
- Basic idea: sequence of state and evidence variables
- $\mathbf{X}_t$ = set of unobservable state variables at time t
  e.g., $BloodSugar_t$, $StomachContents_t$, etc.
- $\mathbf{E}_t$ = set of observable evidence variables at time t
  e.g., $MeasuredBloodSugar_t$, $PulseRate_t$, $FoodEaten_t$
- This assumes **discrete time**; step size depends on problem
- Notation: $\mathbf{X}_{a:b} = \mathbf{X}_a, \mathbf{X}_{a+1}, \ldots, \mathbf{X}_{b-1}, \mathbf{X}_b$

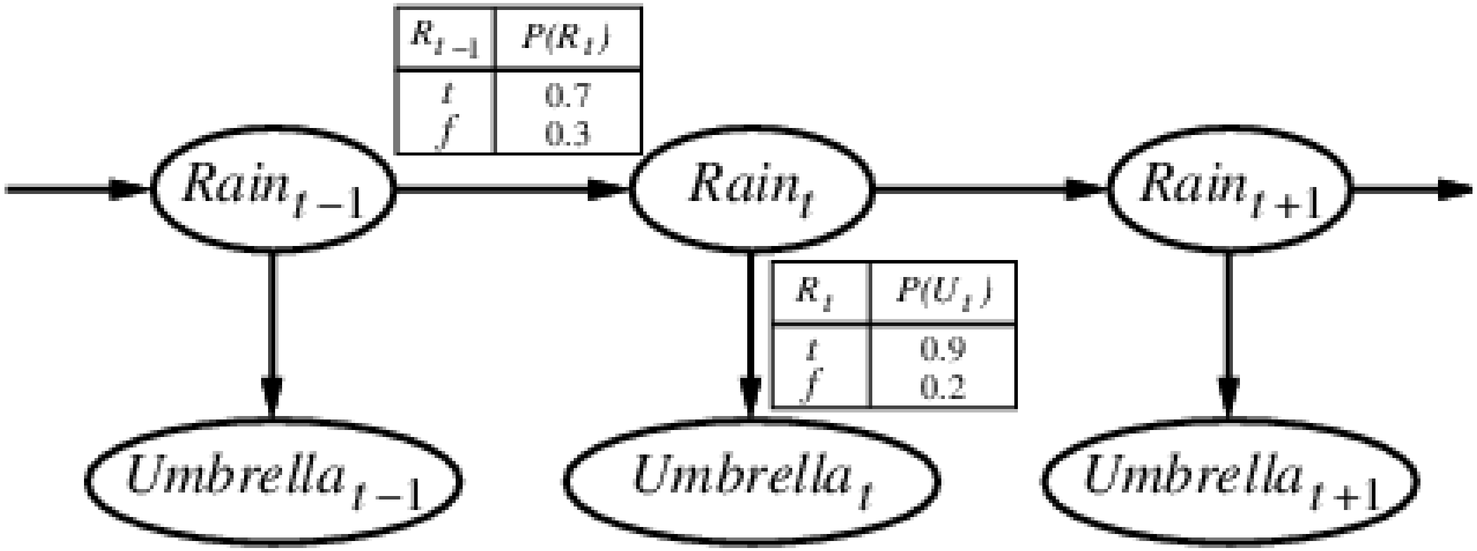
Markov Processes (Markov Chains)

- Construct a Bayes net from these variables: parents?
- Markov assumption: $\mathbf{X}_t$ depends on **bounded** subset of $\mathbf{X}_{0:t-1}$
- First-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) \simeq \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
  Second-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) \simeq \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2}, \mathbf{X}_{t-1})$



- Sensor Markov assumption: $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t}, \mathbf{E}_{0:t-1}) \simeq \mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$
- Stationary process: transition model $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$ and
  sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$ fixed for all t

Example



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

R_t	$P(U_t)$
t	0.9
f	0.2

- First-order Markov assumption not exactly true in real world!

- Possible fixes:
  1. **Increase order** of Markov process
  2. **Augment state**, e.g., add $Temp_t$, $Pressure_t$

inference

Inference Tasks

- Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$

  belief state—input to the decision process of a rational agent

- Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \leq k < t$

  better estimate of past states, essential for learning

- Most likely explanation: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$

  speech recognition, decoding with a noisy channel

Filtering

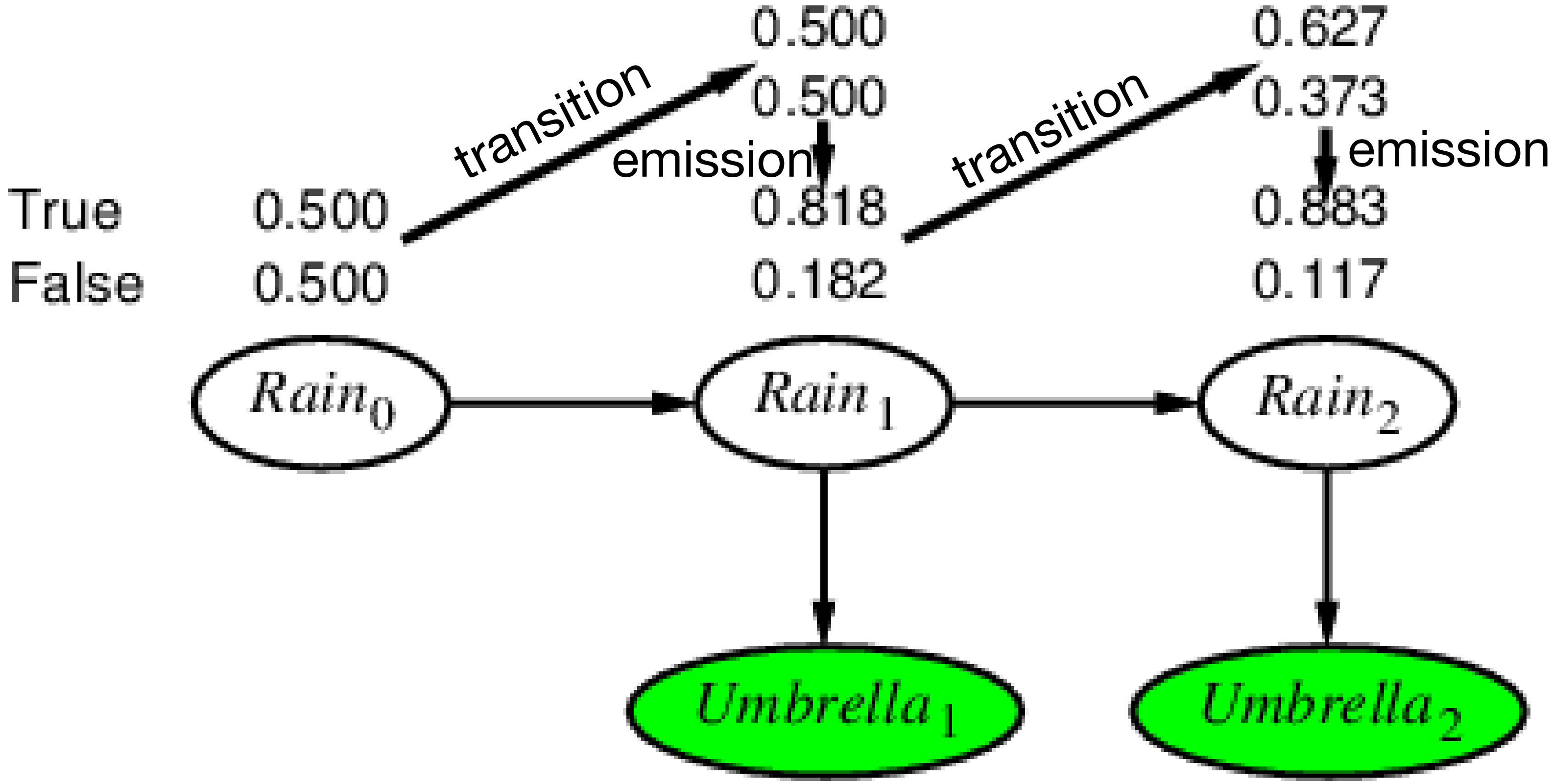
- Aim: devise a **recursive** state estimation algorithm

$$
\begin{aligned}
\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t},\mathbf{e}_{t+1}) & \\
= \quad & \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1},\mathbf{e}_{1:t})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) && \textit{(Bayes rule)} \\
\simeq \quad & \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) && \textit{(Sensor Markov assumption)} \\
= \quad & \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\sum_{\mathbf{x}_t}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t,\mathbf{e}_{1:t})P(\mathbf{x}_t|\mathbf{e}_{1:t}) && \textit{(multiplying out)} \\
\simeq \quad & \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\sum_{\mathbf{x}_t}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{e}_{1:t}) && \textit{(first order Markov model)}
\end{aligned}
$$

- Summary: $\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) \simeq \alpha\,\underbrace{\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})}_{\textbf{emission}}\sum_{\mathbf{x}_t}\underbrace{\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)}_{\textbf{transition}}\,\underbrace{P(\mathbf{x}_t|\mathbf{e}_{1:t})}_{\textbf{recursive call}}$

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

Filtering Example

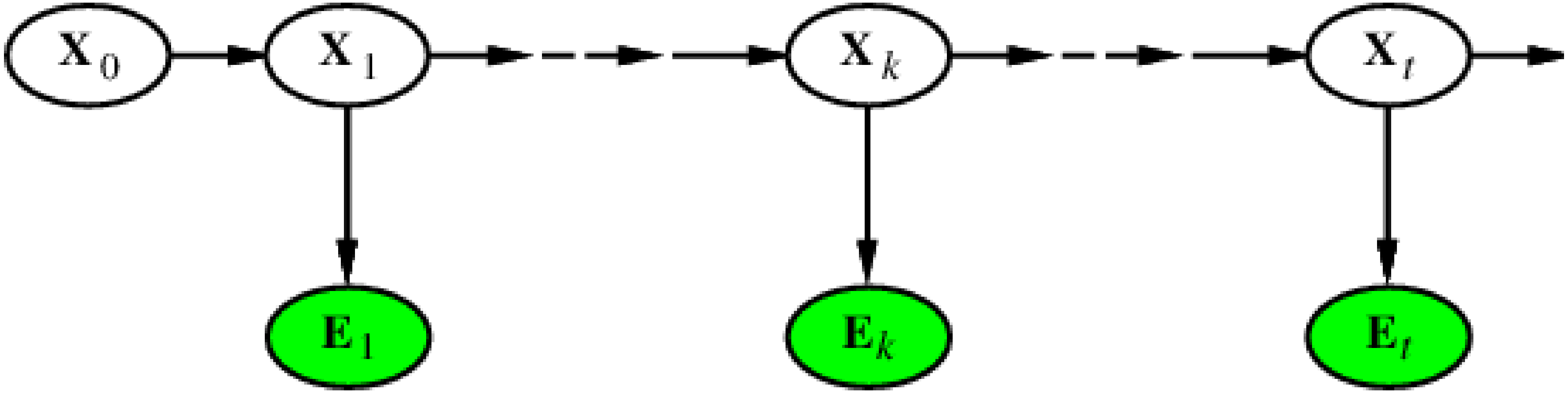


$$P(Rain_t|Rain_{t-1}) = 0.7 \qquad P(Umbrella_t|Rain_t) = 0.9$$

$$P(Rain_t|\overline{Rain}_{t-1}) = 0.3 \qquad P(Umbrella_t|\overline{Rain}_t) = 0.2$$

$$\alpha < 0.9 \times (0.7 \times 0.5 + 0.3 \times 0.5), 0.2 \times (0.7 \times 0.5 + 0.3 \times 0.5) >=< 0.818, 0.182 >$$

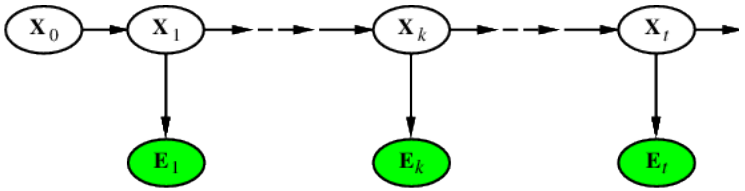
Smoothing



- If full sequence is known
  ⇒ what is the state probability $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ including future evidence?

- Smoothing: sum over all paths

Smoothing



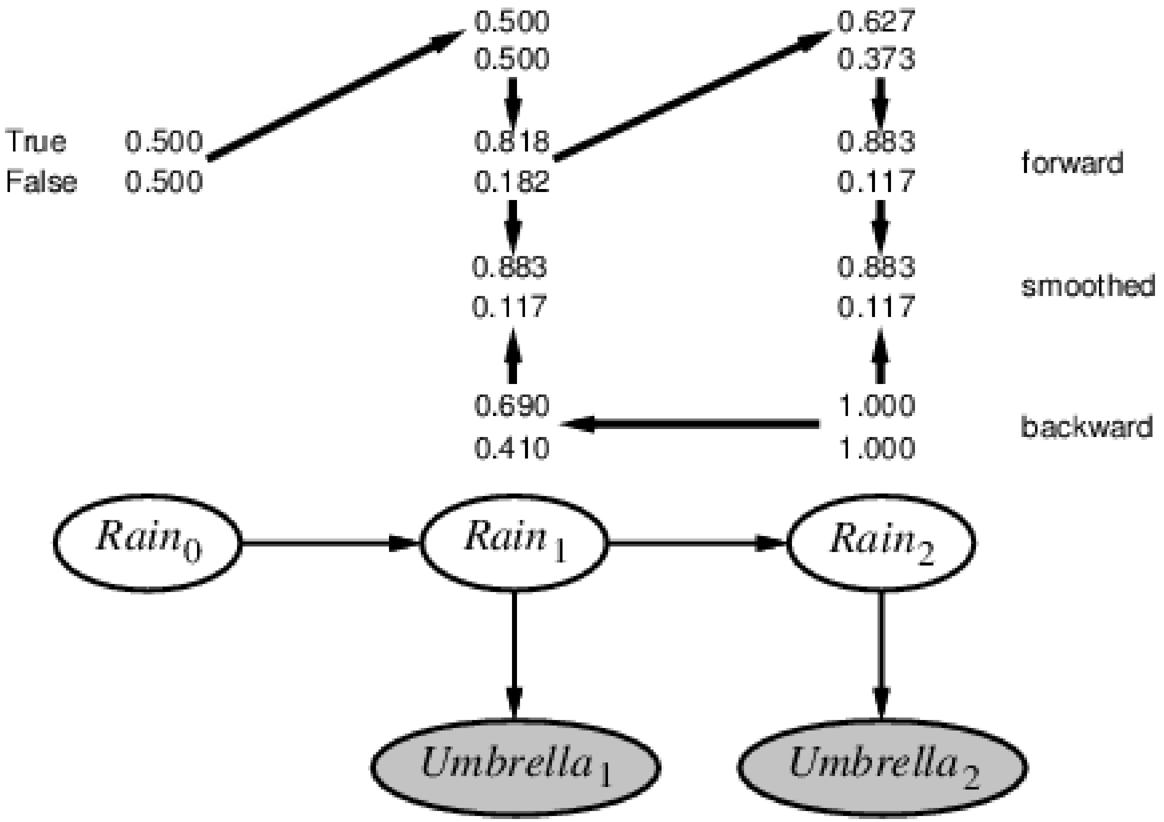
- Divide evidence $\mathbf{e}_{1:t}$ into $\mathbf{e}_{1:k}$, $\mathbf{e}_{k+1:t}$:

$$\begin{aligned}
\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) &= \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k},\mathbf{e}_{k+1:t}) \\
&= \alpha\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k,\mathbf{e}_{1:k}) \\
&\simeq \alpha\,\underbrace{\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})}_{\text{forward}}\,\underbrace{\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k)}_{\text{backward}}
\end{aligned}$$

- Backward message computed by a backwards recursion

$$\begin{aligned}
\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) &= \sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k,\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\
&\simeq \sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\
&= \sum_{\mathbf{x}_{k+1}} \underbrace{\mathbf{P}(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})}_{\text{emission}}\,\underbrace{\mathbf{P}(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})}_{\text{recursion}}\,\underbrace{\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)}_{\text{transition}}
\end{aligned}$$

Smoothing Example



Forward–backward algorithm: cache forward messages along the way

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

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

- 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.

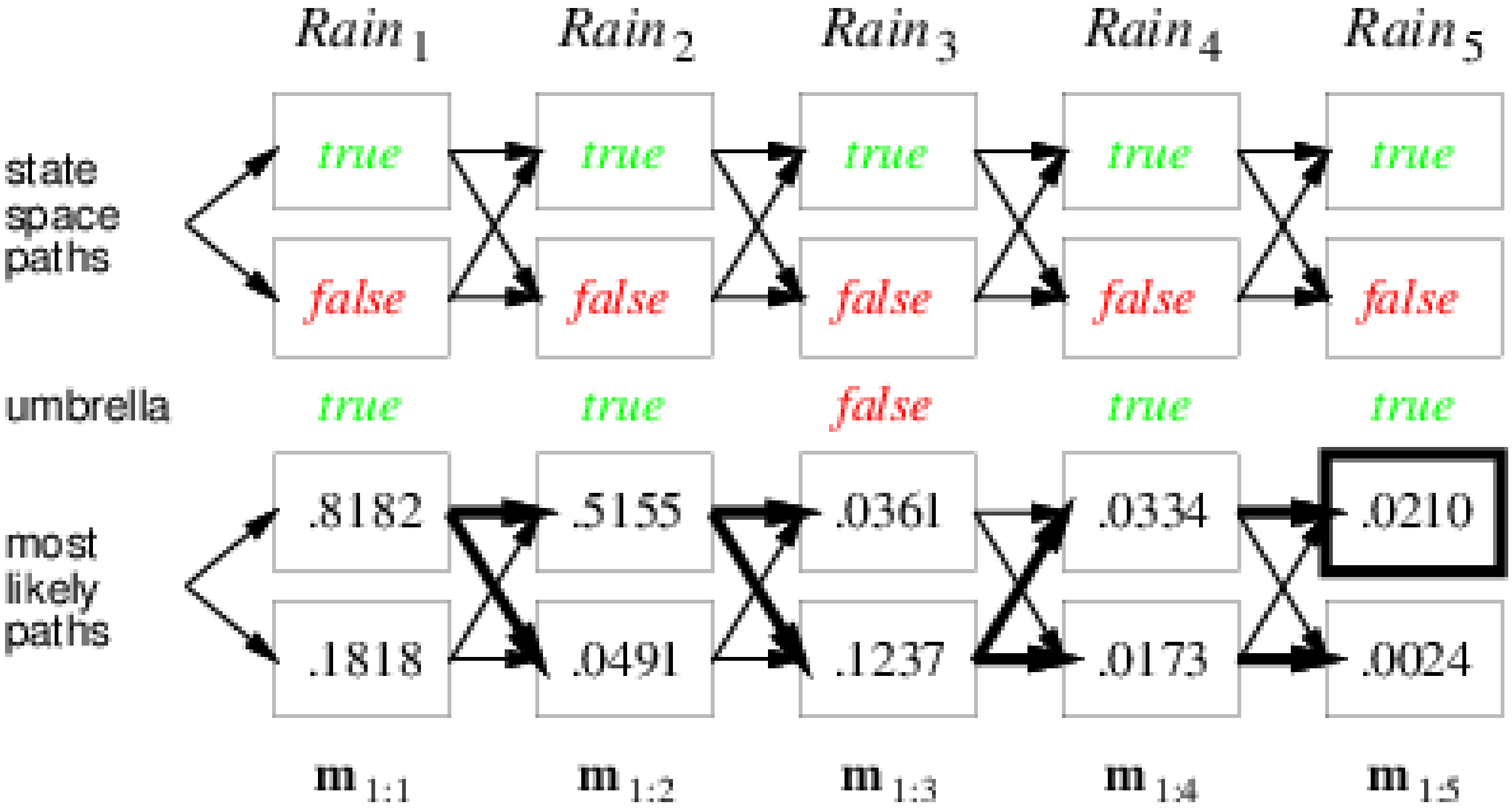
- Update has sum replaced by max, giving the Viterbi algorithm:

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

  Also requires back-pointers for backward pass to retrieve best sequence

$$\mathbf{b}_{\mathbf{X}_{t+1}, t+1} = \text{argmax}_{\mathbf{x}_t} \left(\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) \mathbf{m}_{1:t} \right)$$

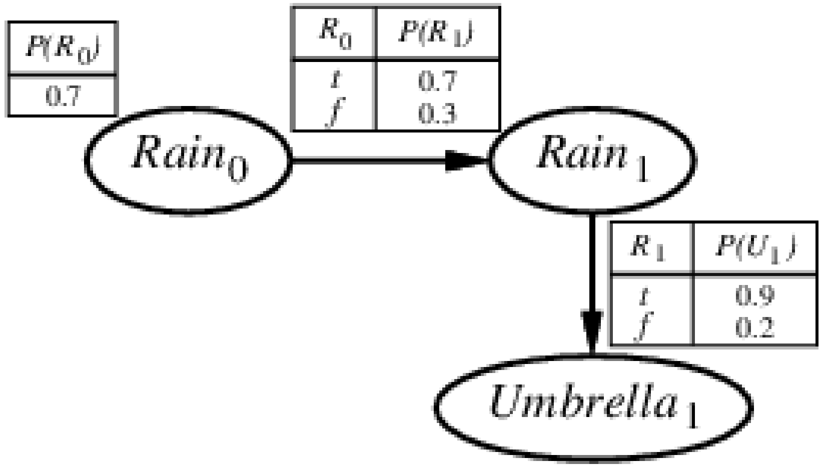
Viterbi Example



dynamic baysian networks

Dynamic Bayesian Networks

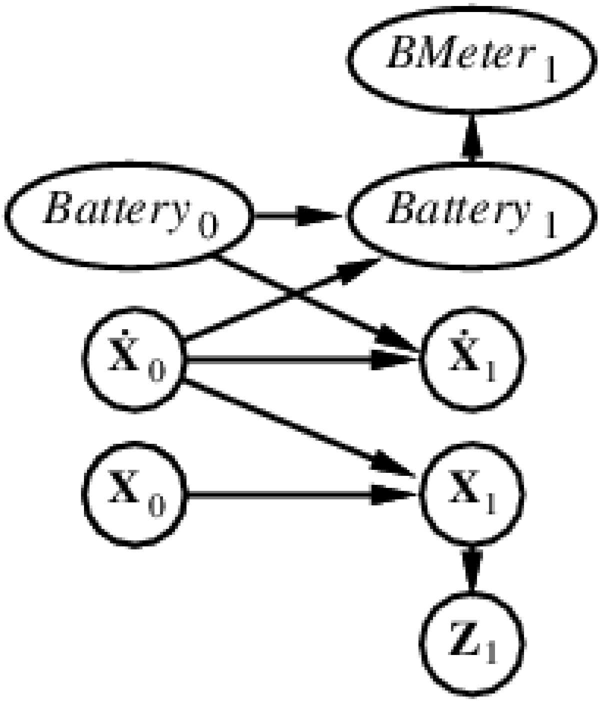
- $\mathbf{X}_t$, $\mathbf{E}_t$ contain arbitrarily many variables in a sequentialized Bayes net



$P(R_0)$
0.7

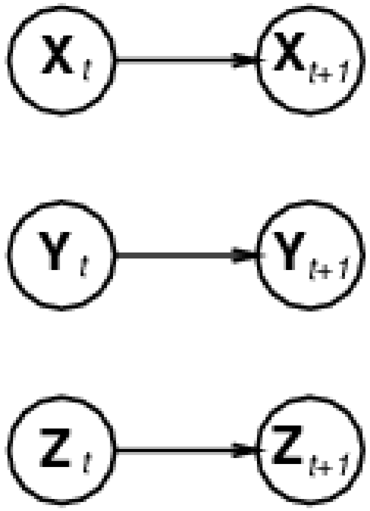
R_0	$P(R_1)$
t	0.7
f	0.3

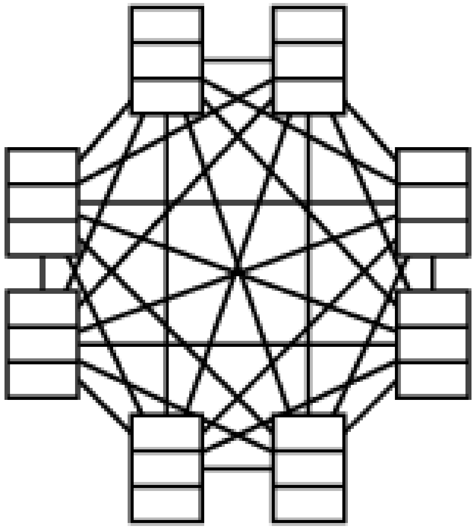
R_1	$P(U_1)$
t	0.9
f	0.2



DBNs vs. HMMs

- Every HMM is a single-variable DBN; every discrete DBN is an HMM





- Sparse dependencies ⇒ exponentially fewer parameters;
  e.g., 20 state variables, three parents each
  HMM has $2^{20} \times 2^{20} \approx 10^{12}$, DBN has $20 \times 2^3 = 160$ parameters

speech recognition

Speech as Probabilistic Inference

It's not easy to wreck a nice beach

- Speech signals are noisy, variable, ambiguous
- What is the **most likely** word sequence, given the speech signal?
  I.e., choose $Words$ to maximize $P(Words|signal)$
- Use Bayes' rule:
  $$P(Words|signal) = \alpha P(signal|Words)P(Words)$$
  i.e., decomposes into acoustic model + language model
- $Words$ are the hidden state sequence, $signal$ is the observation sequence

Phones

- All human speech is composed from 40-50 phones, determined by the configuration of articulators (lips, teeth, tongue, vocal cords, air flow)

- Form an intermediate level of hidden states between words and signal
  ⇒ acoustic model = pronunciation model + phone model

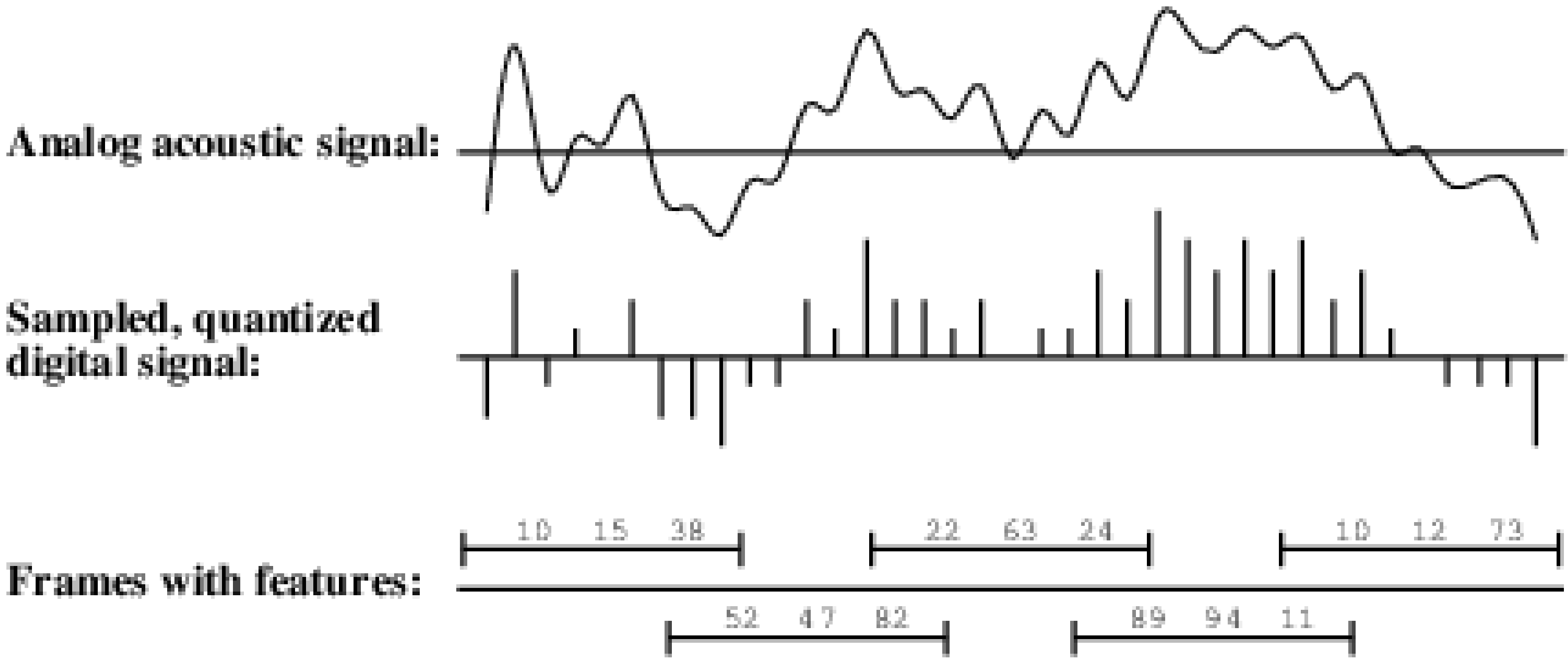
- ARPAbet designed for American English

[iy]	b**ea**t	[b]	**b**et	[p]	**p**et
[ih]	b**i**t	[ch]	**Ch**et	[r]	**r**at
[ey]	b**e**t	[d]	**d**ebt	[s]	**s**et
[ao]	b**ough**t	[hh]	**h**at	[th]	**th**ick
[ow]	b**oa**t	[hv]	**h**igh	[dh]	**th**at
[er]	B**er**t	[l]	**l**et	[w]	**w**et
[ix]	ros**e**s	[ng]	si**ng**	[en]	butt**on**
⋮	⋮	⋮	⋮	⋮	⋮

e.g., “ceiling” is [s iy l ih ng] / [s iy l ix ng] / [s iy l en]

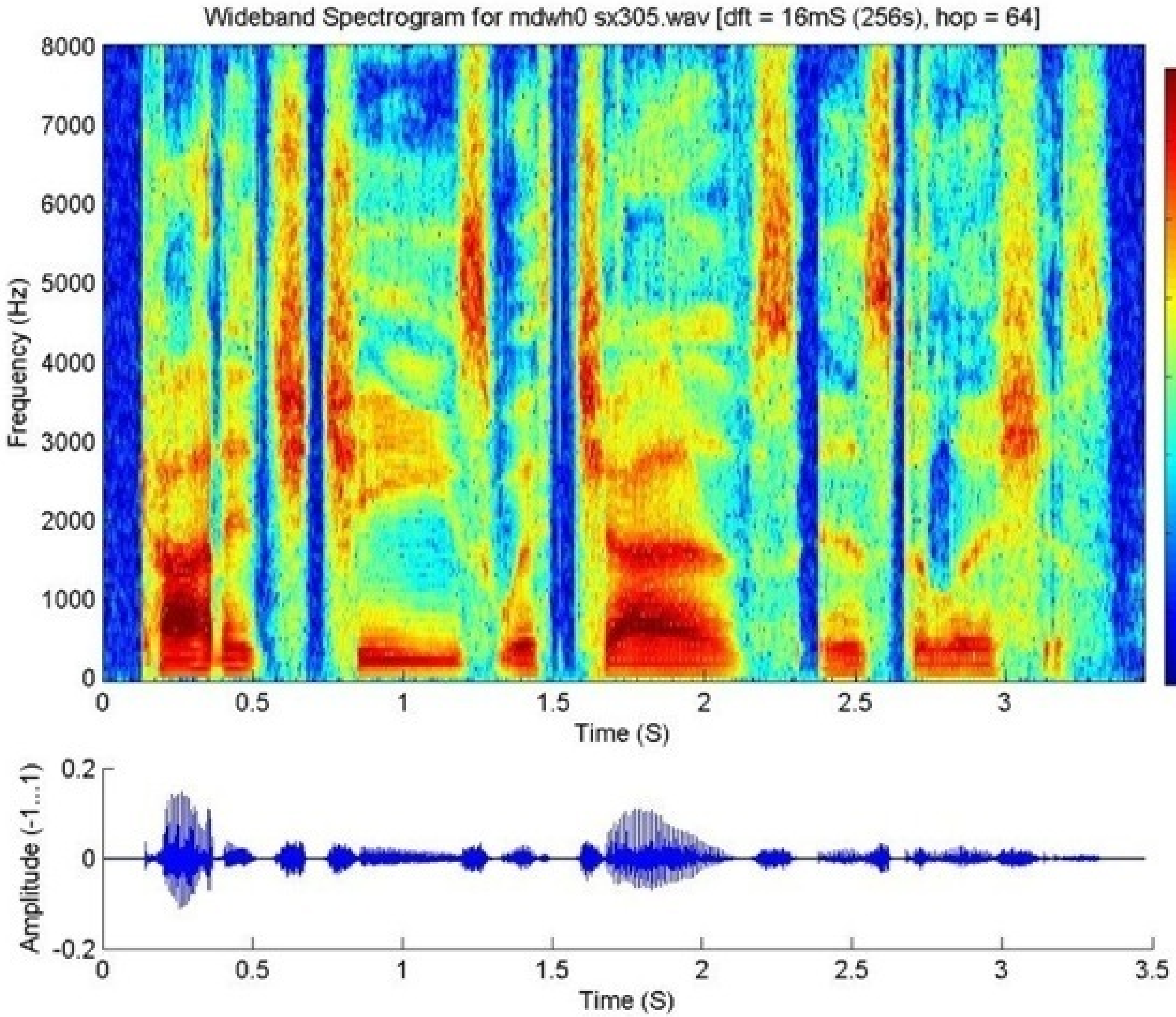
Speech Sounds

- Raw signal is the microphone displacement as a function of time; processed into overlapping 30ms frames, each described by features



- Frame features are typically formants—peaks in the power spectrum

Speech Spectrogram



Phone Models

- Frame features in $P(features|phone)$ summarized by
  – an integer in $[0 \ldots 255]$ (using vector quantization); or
  – the parameters of a mixture of Gaussians

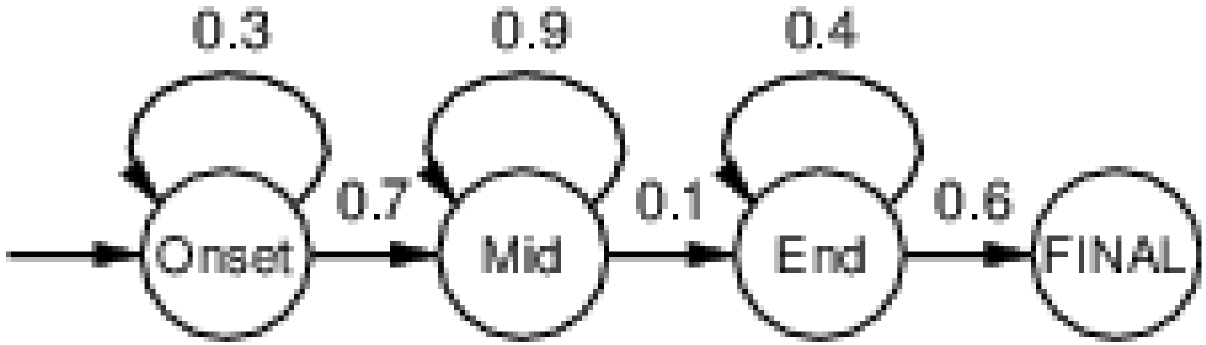
- Three-state phones: each phone has three phases (Onset, Mid, End)
  E.g., [t] has silent Onset, explosive Mid, hissing End
  $\Rightarrow P(features|phone, phase)$

- Triphone context: each phone becomes n^2 distinct phones, depending on the phones to its left and right
  E.g., [t] in "star" is written [t(s,aa)] (different from "tar"!)

- Triphones useful for handling coarticulation effects: the articulators have inertia and cannot switch instantaneously between positions
  E.g., [t] in "eighth" has tongue against front teeth

Phone Model Example

Phone HMM for [m]:



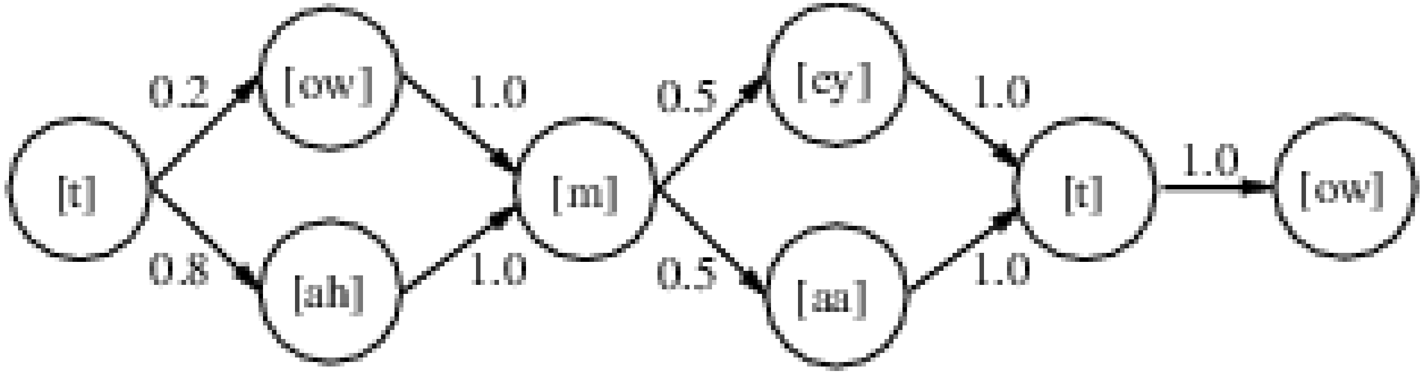
Output probabilities for the phone HMM:

Onset:	Mid:	End:
C1: 0.5	C3: 0.2	C4: 0.1
C2: 0.2	C4: 0.7	C6: 0.5
C3: 0.3	C5: 0.1	C7: 0.4

Word Pronunciation Models

- Each word is described as a distribution over phone sequences

- Distribution represented as an HMM transition model



$$P([towmeytow]|\text{“tomato”}) = P([towmaatow]|\text{“tomato”}) = 0.1$$
$$P([tahmeytow]|\text{“tomato”}) = P([tahmaatow]|\text{“tomato”}) = 0.4$$

- Structure is created manually, transition probabilities learned from data

Recognition of Isolated Words

- Phone models + word models fix likelihood $P(e_{1:t}|word)$ for isolated word

$$P(word|e_{1:t}) = \alpha P(e_{1:t}|word)P(word)$$

- Prior probability $P(word)$ obtained simply by counting word frequencies

  $P(e_{1:t}|word)$ can be computed recursively: define

$$\text{🔔}_{1:t} = \mathbf{P}(\mathbf{X}_t, \mathbf{e}_{1:t})$$

  and use the recursive update

$$\text{🔔}_{1:t+1} = \textsc{Forward}(\ell_{1:t}, \mathbf{e}_{t+1})$$

  and then $P(e_{1:t}|word) = \sum_{\mathbf{x}_t} \text{🔔}_{1:t}(\mathbf{x}_t)$

Continuous Speech

- Not just a sequence of isolated-word recognition problems!
  - adjacent words highly correlated
  - sequence of most likely words ≠ most likely sequence of words
  - segmentation: there are few gaps in speech
  - cross-word coarticulation—e.g., “next thing”

- Complications
  - mismatch between speaker in training and test
  - noise
  - crosstalk
  - bad microphone position

Language Model

- Prior probability of a word sequence is given by chain rule:

$$P(w_1 \cdots w_n) = \prod_{i=1}^{n} P(w_i | w_1 \cdots w_{i-1})$$

- Bigram model:

$$P(w_i | w_1 \cdots w_{i-1}) \approx P(w_i | w_{i-1})$$

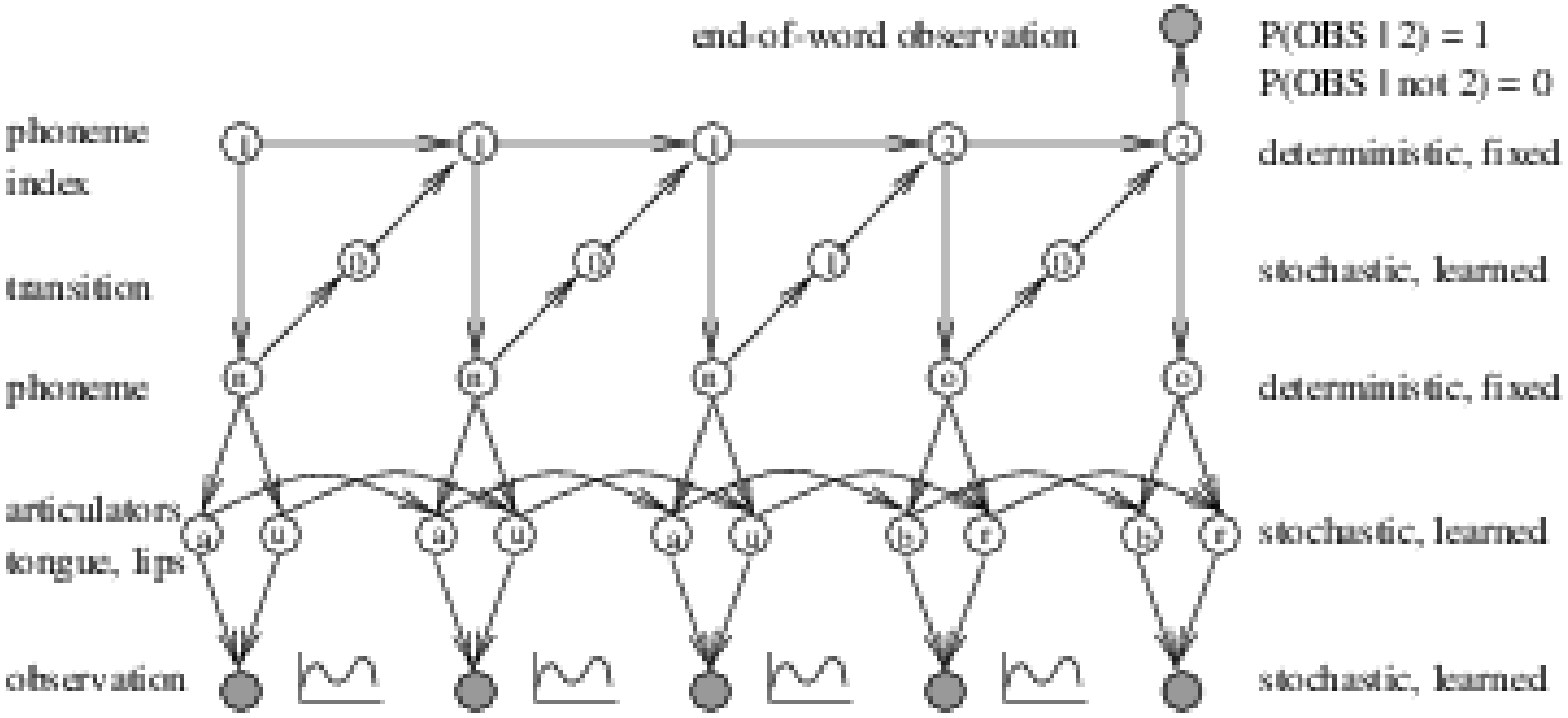
- Train by counting all word pairs in a large text corpus

- More sophisticated models (trigrams, grammars, etc.) help

Combined HMM

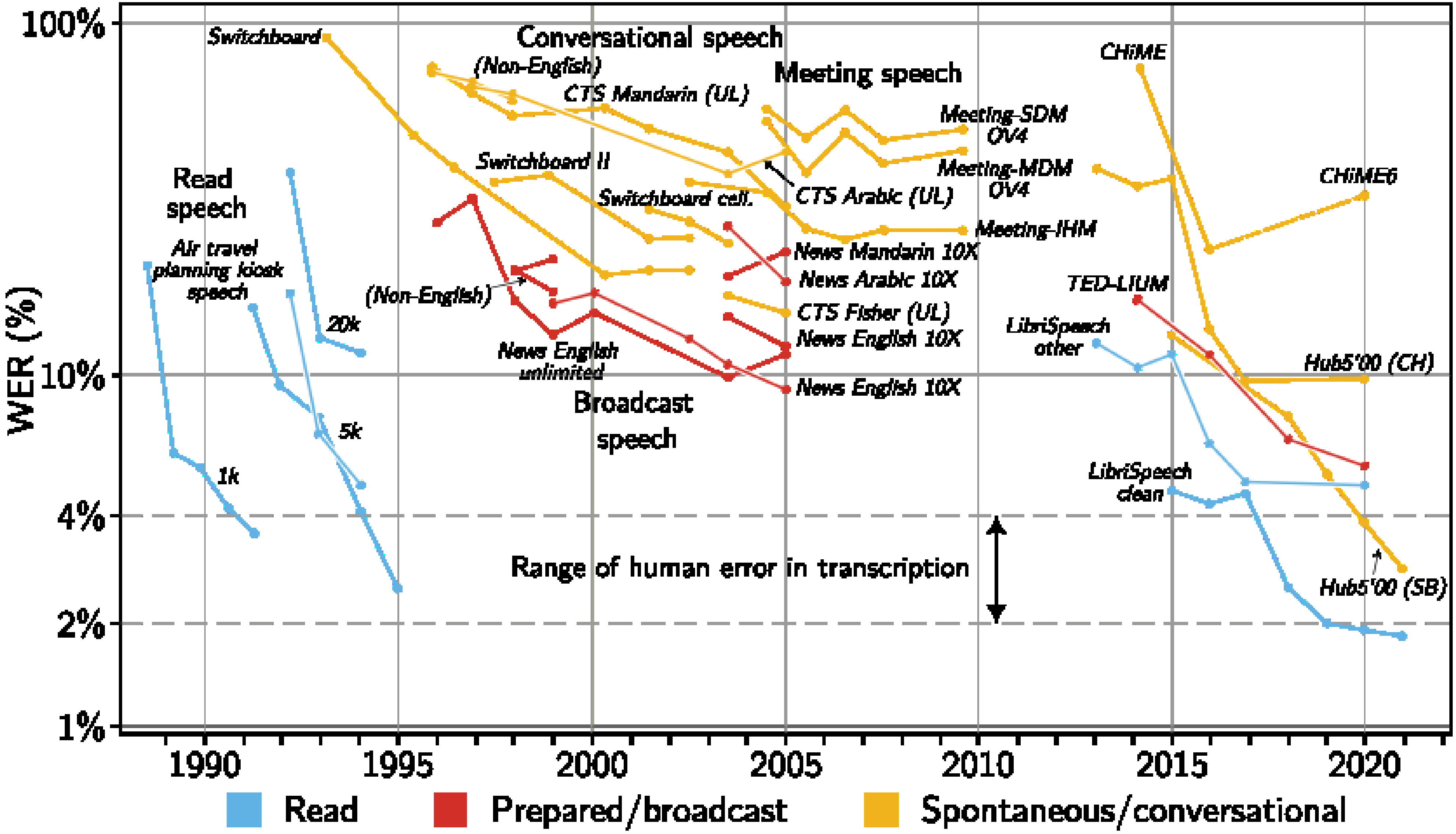
- States of the combined language+word+phone model are labelled by
  - the word we are in
  - the phone in that word
  - the phone state in that phone
- Viterbi algorithm finds the most likely **phone state** sequence
- Segmentation by considering all possible word sequences and boundaries
- Does not always give the most likely word sequence because each word sequence is the sum over many state sequences

DBNs for Speech Recognition



- Also easy to add variables for, e.g., gender, accent, speed

Progress


WER (%)
100%
10%
4%
2%
1%
1990
1995
2000
2005
2010
2015
2020
Switchboard
Conversational speech
(Non-English)
CTS Mandarin (UL)
Meeting speech
Meeting-SDM OV4
Meeting-MDM OV4
CTS Arabic (UL)
Meeting-IHM
CHiME
CHiME6
Read speech
Air travel planning kiosk speech
20k
5k
1k
Switchboard II
Switchboard cell.
(Non-English)
News Mandarin 10X
News Arabic 10X
CTS Fisher (UL)
News English 10X
News English unlimited
News English 10X
Broadcast speech
TED-LIUM
LibriSpeech other
LibriSpeech clean
Hub5'00 (CH)
Hub5'00 (SB)
Range of human error in transcription
🟦 Read
🟥 Prepared/broadcast
🟨 Spontaneous/conversational

Summary

- Temporal models use state and sensor variables replicated over time
- Markov assumptions and stationarity assumption, so we need
  – transition model$\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
  – sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$
- Tasks are filtering, smoothing, most likely sequence;
  all done recursively with constant cost per time step
- Hidden Markov models have a single discrete state variable; used for speech recognition
- Dynamic Bayes nets subsume HMMs
- Speech recognition