TDT4171 Artificial Intelligence Methods

Lecture 4 – Probabilistic Reasoning over Time

Norwegian University of Science and Technology

Helge Langseth
Gamle Fysikk 255
helge.langseth@ntnu.no

NTNU

Outline

1. Summary from last time
2. Probabilistic Reasoning over Time
   - Set-up
   - Example: Basic speech recognition
   - Inference: Filtering, prediction, smoothing, most probable
   - Dynamic Bayesian networks
3. Summary

Summary from Chapter 13

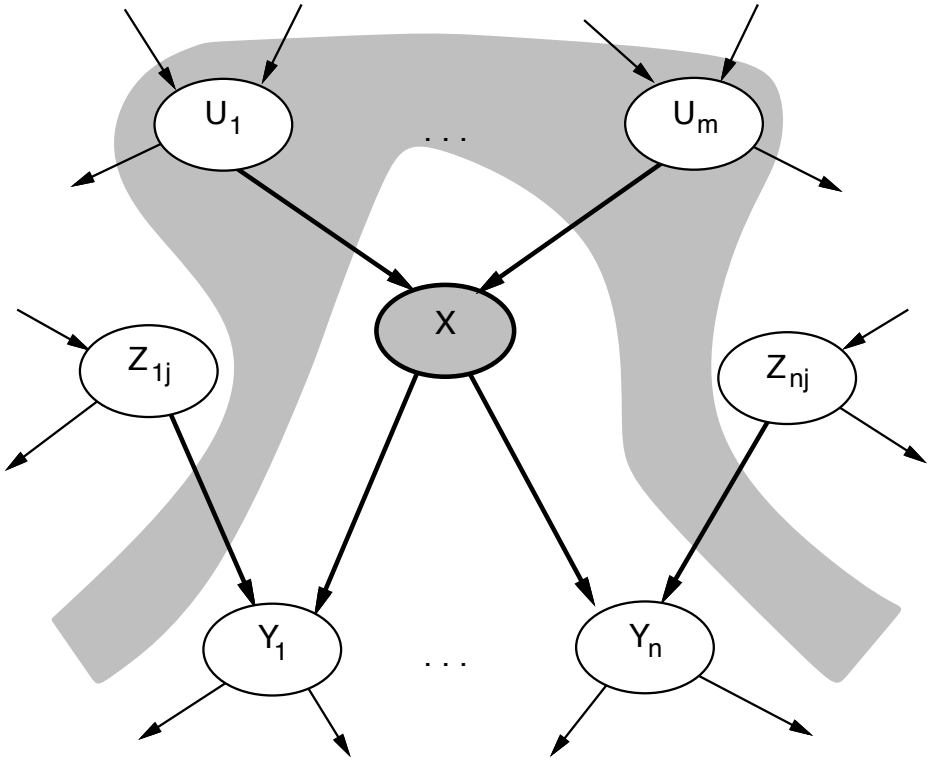
- **Bayes nets** provide a natural representation for (causally induced) conditional independence
- Topology + CPTs = **compact representation** of joint distribution
- Generally **easy to construct** – also for non-experts
- **Canonical distributions** (e.g., noisy-OR) = compact representation of CPTs very useful if a node has many parents
- **Efficient inference** calculations are available (but the good ones are outside the scope of this course)

What you should know:

- How to build models (and verify them using Conditional Independence and Causality)
- What drives the ...
  - model building burden
  - complexity of inference

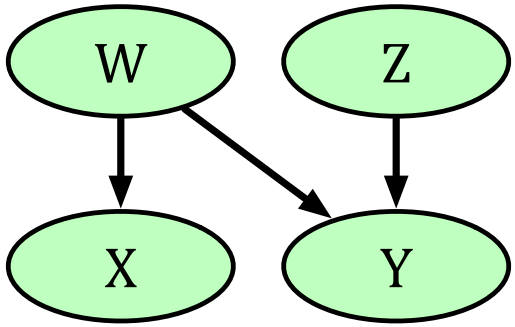
Remember the structural/local semantics

Local semantics: each node is conditionally independent of its non-descendants given its parents



Local/Structural semantics $\Leftrightarrow$ Global/Quantitative semantics

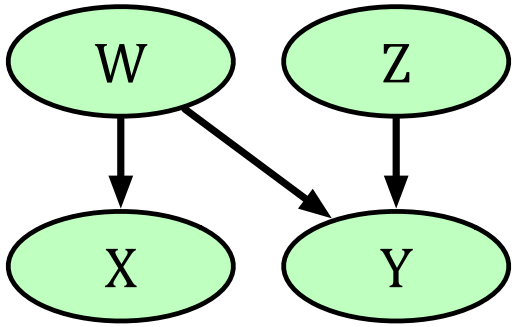
SUFFICIENT and not REQUIRED conditioning set



Let's focus on Y:

- General rule: $Y \perp\!\!\!\perp \text{Non-Descendants}(Y) \mid \text{Parents}(Y)$.
- Here: $Y \perp\!\!\!\perp X \mid \{W, Z\}$.
- With symmetry: $X \perp\!\!\!\perp Y \mid \{W, Z\}$.

SUFFICIENT and not REQUIRED conditioning set



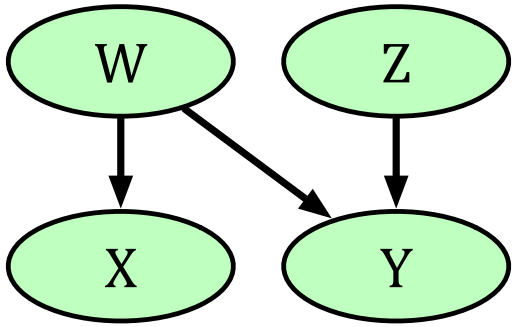
Let's focus on Y:

- General rule: $Y \perp\!\!\!\perp \text{Non-Descendants}(Y) \mid \text{Parents}(Y)$.
- Here: $Y \perp\!\!\!\perp X \mid \{W, Z\}$.
- With symmetry: $X \perp\!\!\!\perp Y \mid \{W, Z\}$.

And now, focus on X:

- General rule: $X \perp\!\!\!\perp \text{Non-Descendants}(X) \mid \text{Parents}(X)$.
- Here: $X \perp\!\!\!\perp \{Y, Z\} \mid W$. In particular: $X \perp\!\!\!\perp Y \mid W$.

SUFFICIENT and not REQUIRED conditioning set



Let's focus on Y:

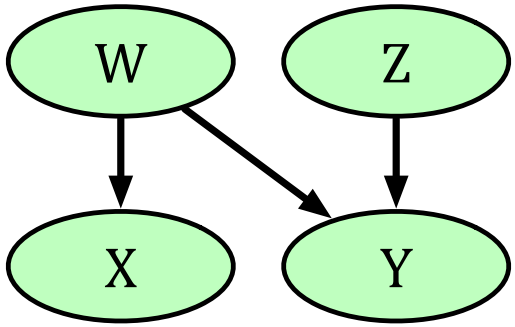
- General rule: $Y \perp\!\!\!\perp \text{Non-Descendants}(Y) \mid \text{Parents}(Y)$.
- Here: $Y \perp\!\!\!\perp X \mid \{W, Z\}$.
- With symmetry: $X \perp\!\!\!\perp Y \mid \{W, Z\}$.

And now, focus on X:

- General rule: $X \perp\!\!\!\perp \text{Non-Descendants}(X) \mid \text{Parents}(X)$.
- Here: $X \perp\!\!\!\perp \{Y, Z\} \mid W$. In particular: $X \perp\!\!\!\perp Y \mid W$.

Both $X \perp\!\!\!\perp Y \mid \{W, Z\}$ and $X \perp\!\!\!\perp Y \mid W$ are true statements.
Typically, the simpler one is preferred.

SUFFICIENT and not REQUIRED conditioning set

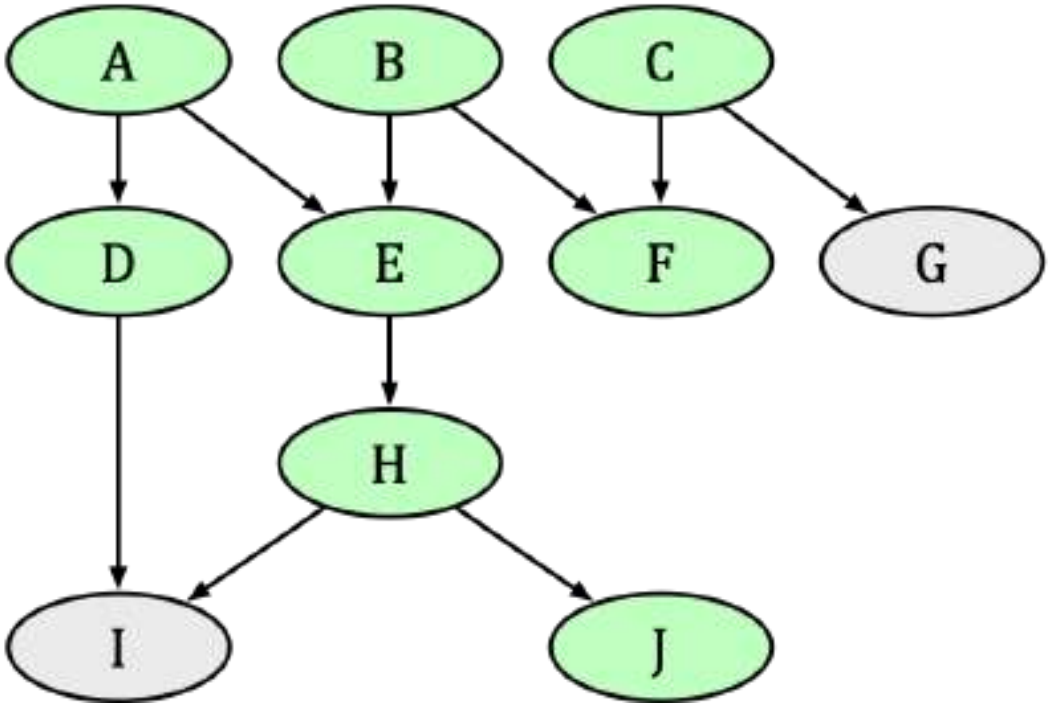


Note that introducing new conditionings **can** break previous independence-statements:

- $W \perp\!\!\!\perp Z$ and $W \perp\!\!\!\perp Z \mid X$.
- **However**, $W \not\perp\!\!\!\perp Z \mid Y$ and $W \not\perp\!\!\!\perp Z \mid \{X, Y\}$!!
- Knowing the common child Y "connects" the parents $\{W, Z\}$.
- The Markov-blanket-rule helps in these situations.

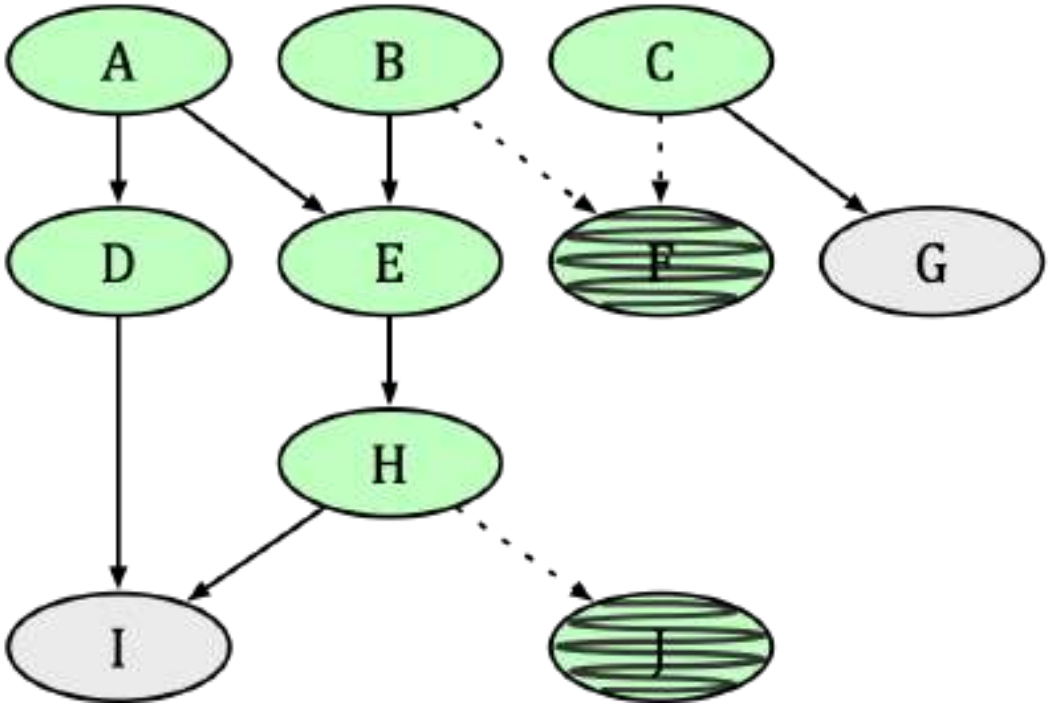
I admit that this stuff is somewhat difficult in the beginning, but I'll also claim that it is **fairly easy** with some practice...

A slightly more involved example



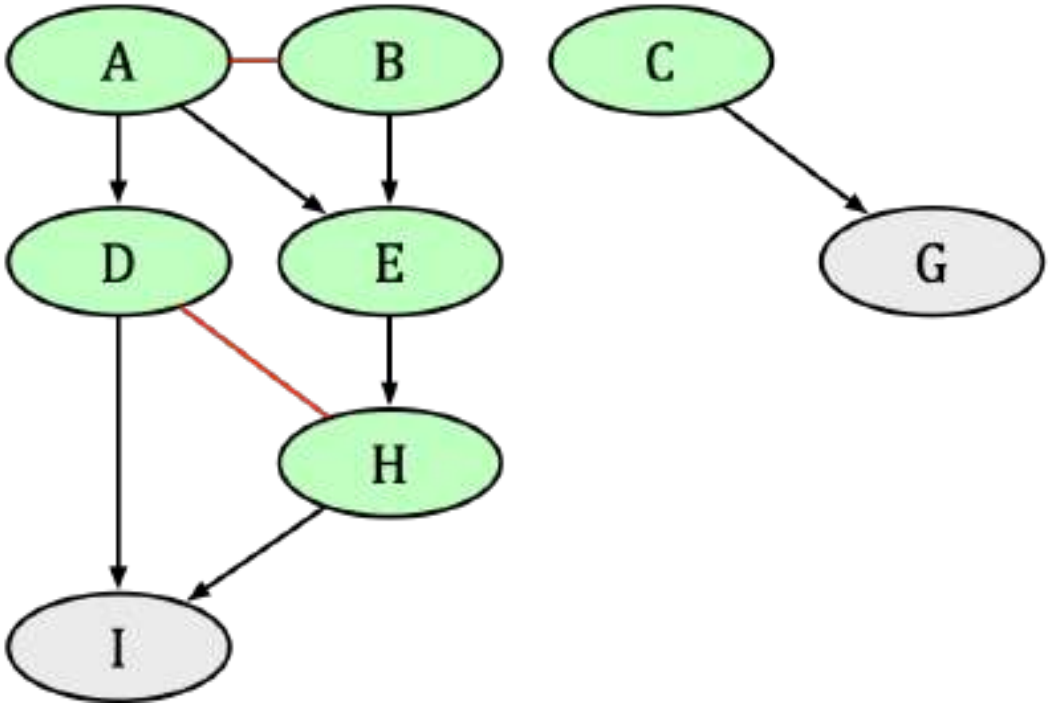
Is $I \perp\!\!\!\perp G$?

A slightly more involved example



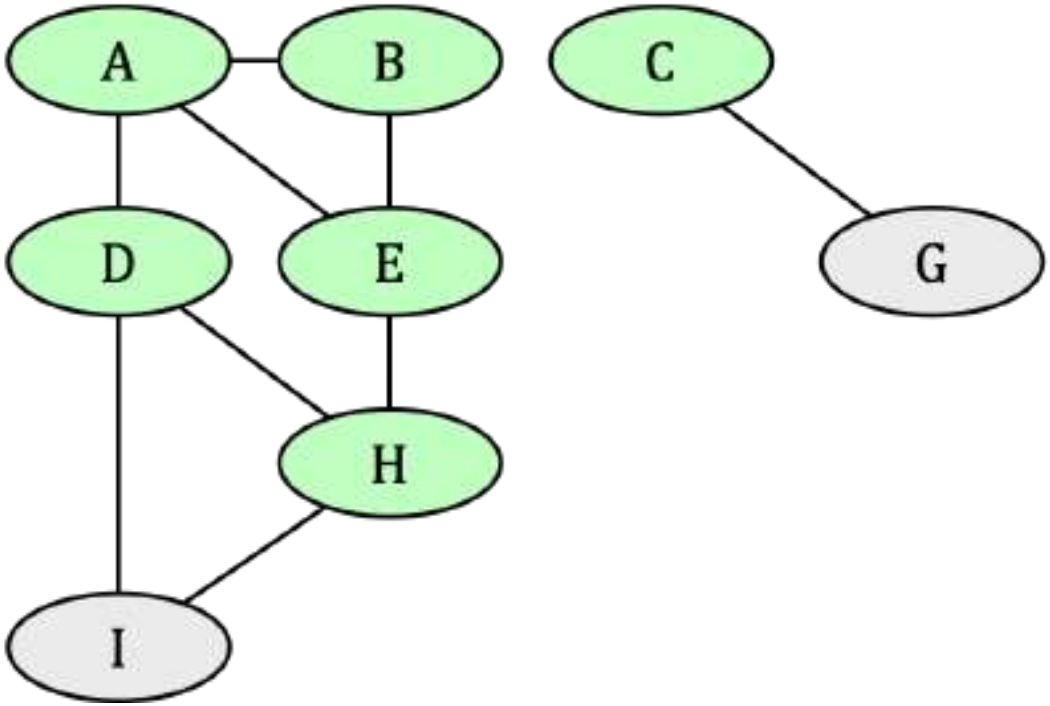
Is $I \perp\!\!\!\perp G$?

A slightly more involved example



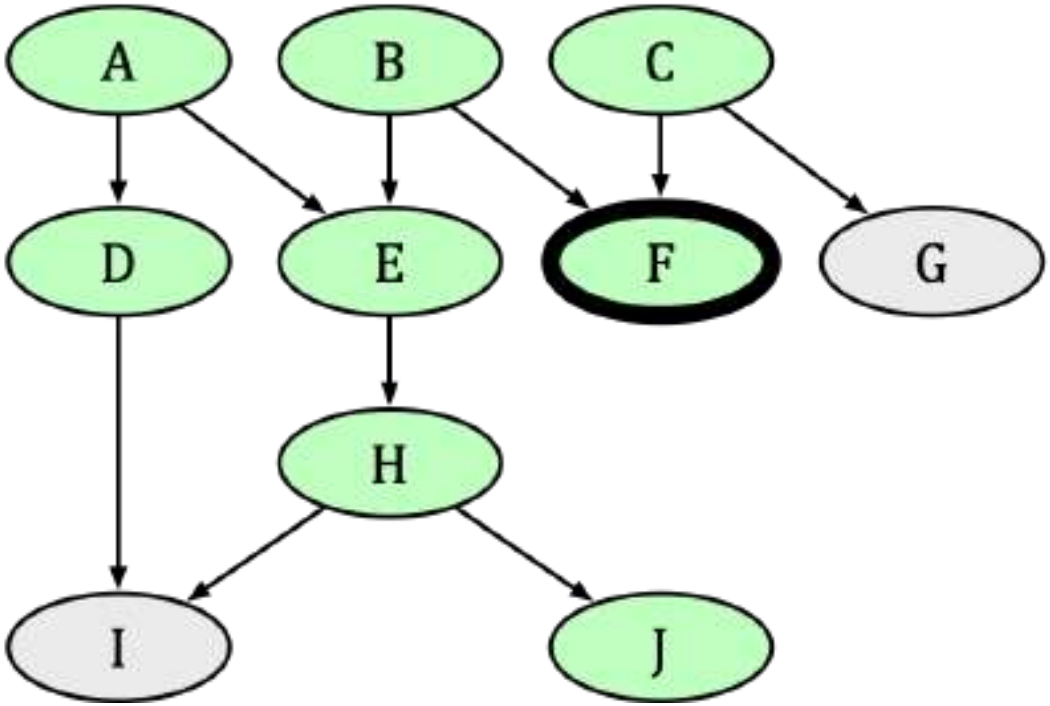
Is $I \perp\!\!\!\perp G$?

A slightly more involved example



Is $I \perp\!\!\!\perp G$?

A slightly more involved example

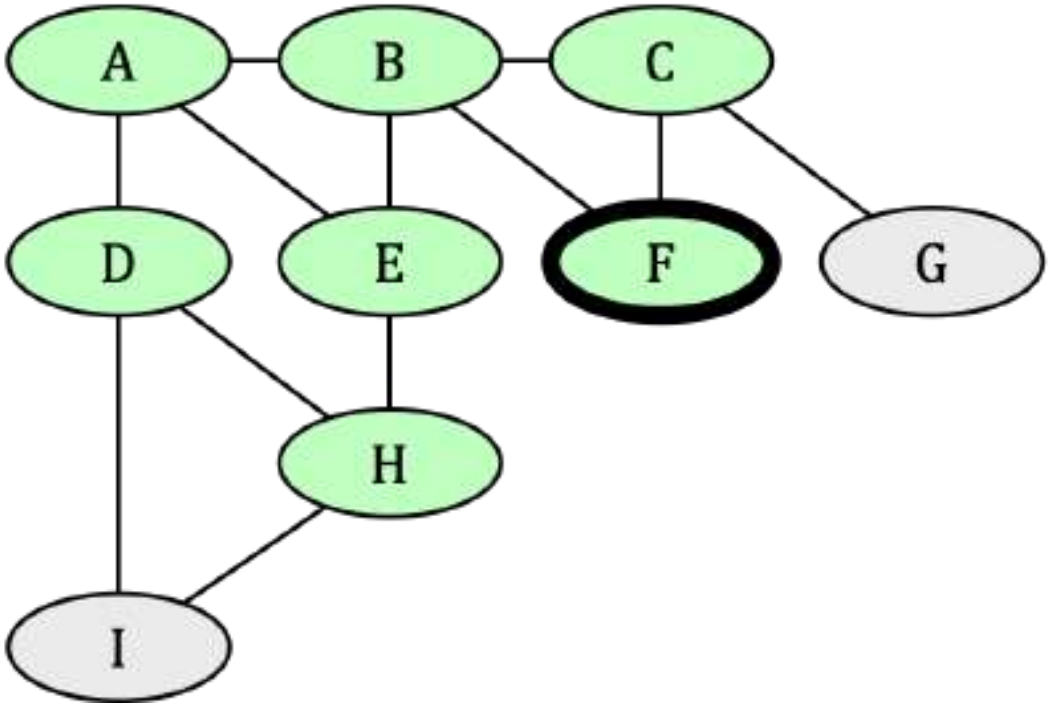


Is $I \perp\!\!\!\perp G$?

Is $I \perp\!\!\!\perp G \mid F$?

Yes!

A slightly more involved example

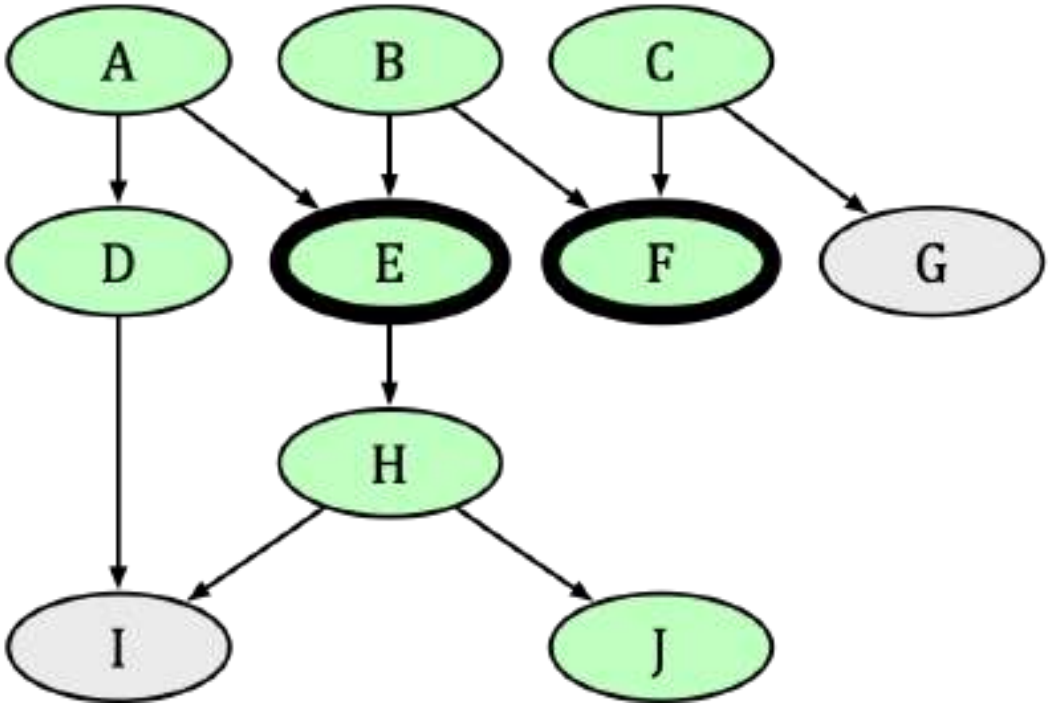


Is $I \perp\!\!\!\perp G$?

Is $I \perp\!\!\!\perp G \mid F$?

Yes!

A slightly more involved example

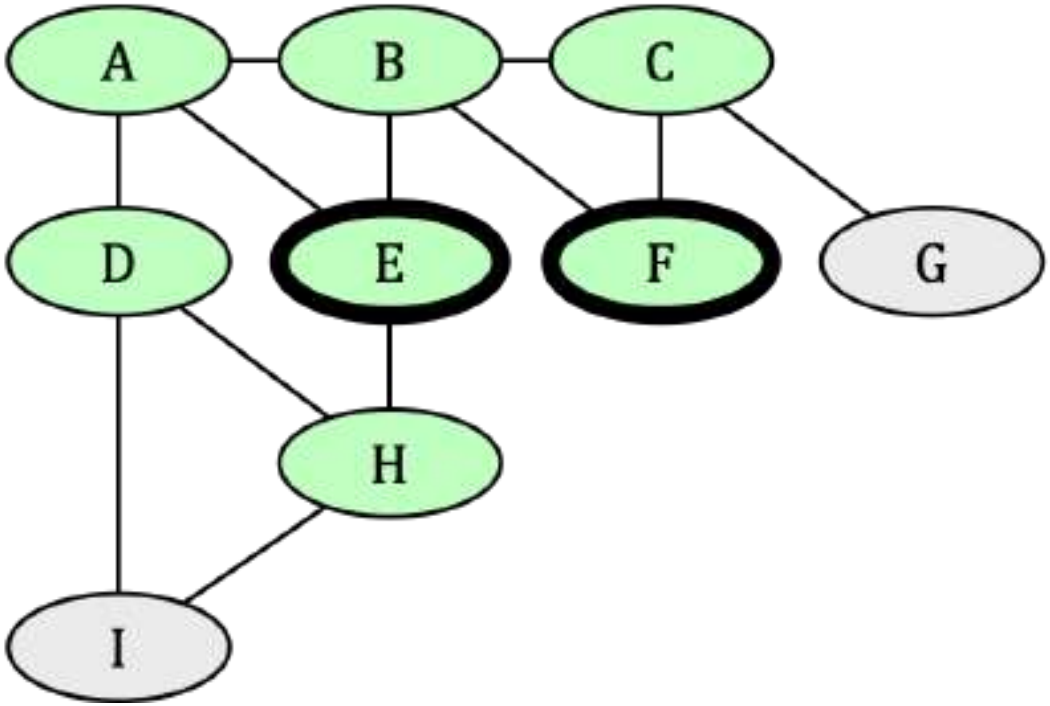


Is $I \perp\!\!\!\perp G$? **Yes!**

Is $I \perp\!\!\!\perp G \mid F$? **No!**

Is $I \perp\!\!\!\perp G \mid \{E, F\}$?

A slightly more involved example

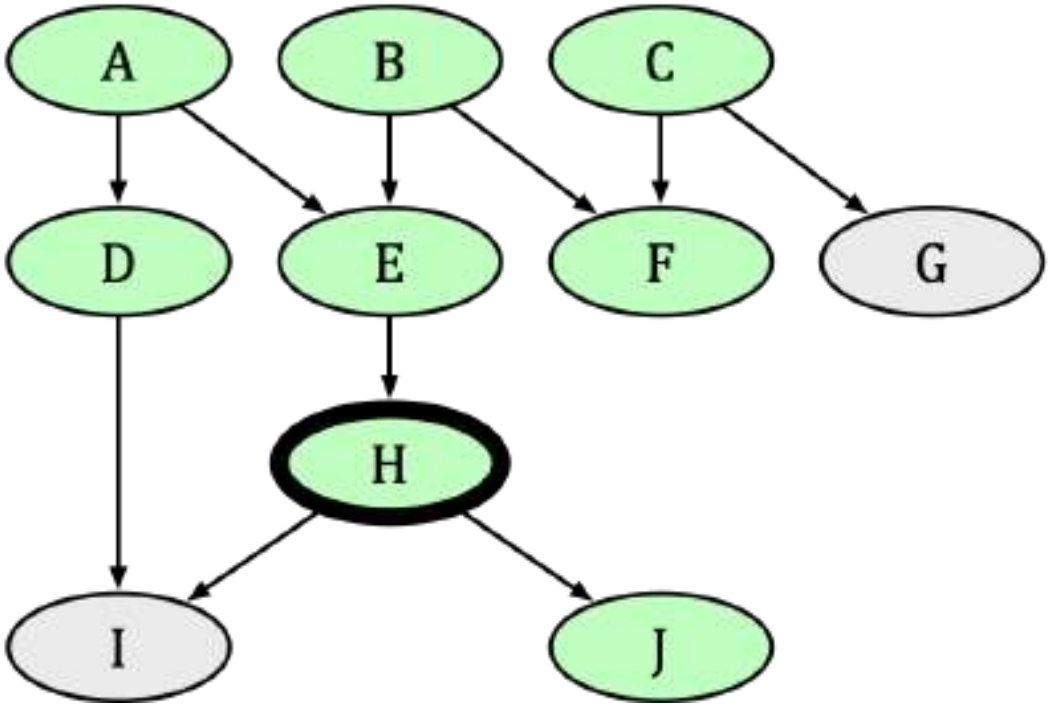


Is $I \perp\!\!\!\perp G$? **Yes!**

Is $I \perp\!\!\!\perp G \mid F$? **No!**

Is $I \perp\!\!\!\perp G \mid \{E, F\}$?

A slightly more involved example



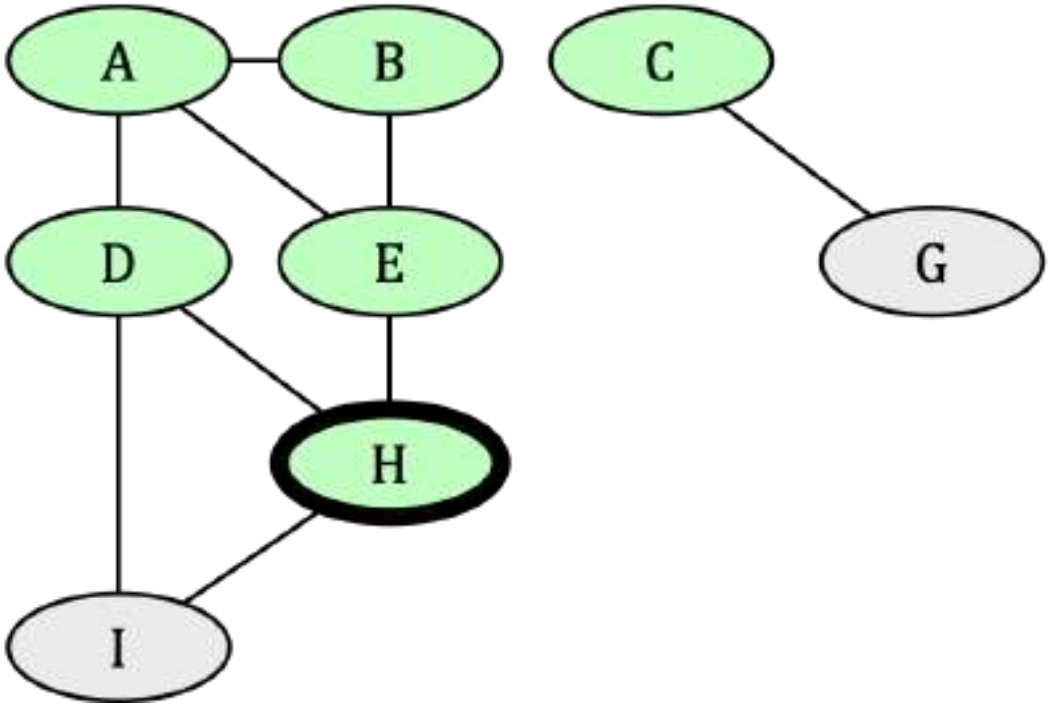
Is $I \perp\!\!\!\perp G$? **Yes!**

Is $I \perp\!\!\!\perp G \mid F$? **No!**

Is $I \perp\!\!\!\perp G \mid \{E, F\}$? **No!**

Is $I \perp\!\!\!\perp G \mid H$?

A slightly more involved example



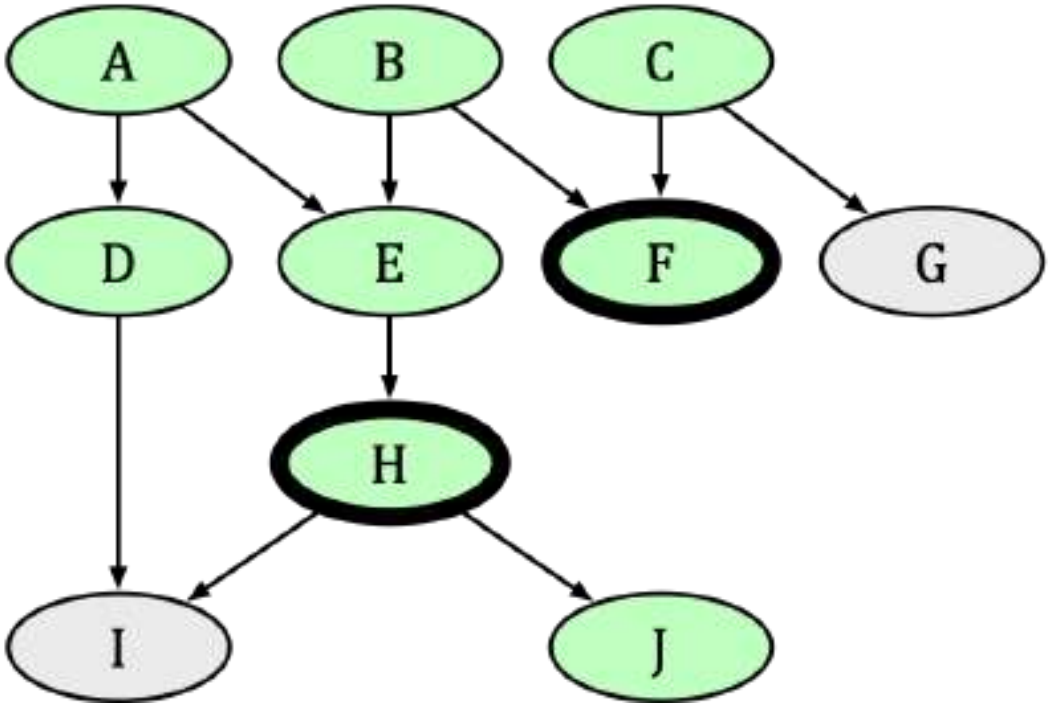
Is $I \perp\!\!\!\perp G$? **Yes!**

Is $I \perp\!\!\!\perp G \mid F$? **No!**

Is $I \perp\!\!\!\perp G \mid \{E, F\}$? **No!**

Is $I \perp\!\!\!\perp G \mid H$?

A slightly more involved example



Is $I \perp\!\!\!\perp G$? **Yes!**

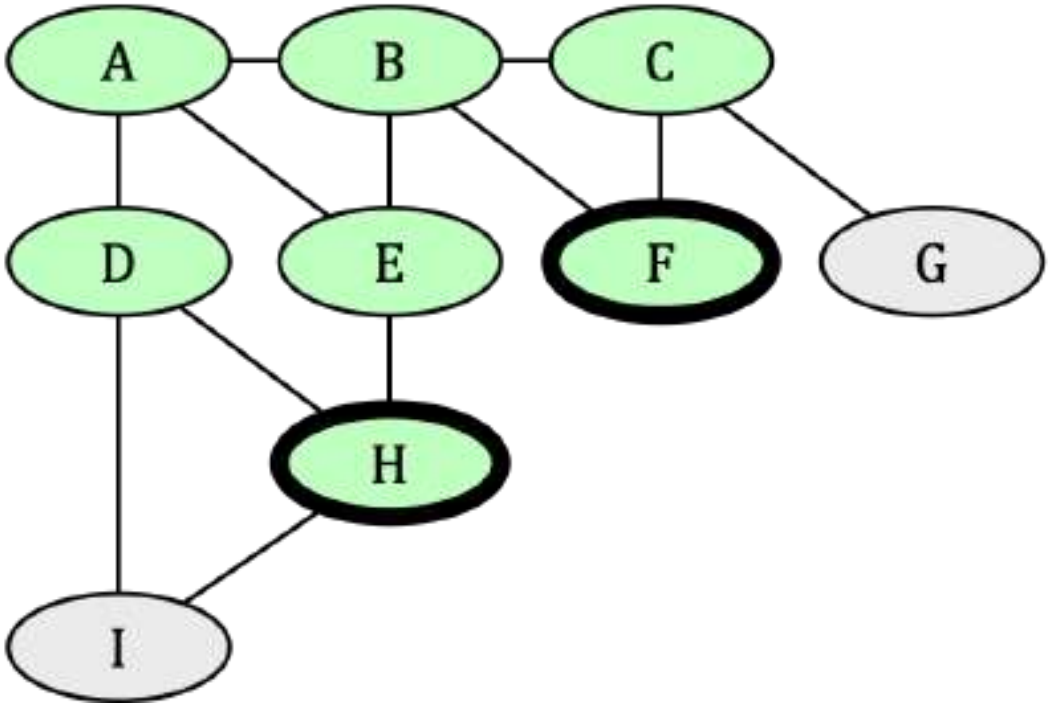
Is $I \perp\!\!\!\perp G \mid F$? **No!**

Is $I \perp\!\!\!\perp G \mid \{E, F\}$? **No!**

Is $I \perp\!\!\!\perp G \mid H$? **Yes!**

Is $I \perp\!\!\!\perp G \mid \{H, F\}$?

A slightly more involved example



Is $I \perp\!\!\!\perp G$? **Yes!**

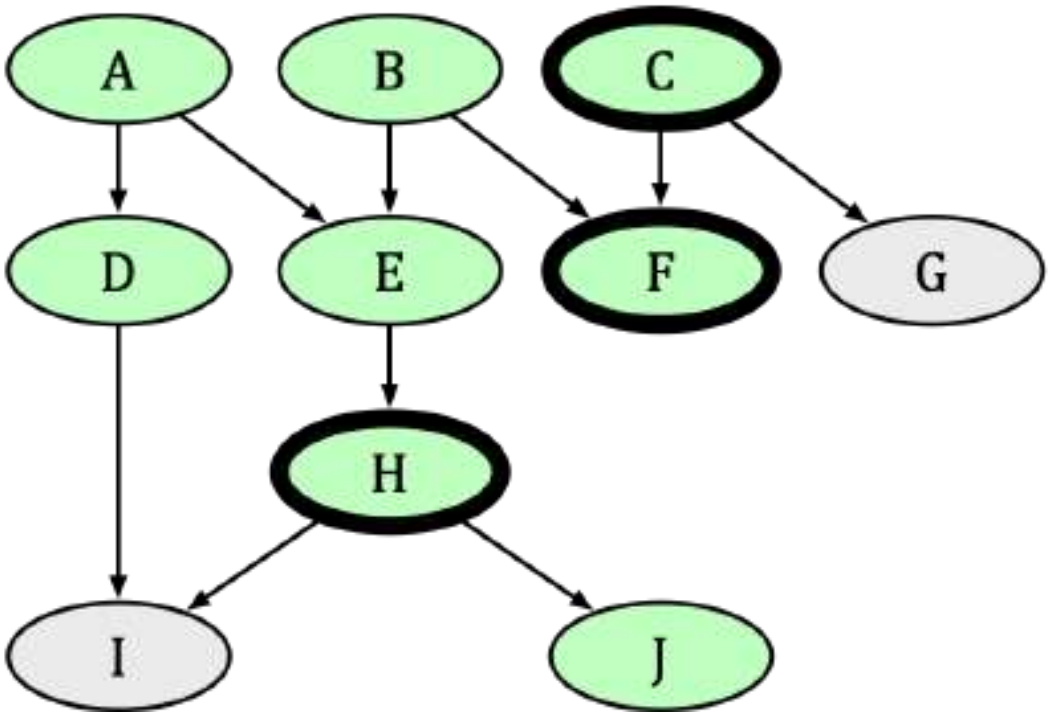
Is $I \perp\!\!\!\perp G \mid F$? **No!**

Is $I \perp\!\!\!\perp G \mid \{E, F\}$? **No!**

Is $I \perp\!\!\!\perp G \mid H$? **Yes!**

Is $I \perp\!\!\!\perp G \mid \{H, F\}$?

A slightly more involved example

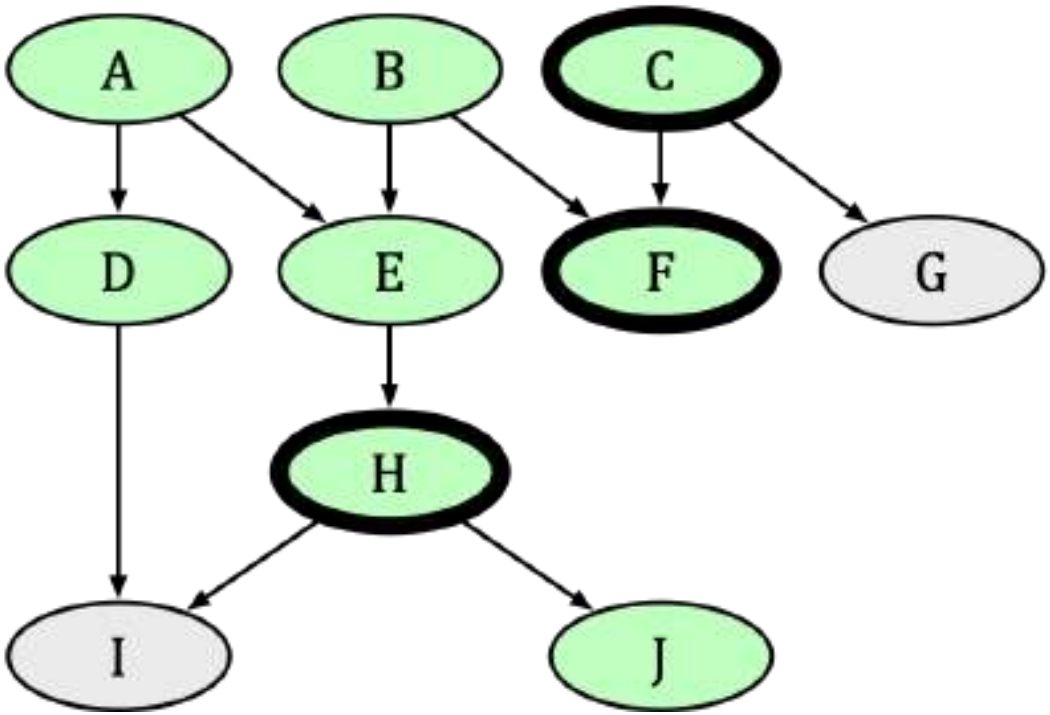


Is $I \perp\!\!\!\perp G$? **Yes!**
Is $I \perp\!\!\!\perp G \mid F$? **No!**
Is $I \perp\!\!\!\perp G \mid \{E, F\}$? **No!**

Is $I \perp\!\!\!\perp G \mid H$? **Yes!**
Is $I \perp\!\!\!\perp G \mid \{H, F\}$? **No!**
Is $I \perp\!\!\!\perp G \mid \{H, F, C\}$?

Discuss with your neighbour for a couple of minutes
Try to use **both** solution strategies we have discussed!

A slightly more involved example

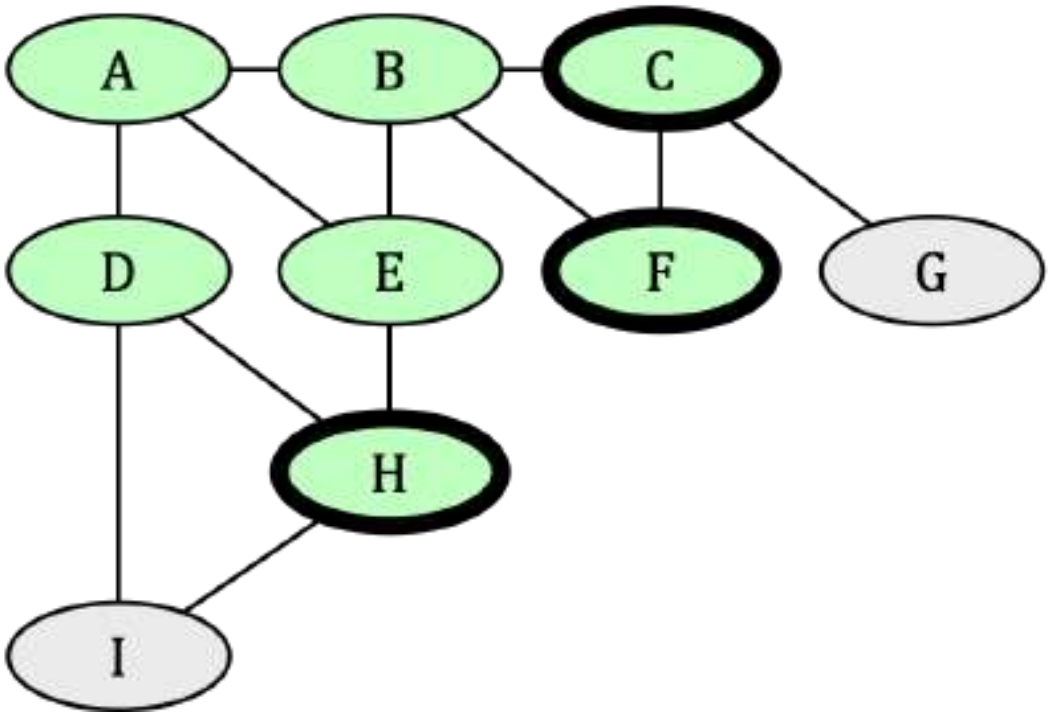


Is $I \perp\!\!\!\perp G$? **Yes!**
Is $I \perp\!\!\!\perp G \mid F$? **No!**
Is $I \perp\!\!\!\perp G \mid \{E, F\}$? **No!**

Is $I \perp\!\!\!\perp G \mid H$? **Yes!**
Is $I \perp\!\!\!\perp G \mid \{H, F\}$? **No!**
Is $I \perp\!\!\!\perp G \mid \{H, F, C\}$?

Discuss with your neighbour for a couple of minutes
Try to use **both** solution strategies we have discussed!

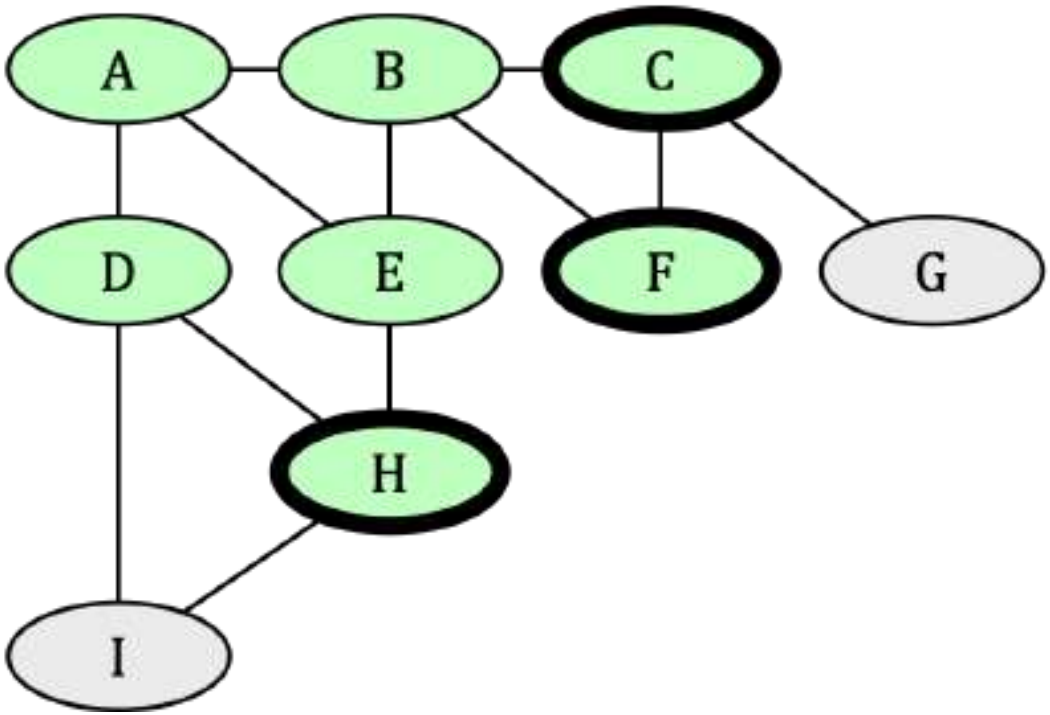
A slightly more involved example



Is $I \perp\!\!\!\perp G$? **Yes!**
Is $I \perp\!\!\!\perp G \mid F$? **No!**
Is $I \perp\!\!\!\perp G \mid \{E, F\}$? **No!**

Is $I \perp\!\!\!\perp G \mid H$? **Yes!**
Is $I \perp\!\!\!\perp G \mid \{H, F\}$? **No!**
Is $I \perp\!\!\!\perp G \mid \{H, F, C\}$?

A slightly more involved example



Is $I \perp\!\!\!\perp G$? **Yes!**
Is $I \perp\!\!\!\perp G \mid F$? **No!**
Is $I \perp\!\!\!\perp G \mid \{E, F\}$? **No!**

Is $I \perp\!\!\!\perp G \mid H$? **Yes!**
Is $I \perp\!\!\!\perp G \mid \{H, F\}$? **No!**
Is $I \perp\!\!\!\perp G \mid \{H, F, C\}$? **Yes!**

Learning goals – Chapter 14

Things to know about:

- Ability to evaluate assumptions:
  - Markov property: What is it? When is it useful? Effect?
  - Stationarity: What is it? When is it useful? Effect?
- Standard model-classes:
  - Markov process
  - Hidden Markov models
  - General-purpose dynamic Bayesian networks
- Inference:
  - Inference in simple dynamic BNs (incl. HMMs)

Skills for the assignment:

- Ability to implement "forward-pass" inference in HMMs

Time and uncertainty

Motivation: The world changes; we may need to (1) track it; (2) predict it

Static (Vehicle diagnosis) vs. **Dynamic** (Diabetes management)

Basic idea: copy state and evidence variables for each time step

$\texttt{Rain}_t$ = Does it rain at time t

This assumes **discrete time**; step size depends on problem

Here: A timestep is one day (I guess. . .)

Markov processes (Markov chains)

If we want to construct a Bayes net from these variables, then what are the parents?

- No links from the **future** – we like the causal interpretation
- Assume we have observations of $\texttt{Rain}_0, \texttt{Rain}_1, \ldots, \texttt{Rain}_t$ and want to **predict** whether or not it rains at day $t+1$: $\mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_0, \texttt{Rain}_1, \ldots, \texttt{Rain}_t)$
- Try to build a BN over $\texttt{Rain}_0, \texttt{Rain}_1, \ldots, \texttt{Rain}_{t+1}$:
  - $\mathbf{P}(\texttt{Rain}_{t+1}) \neq \mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_t)$; base on $\texttt{Rain}_t$.
  - $\mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_t) \approx \mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_t, \texttt{Rain}_{t-1}, \ldots, \texttt{Rain}_0)$.

  (My model , My decision!)

Markov processes (Markov chains)

If we want to construct a Bayes net from these variables, then what are the parents?

- No links from the **future** – we like the causal interpretation
- Assume we have observations of $\texttt{Rain}_0, \texttt{Rain}_1, \ldots, \texttt{Rain}_t$ and want to **predict** whether or not it rains at day $t+1$:
  $\mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_0, \texttt{Rain}_1, \ldots, \texttt{Rain}_t)$
- Try to build a BN over $\texttt{Rain}_0, \texttt{Rain}_1, \ldots, \texttt{Rain}_{t+1}$:
  - $\mathbf{P}(\texttt{Rain}_{t+1}) \neq \mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_t)$; base on $\texttt{Rain}_t$.
  - $\mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_t) \approx \mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_t, \texttt{Rain}_{t-1}, \ldots, \texttt{Rain}_0)$.

  (My model , My decision!)

First-order Markov process:

$$\mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_0, \ldots, \texttt{Rain}_t) = \mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_t)$$

"Future is cond. independent of Past given Present"

Markov processes (Markov chains)

If we want to construct a Bayes net from these variables, then what are the parents?

- No links from the **future** – we like the causal interpretation
- Assume we have observations of $\texttt{Rain}_0, \texttt{Rain}_1, \ldots, \texttt{Rain}_t$ and want to **predict** whether or not it rains at day $t+1$:
  $\mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_0, \texttt{Rain}_1, \ldots, \texttt{Rain}_t)$
- Try to build a BN over $\texttt{Rain}_0, \texttt{Rain}_1, \ldots, \texttt{Rain}_{t+1}$:
  - $\mathbf{P}(\texttt{Rain}_{t+1}) \neq \mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_t)$; base on $\texttt{Rain}_t$.
  - $\mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_t) \approx \mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_t, \texttt{Rain}_{t-1}, \ldots, \texttt{Rain}_0)$.

  (My model , My decision!)

***k*'th-order Markov process:**

$\mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_0, \ldots, \texttt{Rain}_t) = \mathbf{P}(\texttt{Rain}_{t+1} | \texttt{Rain}_{t-k+1}, \ldots, \texttt{Rain}_t)$

Notation: $\mathbf{X}_{a:b} = \mathbf{X}_a, \mathbf{X}_{a+1}, \ldots, \mathbf{X}_{b-1}, \mathbf{X}_b$

Markov processes as Bayesian networks

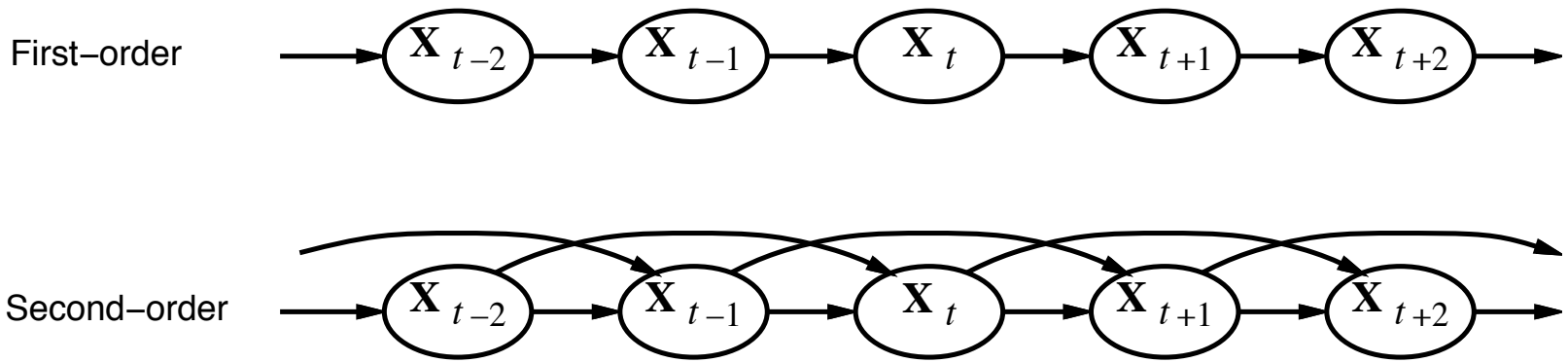
If we want to construct a Bayes net from these variables, then what are the parents?

Markov assumption: X_t depends on **bounded** subset of $X_{0:t-1}$

First-order Markov process: $\mathbf{P}(X_t|X_{0:t-1}) = \mathbf{P}(X_t \mid X_{t-1})$

Second-order Markov process:

$$\mathbf{P}(X_t|X_{0:t-1}) = \mathbf{P}(X_t|X_{t-2}, X_{t-1})$$



(Observable) Markov processes; full set of assumptions

Stationary process:

Transition model $\mathbf{P}(\mathbf{X}_t|\mathrm{pa}(\mathbf{X}_t))$ **fixed** for all $t > 0$

k'th-order Markov process:

$$\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-k:t-1})$$

First-order Markov process – most commonly used:

$$\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$$

Parameters:

- Transition distribution $\boldsymbol{T}$: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-k:t-1})$;
  For $k = 1$ this simplifies to $\boldsymbol{T}$: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
- Prior distribution π: $\mathbf{P}(\mathbf{X}_{0:k-1})$;
  For $k = 1$ this simplifies to π: $\mathbf{P}(\mathbf{X}_0)$

Is a first-order Markov process suitable?

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

Possible fixes:

1. **Increase order** of Markov process
2. **Augment state**, e.g., add $\texttt{Temp}_t$, $\texttt{Pressure}_t$

State augmentation is our preferred solution!

We will focus on first order processes from now on.
Typically produces more understandable models, and with clever modelling, $k = 1$ will suffice.

Unobservable variables – Diabetes example

Consider a **diabetes** patient, who needs to infer the blood sugar level at time t.

Two types of variables:

- $\mathbf{X}_t$ = set of **unobservable state variables** at time t
  e.g., $\texttt{BloodSugar}_t$, $\texttt{StomachContents}_t$, etc.
- $\mathbf{E}_t$ = set of **observable evidence variables** at time t
  e.g., $\texttt{MeasuredBloodSugar}_t$, $\texttt{PulseRate}_t$, $\texttt{FoodEaten}_t$

Note difference between "actual value" (e.g., $\texttt{BloodSugar}_t$) and "measured value" (e.g., $\texttt{MeasuredBloodSugar}_t$)

Unobservable variables – Diabetes example

Consider a **diabetes** patient, who needs to infer the blood sugar level at time t.

Two types of variables:

- $\mathbf{X}_t$ = set of **unobservable state variables** at time t
  e.g., $\texttt{BloodSugar}_t$, $\texttt{StomachContents}_t$, etc.
- $\mathbf{E}_t$ = set of **observable evidence variables** at time t
  e.g., $\texttt{MeasuredBloodSugar}_t$, $\texttt{PulseRate}_t$, $\texttt{FoodEaten}_t$

Note difference between "actual value" (e.g., $\texttt{BloodSugar}_t$) and "measured value" (e.g., $\texttt{MeasuredBloodSugar}_t$)

How can we structure a model over the variables $\{\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$? How do these variables relate within one fixed t? Between different values of t?

Discuss with your neighbour for a couple of minutes

Hidden Markov models

- The "important" variables, X_t, are not observable themselves. They vary over time in some "structured" way.
- X_t is partially disclosed by an observation at time t. We call the observation $\boldsymbol{E}_t$.

Reasonable (sometimes) assumptions to make:

First order Markov process:

$$\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$$

Sensor Markov assumption:

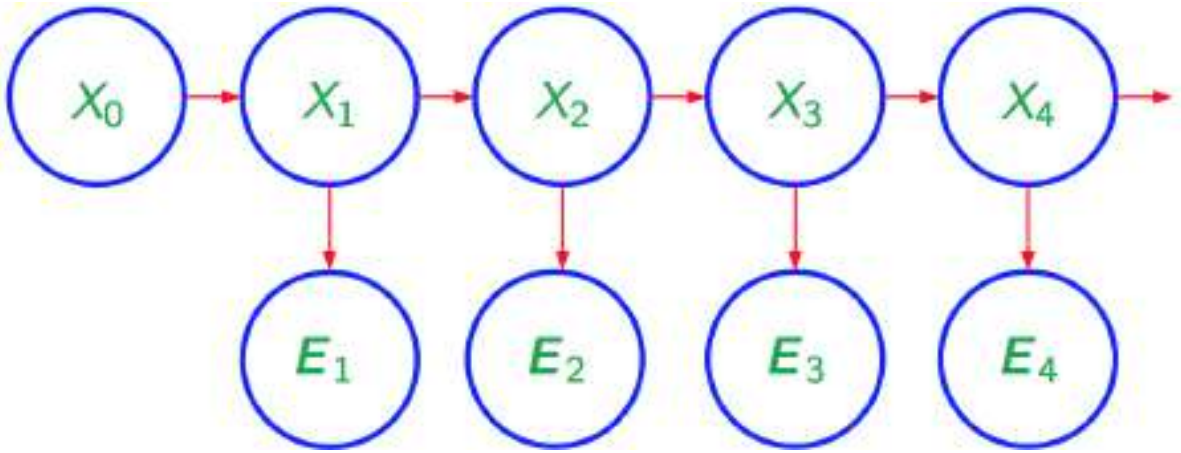
$$\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{1:t}, \mathbf{E}_{1:t-1}) = \mathbf{P}(\mathbf{E}_t|\mathbf{X}_t).$$

Stationary process:

Transition model $\mathbf{P}(\mathbf{X}_t|\mathrm{pa}(\mathbf{X}_t))$ **fixed** for all $t > 0$

We call this model a **Hidden Markov Model**.

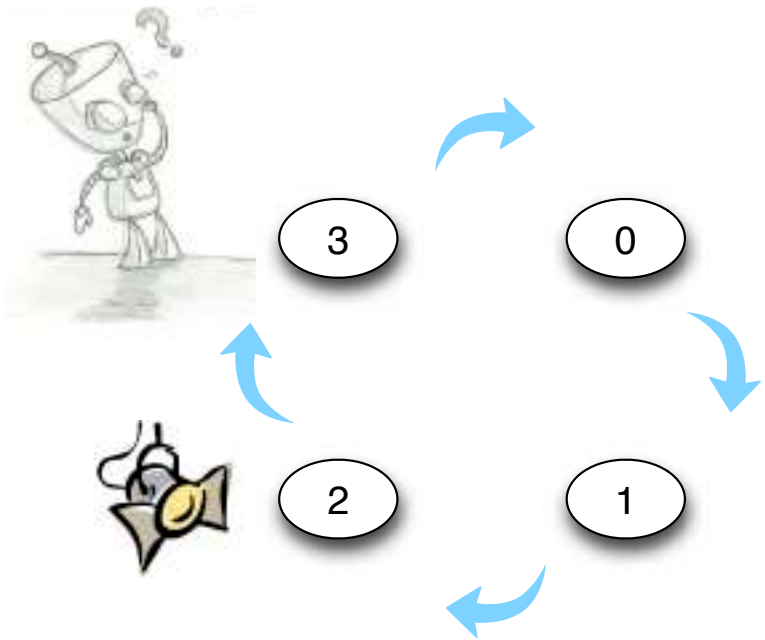
Hidden Markov models as Bayesian networks



The HMM model as a (dynamic) Bayesian net:

- The variables X_t are discrete and one-dimensional
- The variables $\boldsymbol{E}_t$ are vectors of variables

Dynamic Models at work: The confused robot



- **Movements:**
  - Robot moves "up" with probability $p = .8$
  - Robot stands still with probability $p = .15$
  - Robot moves "down" with probability $p = .05$
- **Observations:**
  - Sees light when it is there with $p = 0.8$.
  - Sees light when it is **not** there with $p = 0.1$.

Dynamic Models in GeNIe

Define the "robot problem" as a Bayesian net

- Define variables, the "causal story" and the Bayes net structure. The goal is for the robot to "know" where he is.
- What will the CPTs look like?
- What assumptions have you made? . . . and are they "realistic"?

Discuss with your neighbour for a couple of minutes

Dynamic Models in GeNIe

Define the "robot problem" as a Bayesian net

- Define variables, the "causal story" and the Bayes net structure. The goal is for the robot to "know" where he is.
- What will the CPTs look like?
- What assumptions have you made? . . . and are they "realistic"?

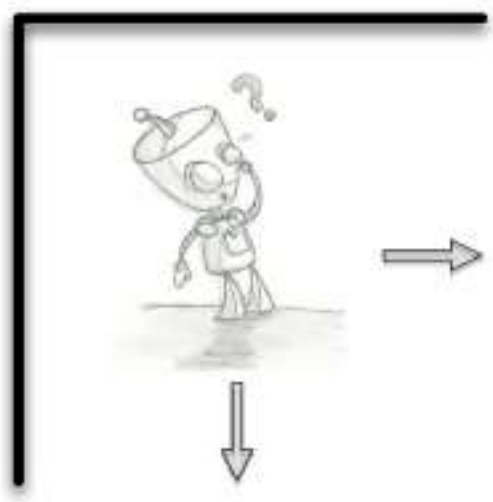
Discuss with your neighbour for a couple of minutes

How to do it in GeNIe:

- GeNIe uses a **plate** reresentation for dynamic models.
- Adds the notion of **temporal links**.
- "Unrolls" the compact model into a standard Bayes net.

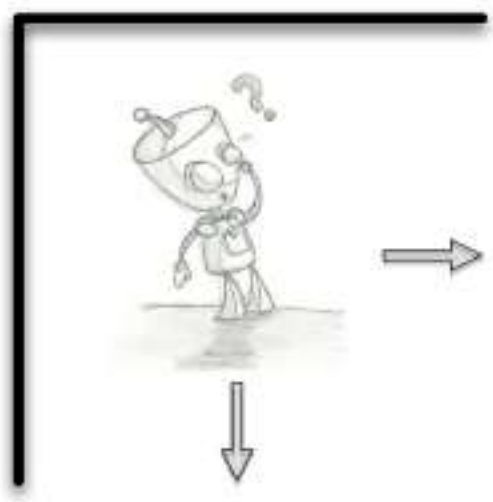
`robot.xdsl`

Another example: The even more confused robot



- **Movement of a robot in a maze:**
  - Robot choses a direction open to it with uniform probability.
- **Observations:**
  - Sees walls "up", "down", "left" and "right" if they are there.
  - Reports each sensor/direction correctly with $p = .85$.
- **Is it difficult to phrase this as an HMM?**

Another example: The even more confused robot



- **Movement of a robot in a maze:**
  - Robot choses a direction open to it with uniform probability.
- **Observations:**
  - Sees walls "up", "down", "left" and "right" if they are there.
  - Reports each sensor/direction correctly with $p = .85$.
- **Is it difficult to phrase this as an HMM?**
- **Of course not!** X_t: Location; E_t: Observation of walls; $P(E_t|X_t)$: Observing walls per location; $P(X_t|X_{t-1})$: Legal moves.

Final example: Speech as probabilistic inference

How can we recognize speech?

- Speech signals are noisy, variable, ambiguous
- What is the **most likely** word, given the speech signal?
- Why not choose `Word` to maximize $\mathbf{P}(\texttt{Word}|\texttt{signal})$??
- Use Bayes' rule:

$$\mathbf{P}(\texttt{Word}|\texttt{signal}) = \alpha\mathbf{P}(\texttt{signal}|\texttt{Word})\mathbf{P}(\texttt{Word})$$

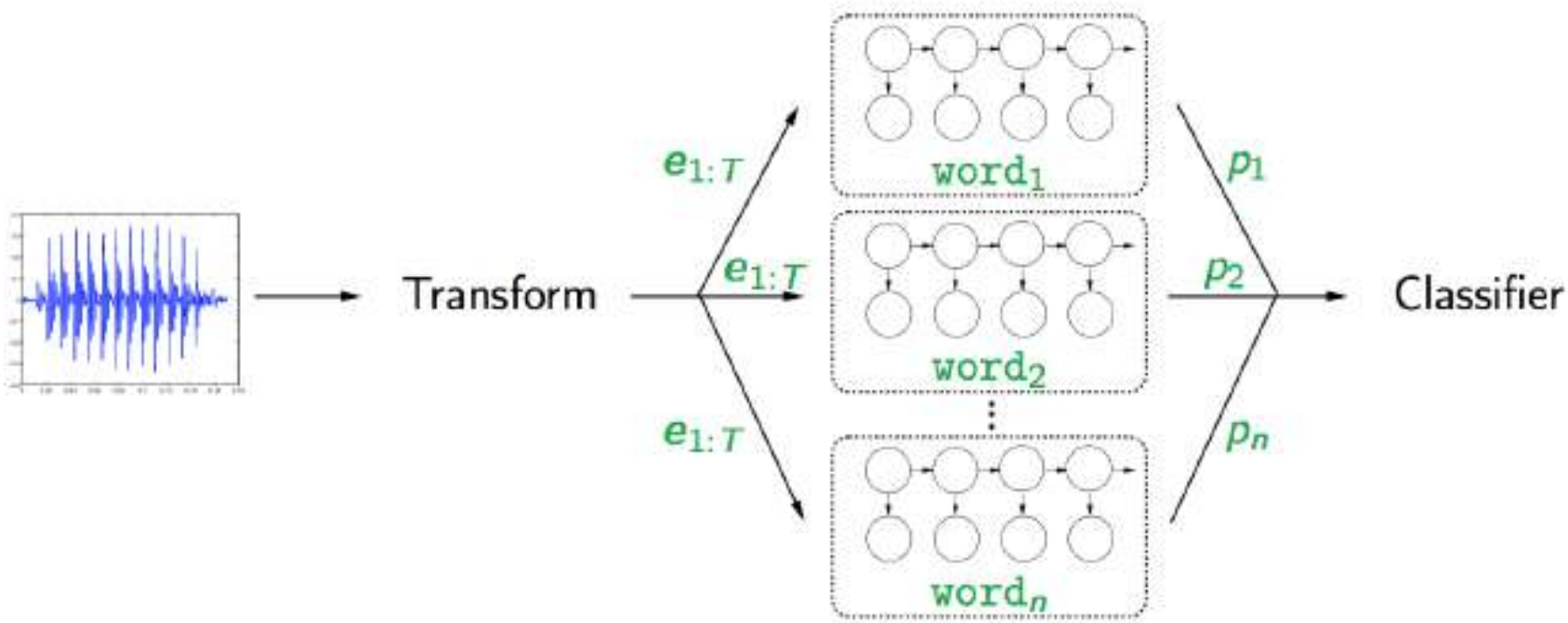
  I.e., decomposes into **acoustic model** + **language model**

- Need to be able to do the required calculations!!

Note! HMMs are no longer state of the art here. Still, current systems use same core ideas, and HMMs are therefore relevant to discuss.

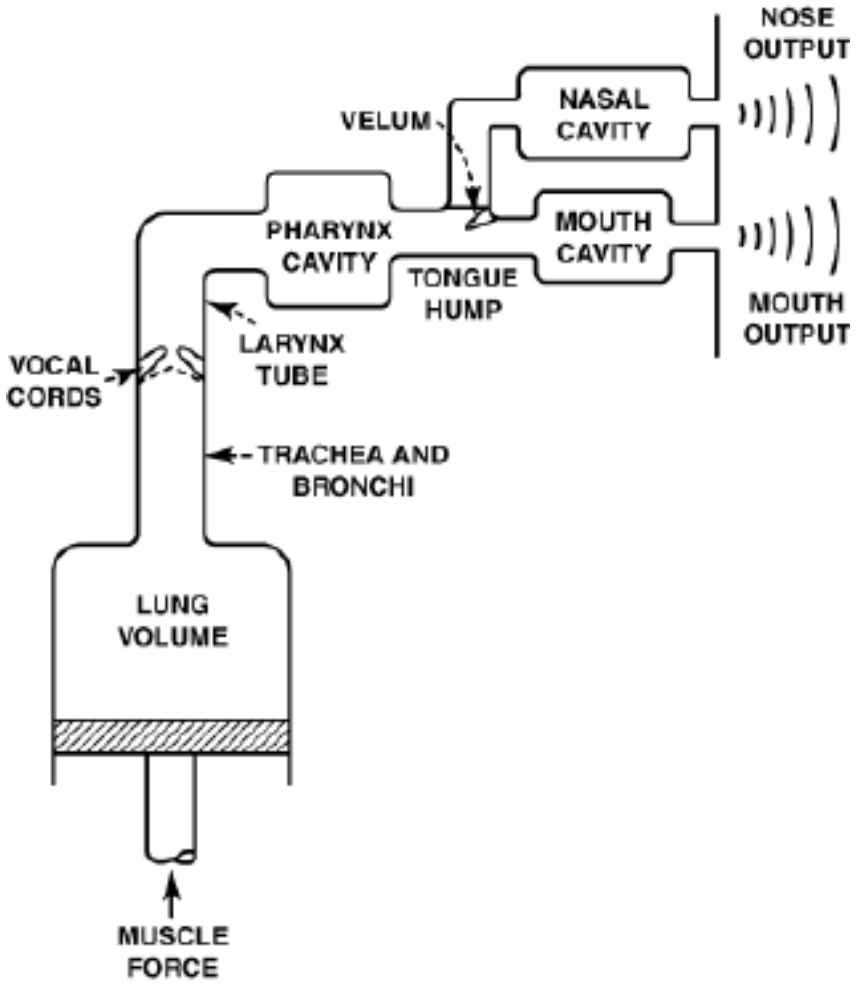
Single-word classifier

Sneak peak – Where this is heading:

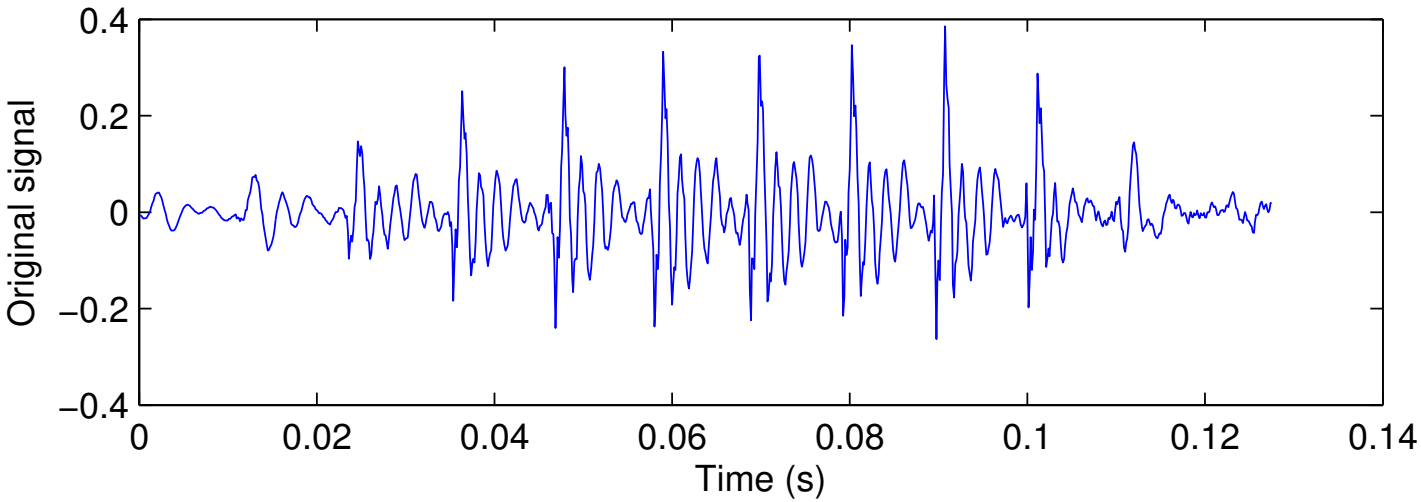


- The top-level structure for the classifier has on model per word.
- Each model reports $p_j = P(\boldsymbol{e}_{1:T}|\texttt{word}_j) \cdot P(\texttt{word}_j)$.

Generation of Speech

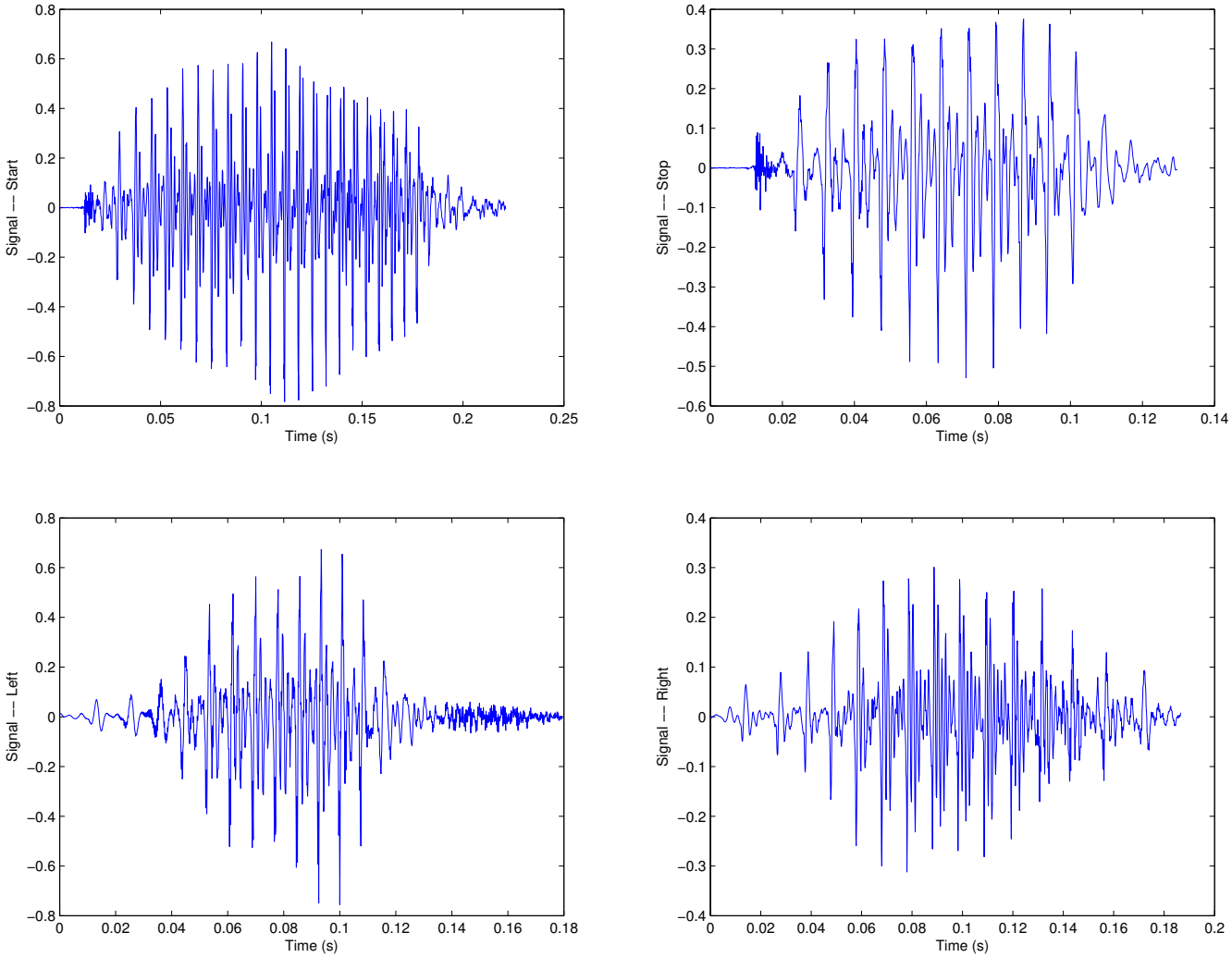


The sound signal – Characteristics



- Sound is **dynamic**, and we must take this into account to represent it faithfully.
- Sound is a "wavy" signal-train, with **amplitude** and **frequency** information changing all the time.
  - Volume of speech $\leftrightarrow$ Global change of amplitudes
  - Speed of speech $\leftrightarrow$ Global change of frequencies
- Most information is carried by the frequencies around 1kHz

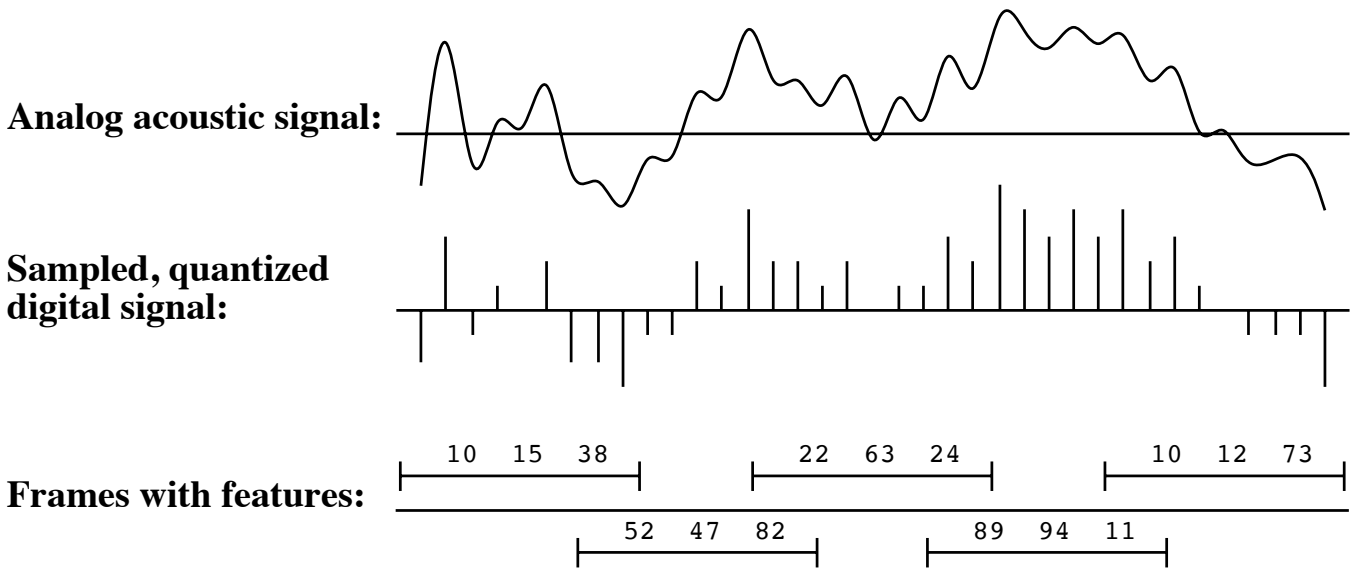
The raw sound for recognition/classification



The raw signal of the the words “Start”, “Stop”, “Left”, and “Right”.

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)

Phones

- All human speech is composed from 40-50 **phones**, determined by the configuration of **articulators**
- Form an intermediate (hidden) level between words and signal
  $\Rightarrow$ **speech of a word** = **uttering a sequence of phones**.
- 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]

Markov processes and speech

Assume we observe phones directly. Let X_t be the phone uttered inside frame t:

- X_t is a single, discrete variable.
- X_t takes on a value from the state-space $\{1, 2, \ldots, N\}$, where N is the total number of phones.
- A useful observation sequence is $\{x_1, x_2, \ldots, x_T\}$ (use $\boldsymbol{x}_{1:T}$ as a shorthand).
- It is common to assume **a Markov process** for speech signals.

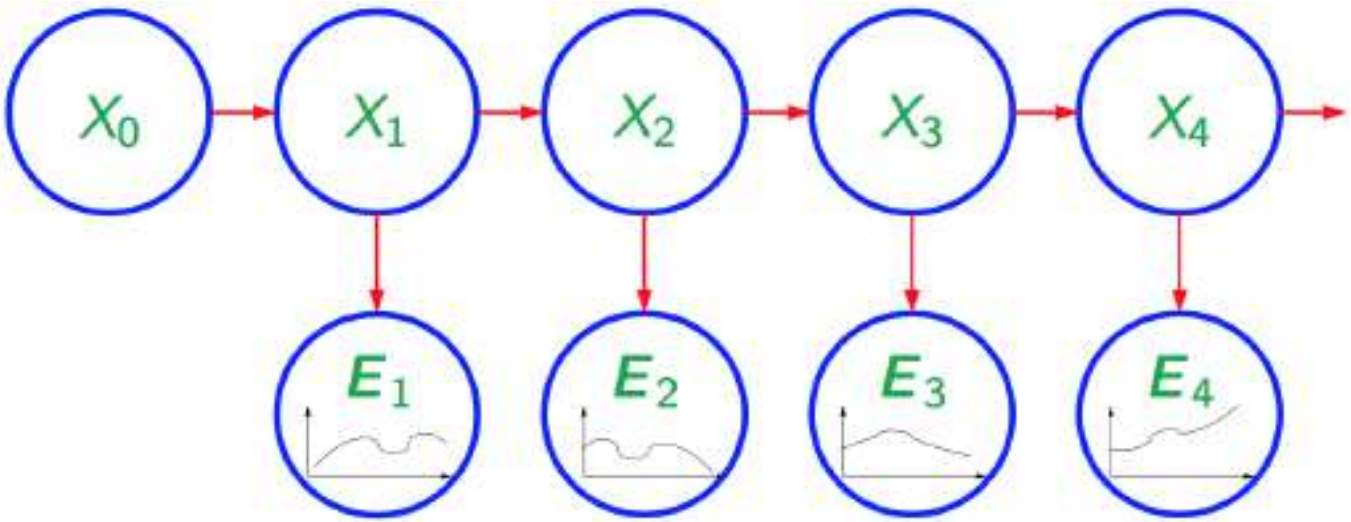
Markov processes and speech

Assume we observe phones directly. Let X_t be the phone uttered inside frame t:

- X_t is a single, discrete variable.
- X_t takes on a value from the state-space $\{1, 2, \ldots, N\}$, where N is the total number of phones.
- A useful observation sequence is $\{x_1, x_2, \ldots, x_T\}$ (use $\boldsymbol{x}_{1:T}$ as a shorthand).
- It is common to assume **a Markov process** for speech signals.

Problem:
We don't observe X_t directly, only the sound signals (which are the speaker's utterances of the phones). Thus, we need an **HMM**!

Hidden Markov models for speech recognition



An HMM model structure for speech analysis:

- The variables X_t are discrete and one-dimensional (representing phones)
- The variables $\boldsymbol{E}_t$ are vectors of variables used to represent the sound signal in a that frame.

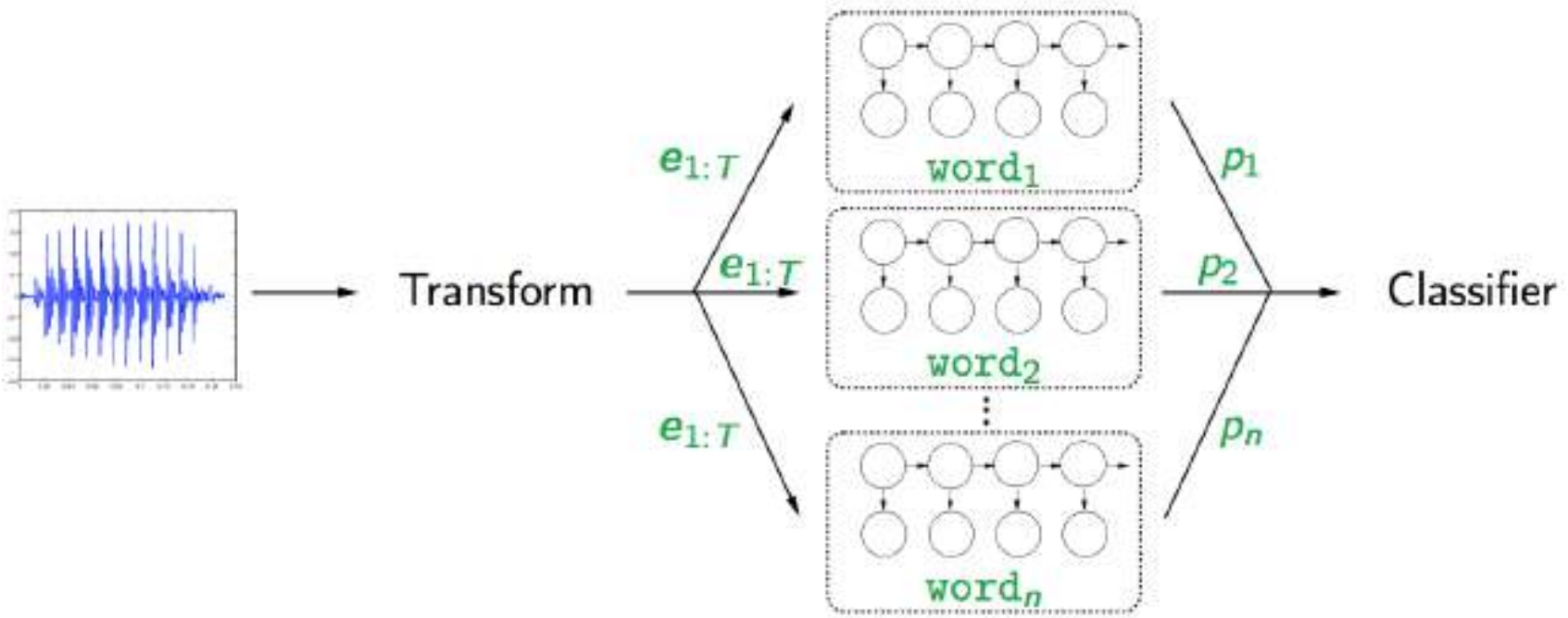
Recognition of isolated words

- Let $\boldsymbol{e}_{1:T}$ denote the observation of a sound signal over T frames.
- Must define model to find likelihood $P(\boldsymbol{e}_{1:T}|\texttt{word})$ for **isolated word**

$$P(\texttt{word}|\boldsymbol{e}_{1:T}) = \alpha P(\boldsymbol{e}_{1:T}|\texttt{word})P(\texttt{word})$$

- Prior probability $P(\texttt{word})$ by counting word frequencies.
- **This leaves us with the problem of calculating $P(\boldsymbol{e}_{1:T}|\texttt{word})$ to make single-word speech recognition.**
- Isolated-word dictation systems with training reach 95% – 99% accuracy

Top level design of a simple classifier



- The top-level structure for the classifier has on model per word.
- The same data is sent to all the different models, and $p_j = P(\boldsymbol{e}_{1:T}|\texttt{word}_j) \cdot P(\texttt{word}_j)$ is returned.
- **Must be able to calculate $P(\boldsymbol{e}_{1:T}|\texttt{word}_j)$ efficiently!**

Inference tasks

Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$. This is the **belief state** – input to the decision process of a rational agent. Also, as a artifact of the calculation scheme, we **can also get the probability needed for speech recognition** if we are interested.

Prediction: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$ for $k > 0$. Evaluation of possible action sequences; like filtering without the evidence

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

Quiz: What kind of inference?

1. In *Dexter*, the main character analyses blood spatter to deduce how a murder has gone down.
2. A *submarine captain* follows the "blips" of a ship on his sonar to understand where the ship is.
3. He fires a torpedo to take the ship out.
4. He continues watching, and after two minutes he says: "We missed. At the time I planned the torpedo to hit, the ship was *there*, not were I aimed".

For each inference, you are asked to decide if it is **Filtering**, **Prediction**, **Smoothing** or **Most Likely Explanation**.

Discuss with your neighbour for a couple of minutes.

Quiz: What kind of inference?

1. In *Dexter*, the main character analyses blood spatter to deduce how a murder has gone down. **Most Likely Explanation.**
2. A *submarine captain* follows the "blips" of a ship on his sonar to understand where the ship is.
3. He fires a torpedo to take the ship out.
4. He continues watching, and after two minutes he says: "We missed. At the time I planned the torpedo to hit, the ship was *there*, not were I aimed".

For each inference, you are asked to decide if it is **Filtering**, **Prediction**, **Smoothing** or **Most Likely Explanation**.

Discuss with your neighbour for a couple of minutes.

Quiz: What kind of inference?

1. In *Dexter*, the main character analyses blood spatter to deduce how a murder has gone down. **Most Likely Explanation.**
2. A *submarine captain* follows the "blips" of a ship on his sonar to understand where the ship is. **Filtering.**
3. He fires a torpedo to take the ship out.
4. He continues watching, and after two minutes he says: "We missed. At the time I planned the torpedo to hit, the ship was *there*, not were I aimed".

For each inference, you are asked to decide if it is **Filtering**, **Prediction**, **Smoothing** or **Most Likely Explanation**.

Discuss with your neighbour for a couple of minutes.

Quiz: What kind of inference?

1. In *Dexter*, the main character analyses blood spatter to deduce how a murder has gone down. **Most Likely Explanation.**
2. A *submarine captain* follows the "blips" of a ship on his sonar to understand where the ship is. **Filtering.**
3. He fires a torpedo to take the ship out. **Prediction.**
4. He continues watching, and after two minutes he says: "We missed. At the time I planned the torpedo to hit, the ship was *there*, not were I aimed".

For each inference, you are asked to decide if it is **Filtering**, **Prediction**, **Smoothing** or **Most Likely Explanation**.

Discuss with your neighbour for a couple of minutes.

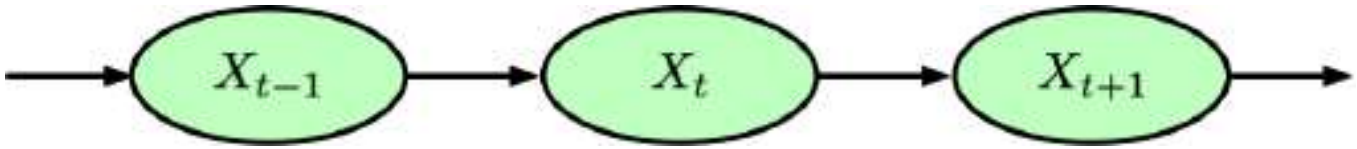
Quiz: What kind of inference?

1. In *Dexter*, the main character analyses blood spatter to deduce how a murder has gone down. **Most Likely Explanation.**
2. A *submarine captain* follows the "blips" of a ship on his sonar to understand where the ship is. **Filtering.**
3. He fires a torpedo to take the ship out. **Prediction.**
4. He continues watching, and after two minutes he says: "We missed. At the time I planned the torpedo to hit, the ship was *there*, not were I aimed". **Smoothing.**

For each inference, you are asked to decide if it is **Filtering**, **Prediction**, **Smoothing** or **Most Likely Explanation**.

Discuss with your neighbour for a couple of minutes.

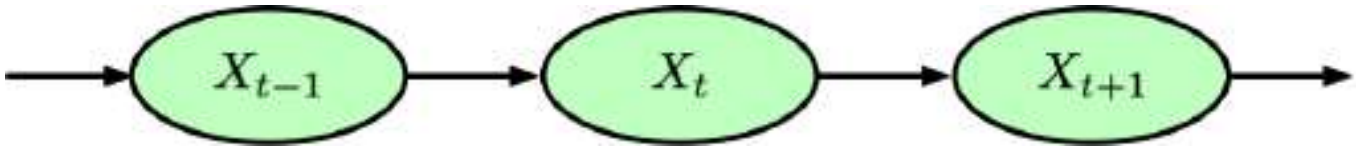
Quiz: How to do the inference?



How can we calculate $\mathbf{P}(X_{t+1} \mid x_{t-1})$? You should only use the probabilities defined in the model, like $\mathbf{P}(X_{t+1}|x_t)$ or $P(x_t|x_{t-1})$, and not $\mathbf{P}(X_t)$ or $P(x_{t+1}|x_{t-1})$.

Discuss with your neighbour for a couple of minutes.

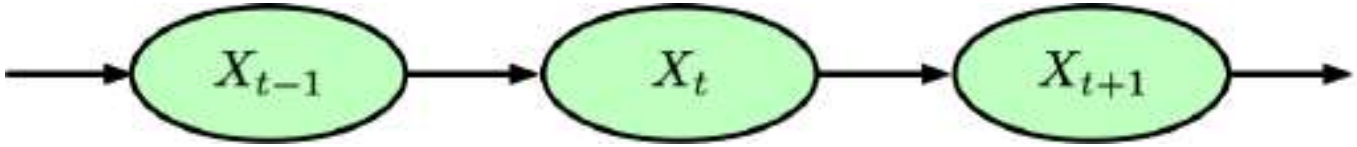
Quiz: How to do the inference?



- Everything had been easy if we had also observed X_t:

$$\mathbf{P}(X_{t+1} \mid X_t = x_t, X_{t-1} = x_{t-1}) = \mathbf{P}(X_{t+1} \mid X_t = x_t)$$

Quiz: How to do the inference?



- Everything had been easy if we had also observed X_t:

$$\mathbf{P}(X_{t+1} \mid X_t = x_t, X_{t-1} = x_{t-1}) = \mathbf{P}(X_{t+1} \mid X_t = x_t)$$

- Use this idea, but sum over the uncertainty we have about X_t:

$$\mathbf{P}(X_{t+1} \mid X_{t-1} = x_{t-1}) = \sum_{x_t} \mathbf{P}(X_{t+1}, X_t = x_t \mid X_{t-1} = x_{t-1})$$

$$= \sum_{x_t} \mathbf{P}(X_{t+1} \mid X_t = x_t, X_{t-1} = x_{t-1}) \cdot P(X_t = x_t \mid X_{t-1} = x_{t-1})$$

$$= \sum_{x_t} \mathbf{P}(X_{t+1} \mid X_t = x_t) \cdot P(X_t = x_t \mid X_{t-1} = x_{t-1})$$

Filtering

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

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = \text{Some-Func}(\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t}), \mathbf{e}_{t+1})$$

$$\begin{aligned}
\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \mathbf{P}(\mathbf{X}_{t+1}, \mathbf{e}_{1:t}, \mathbf{e}_{t+1})/P(\mathbf{e}_{1:t+1}) \\
&= \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}, \mathbf{e}_{1:t}) \cdot \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \cdot P(\mathbf{e}_{1:t})/P(\mathbf{e}_{1:t+1}) \\
&= \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \cdot \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \cdot \alpha \quad (\text{as } \boldsymbol{E}_{t+1} \perp\!\!\!\perp \boldsymbol{E}_{1:t}|\mathbf{X}_{t+1}) \\
&= \alpha \cdot \underbrace{\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})}_{\text{Evidence}} \cdot \underbrace{\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t})}_{\text{Prediction}}
\end{aligned}$$

So, filtering is a **prediction updated by evidence**.

Filtering

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

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = \text{Some-Func}(\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t}), \mathbf{e}_{t+1})$$

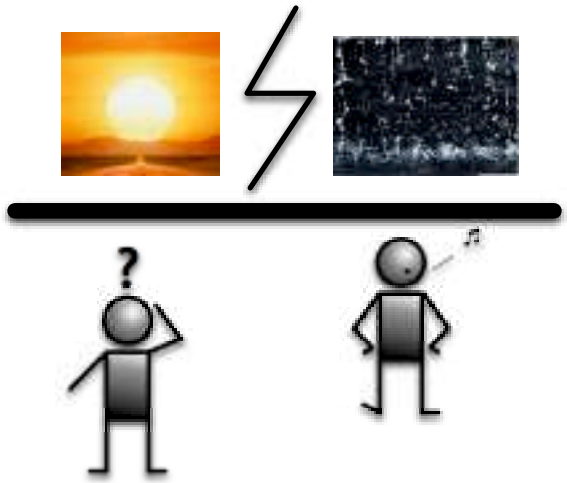
Prediction by summing out $\mathbf{X}_t$:

$$\begin{aligned}
&\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) \\
&= \alpha \cdot \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \cdot \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \\
&= \alpha \cdot \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \cdot \{\Sigma_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t, \mathbf{e}_{1:t}) \cdot P(\mathbf{x}_t|\mathbf{e}_{1:t})\} \\
&= \alpha \cdot \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \cdot \underbrace{\{\Sigma_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t \quad) \cdot P(\mathbf{x}_t|\mathbf{e}_{1:t})\}}_{\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \text{ using what we have already}}
\end{aligned}$$

All relevant information contained in $\mathbf{f}_{1:t} = \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$; belief revision using $\mathbf{f}_{1:t+1} = \textsc{Forward}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$.

Note! Time and space requirements for calculating $\mathbf{f}_{1:t+1}$ is **constant** (independent of t)

Example of Hidden Markov Model from the book

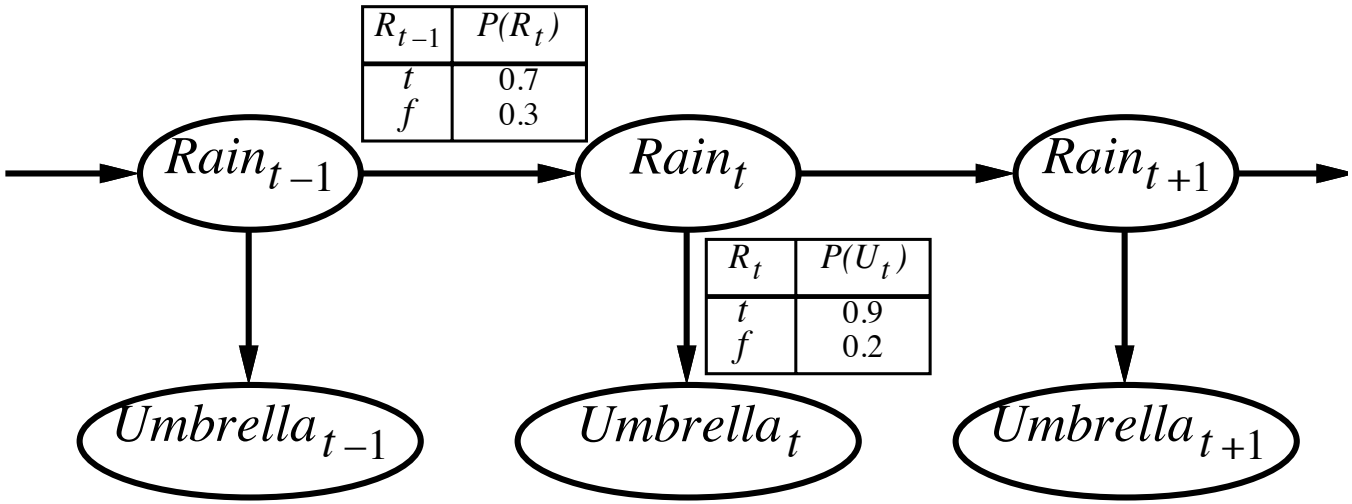


- **Problem:** Our guy sits in a bunker underground, wondering what the weather is like each day: Rain or shine?
- **Sensors:** His boss walking by is bringing an umbrella with $p = .9$ if raining and $p = .2$ if sunshine.
- **Dynamics:** Weather is the same as yesterday with $p = .7$.

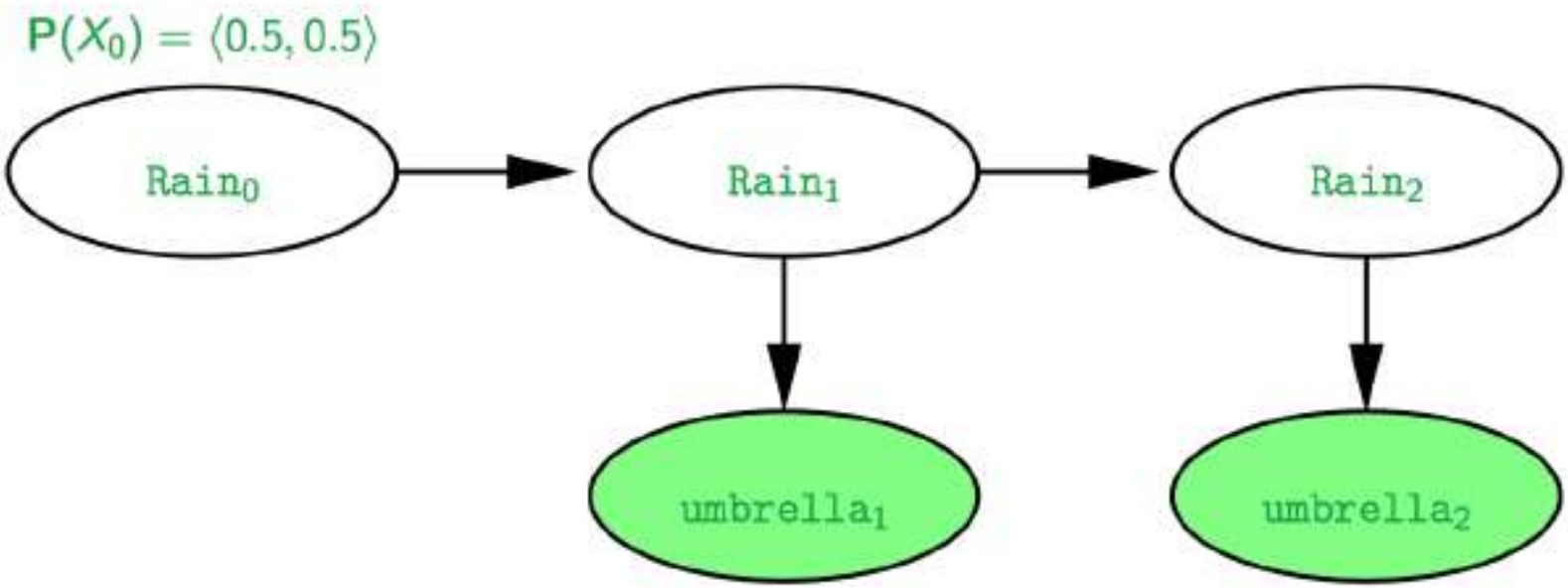
Formalize the problem as a Hidden Markov model. What inference is our guy doing each morning as he sees the boss?

Discuss with your neighbour for a couple of minutes.

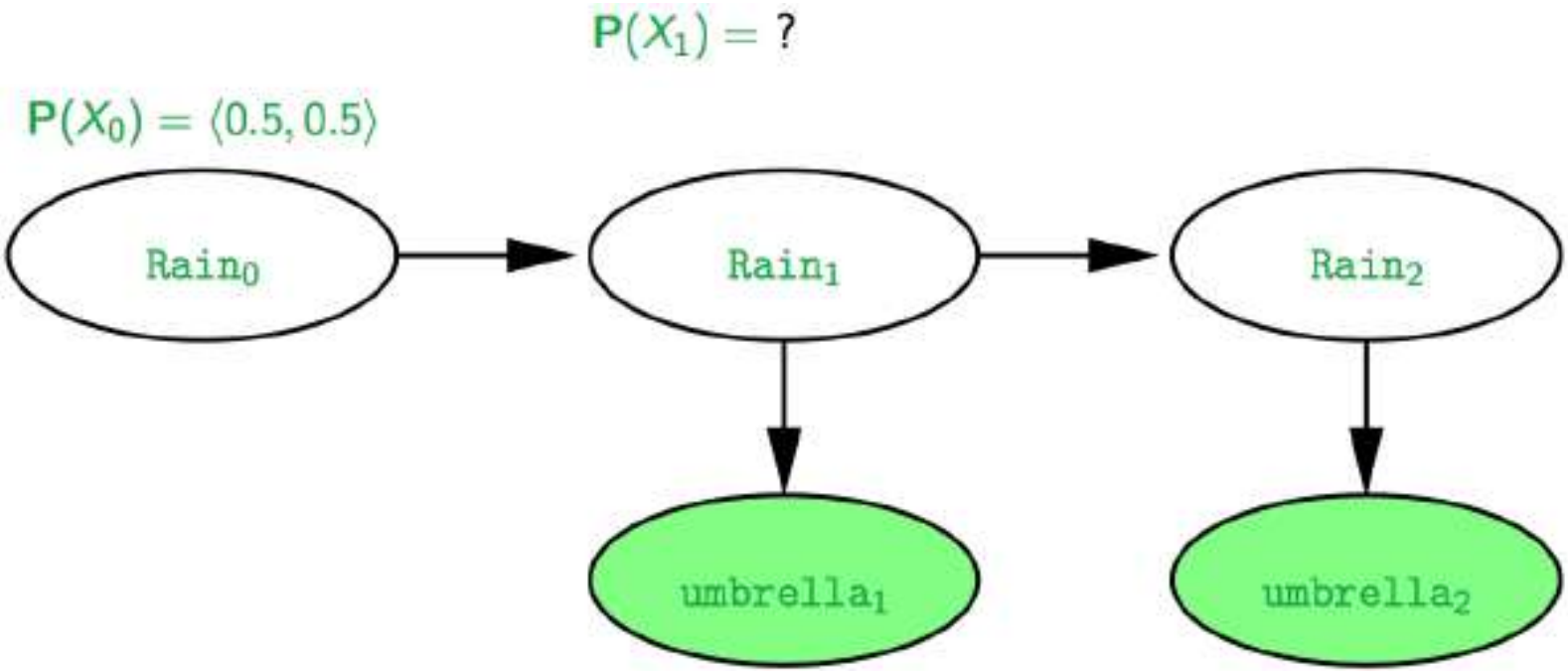
Example of Hidden Markov Model from the book



Filtering example

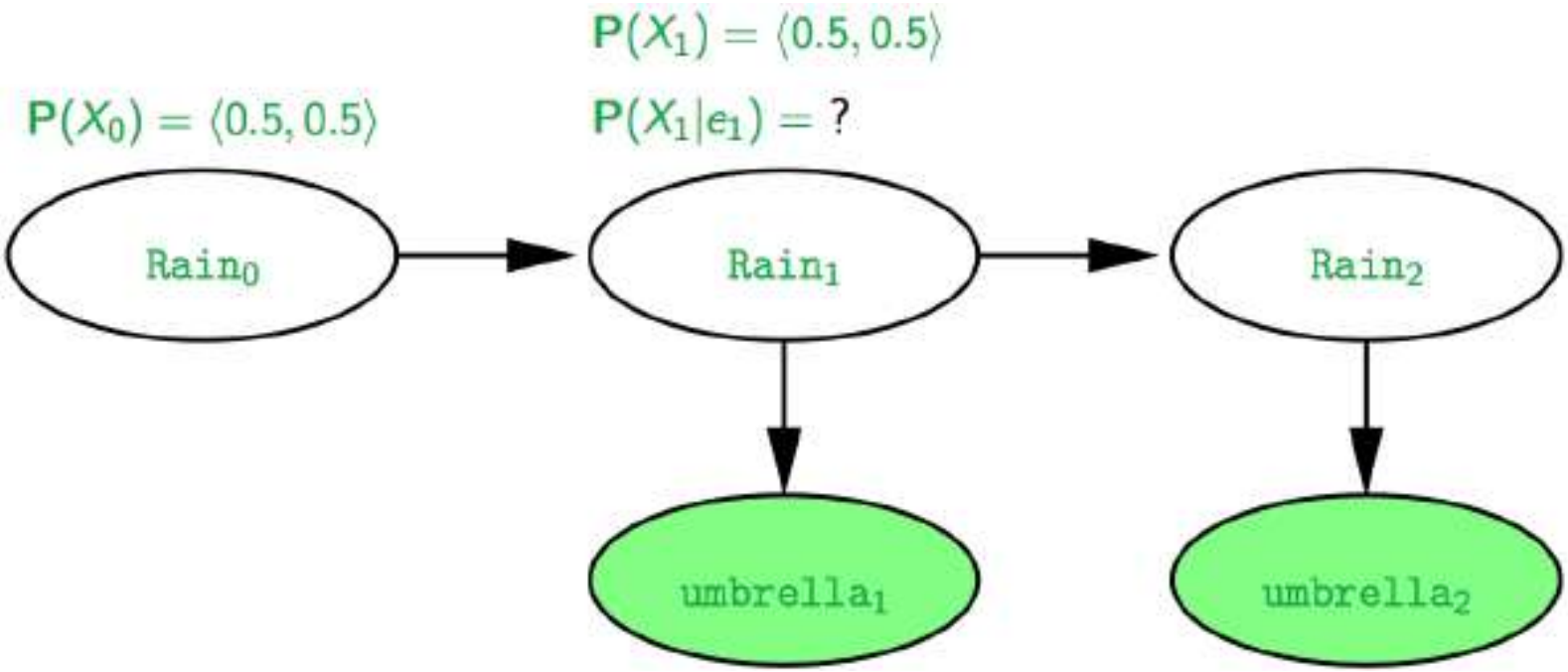


Filtering example



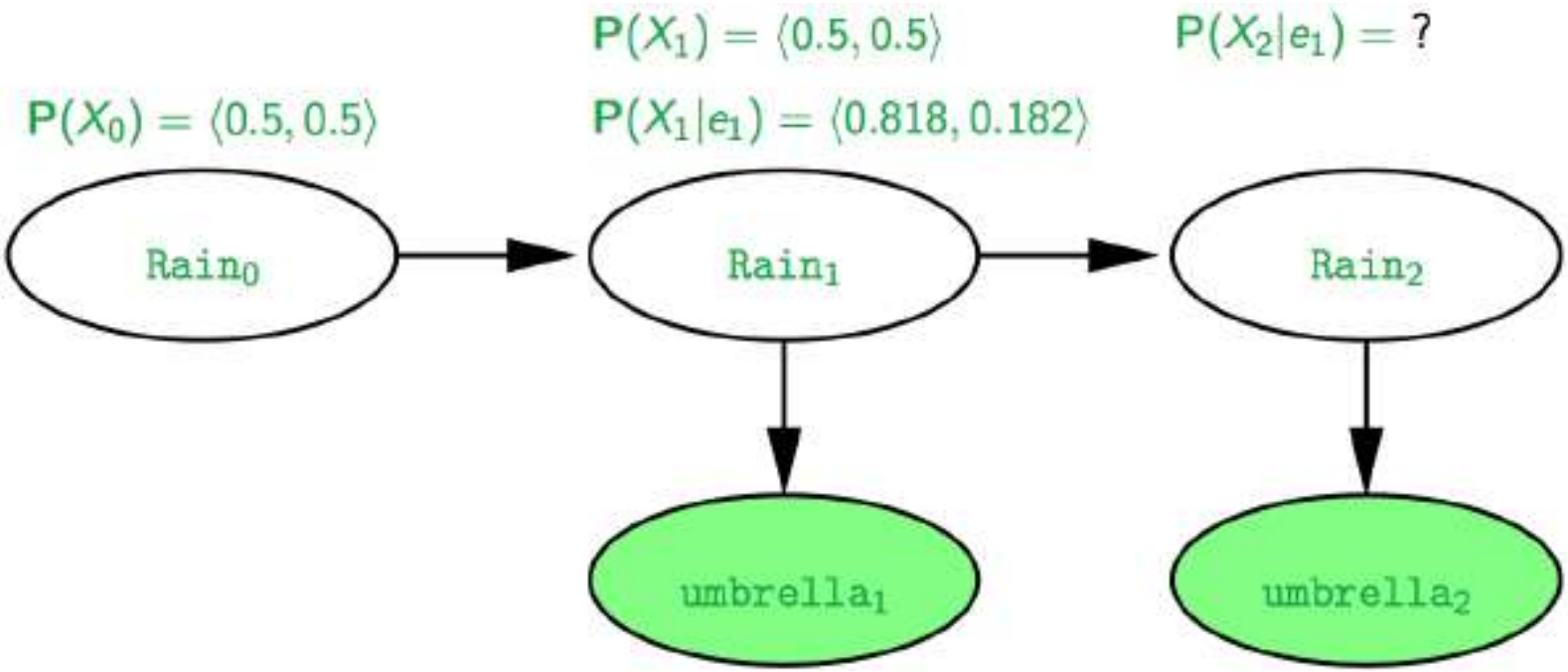
$$\begin{aligned}
\mathbf{P}(X_1) &= \sum_{x_0} \mathbf{P}(X_1|x_0) \cdot P(x_0) \\
&= \langle 0.7, 0.3 \rangle \cdot 0.5 + \langle 0.3, 0.7 \rangle \cdot 0.5 \\
&= \langle 0.5, 0.5 \rangle
\end{aligned}$$

Filtering example



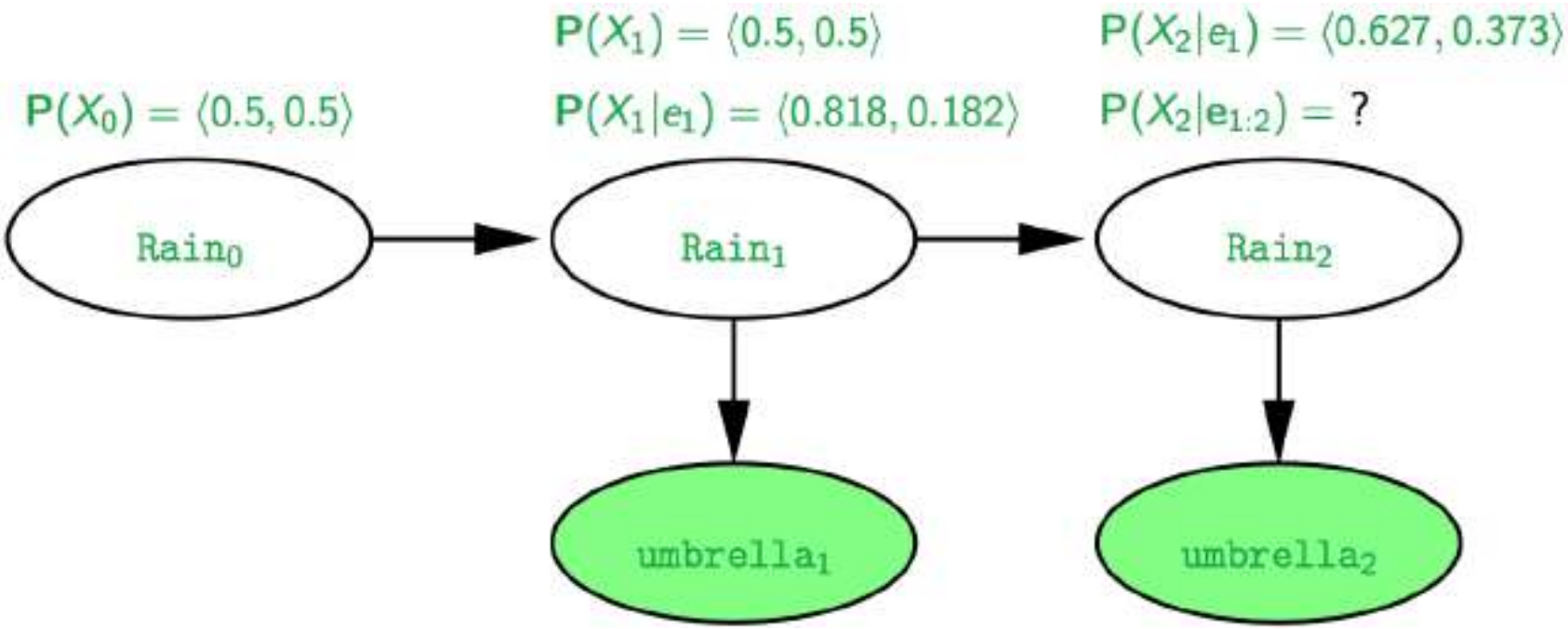
$$\begin{aligned}\mathbf{P}(X_1|e_1) &= \alpha \cdot \mathbf{P}(e_1|X_1)\mathbf{P}(X_1) \\ &= \alpha \cdot \langle 0.9 \cdot 0.5, 0.2 \cdot 0.5 \rangle \\ &= \langle 0.818, 0.182 \rangle\end{aligned}$$

Filtering example



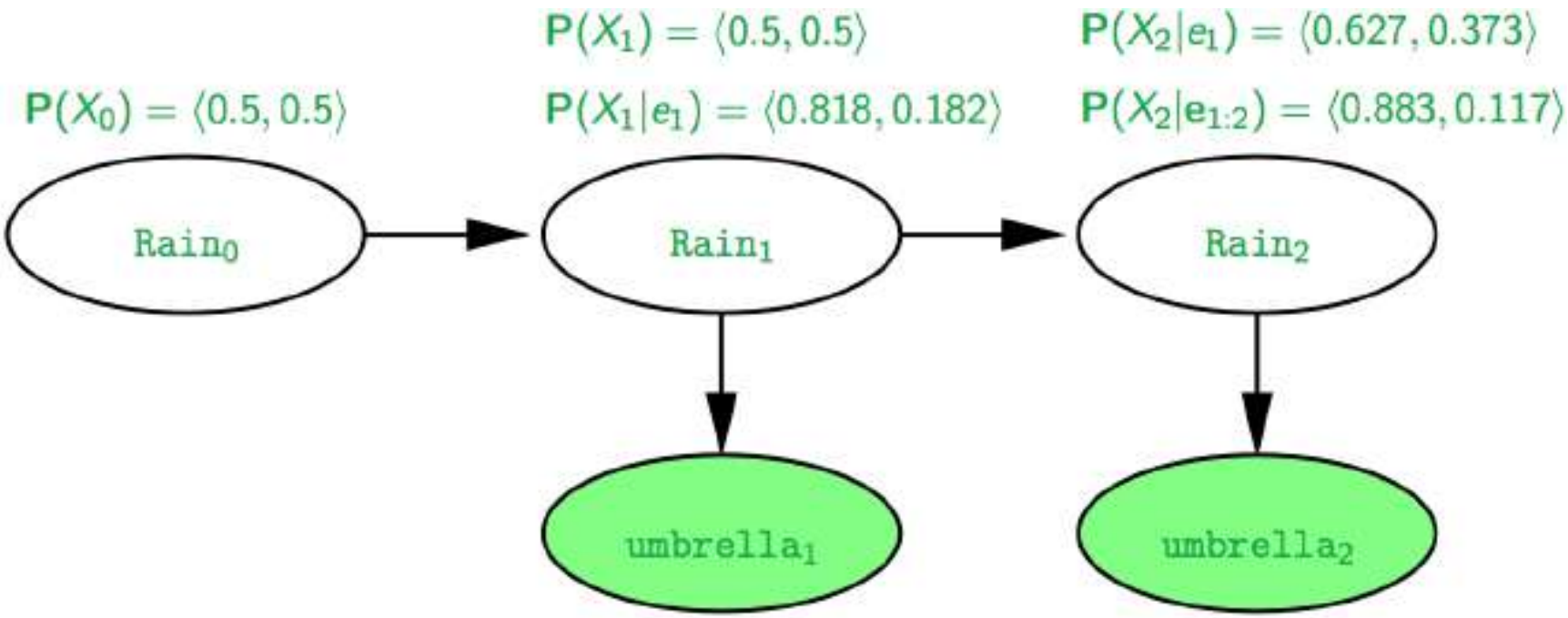
$$
\begin{aligned}
\mathbf{P}(X_2|e_1) &= \sum_{x_1} \mathbf{P}(X_2|x_1) \cdot P(x_1|e_1) \\
&= \langle 0.7, 0.3 \rangle \cdot 0.818 + \langle 0.3, 0.7 \rangle \cdot 0.182 \\
&= \langle 0.627, 0.373 \rangle
\end{aligned}
$$

Filtering example



$$
\begin{aligned}
\mathbf{P}(X_2|\mathbf{e}_{1:2}) &= \alpha \cdot \mathbf{P}(e_2|X_2) \cdot \mathbf{P}(X_2 \mid e_1) \\
&= \alpha \cdot \langle 0.9, 0.2 \rangle \cdot \langle 0.627, 0.373 \rangle \\
&= \alpha \cdot \langle 0.565, 0.075 \rangle \\
&= \langle 0.883, 0.117 \rangle
\end{aligned}
$$

Filtering example



Demo: GeNIe – `rainPlate.xdsl` vs. `rain.xdsl`

Simplifications for Hidden Markov models

$\mathbf{X}_t$ is a single, discrete variable (as is $\mathbf{E}_t$ usually, too)
Domain of X_t is $\{1, \ldots, S\}$

Transition matrix $\mathbf{T}_{ij} = P(X_t = j | X_{t-1} = i)$, e.g., $\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$

Sensor matrix $\mathbf{O}_t$ for each t, diagonal elements $P(e_t | X_t = i)$.
For instance, with $U_1 = \texttt{true}$ we get

$$\mathbf{O}_1 = \begin{pmatrix} P(u_1|x_1) & 0 \\ 0 & P(u_1|\neg x_1) \end{pmatrix} = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$$

Define $\mathbf{f}_{1:t} = \mathbf{P}(\mathbf{X}_t | \mathbf{e}_{1:t})$. These are called **forward messages**.

Now, forward messages can be calculated by simple matrix operations (using $\mathbf{f}_{1:0} = \mathbf{P}(\mathbf{X}_0)$):

$$\mathbf{f}_{1:t} = \alpha \mathbf{O}_t \mathbf{T}^\top \mathbf{f}_{1:t-1}$$

Prediction

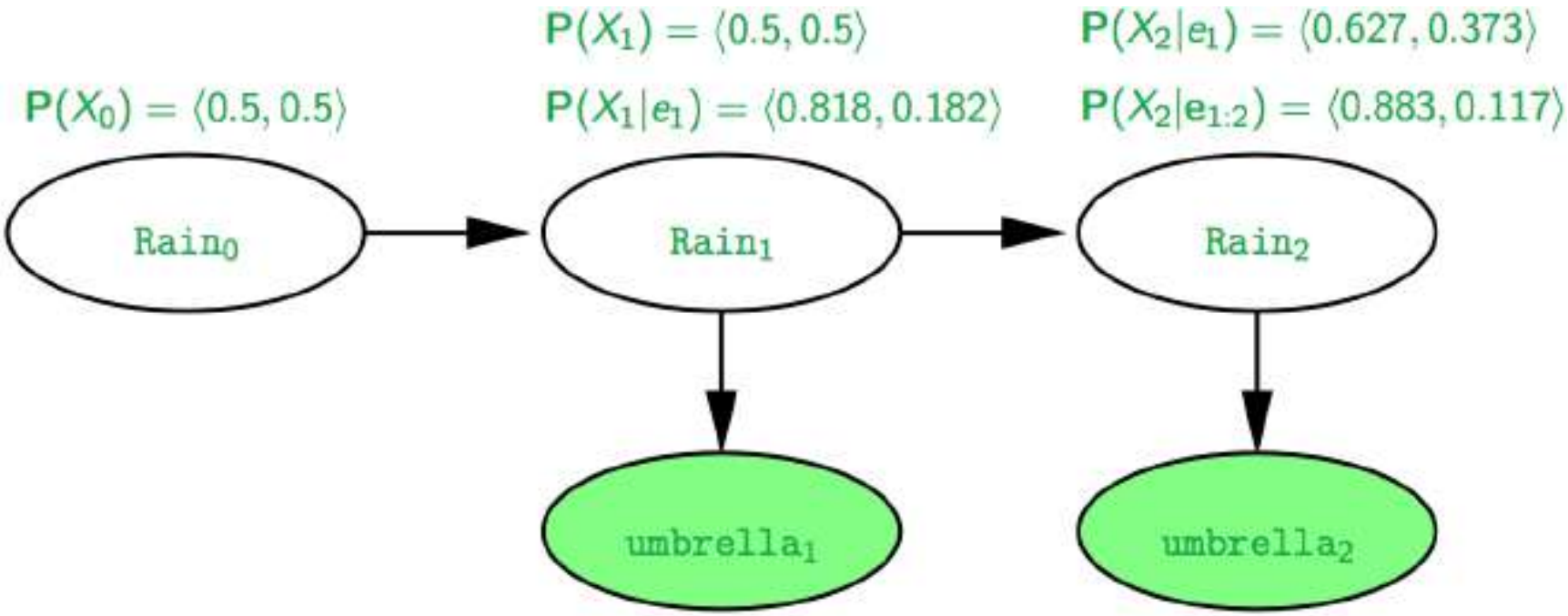
$$\mathbf{P}(\mathbf{X}_{t+k+1}|\mathbf{e}_{1:t}) = \Sigma_{\mathbf{x}_{t+k}} \mathbf{P}(\mathbf{X}_{t+k+1}|\mathbf{x}_{t+k}) P(\mathbf{x}_{t+k}|\mathbf{e}_{1:t})$$

Again we have a recursive formulation – This time over k...

As $k \to \infty$, $P(\mathbf{x}_{t+k}|\mathbf{e}_{1:t})$ tends to the **stationary distribution** of the Markov chain. This means that the effect of $\mathbf{e}_{1:t}$ will vanish as k increases, and predictions will become more and more dubious.

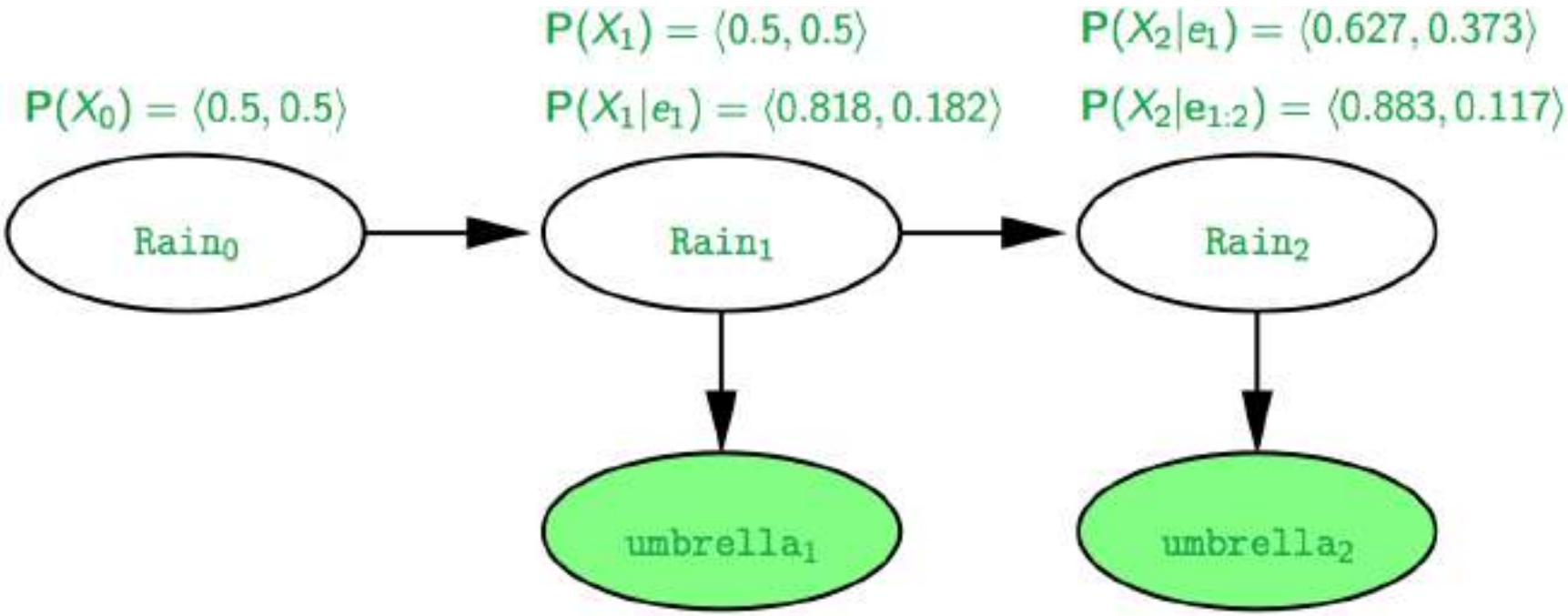
Mixing time depends on how **stochastic** the chain is ("how persistent $\mathbf{X}$ is")

Prediction – Example



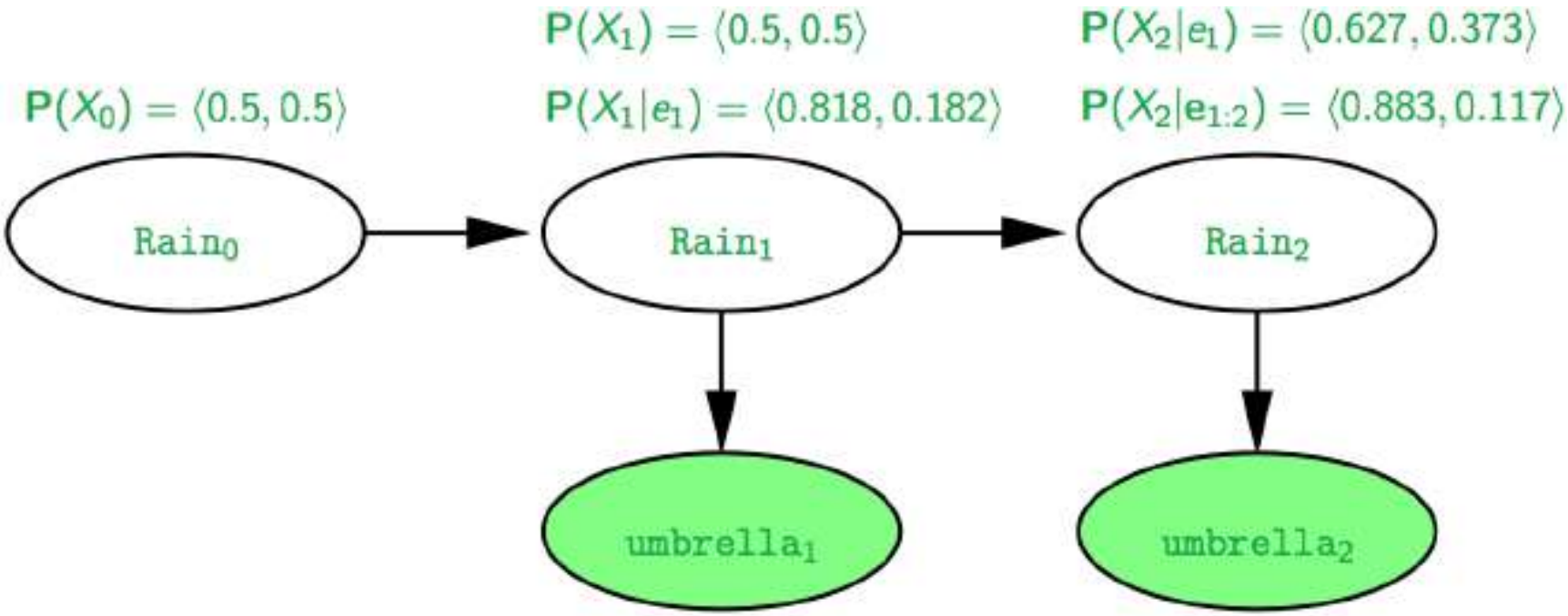
$$\begin{aligned}
\mathbf{P}(X_3|\mathbf{e}_{1:2}) &= \sum_{x_2} \mathbf{P}(X_3|x_2) \cdot P(x_2|\mathbf{e}_{1:2}) \\
&= \langle 0.7, 0.3 \rangle \cdot 0.883 + \langle 0.3, 0.7 \rangle \cdot 0.117 \\
&= \langle 0.653, 0.347 \rangle
\end{aligned}$$

Prediction – Example



$$\begin{aligned}
\mathbf{P}(X_4|\mathbf{e}_{1:2}) &= \sum_{x_3} \mathbf{P}(X_4|x_3) \cdot P(x_3|\mathbf{e}_{1:2}) \\
&= \langle 0.7, 0.3 \rangle \cdot 0.653 + \langle 0.3, 0.7 \rangle \cdot 0.347 \\
&= \langle 0.561, 0.439 \rangle
\end{aligned}$$

Prediction – Example

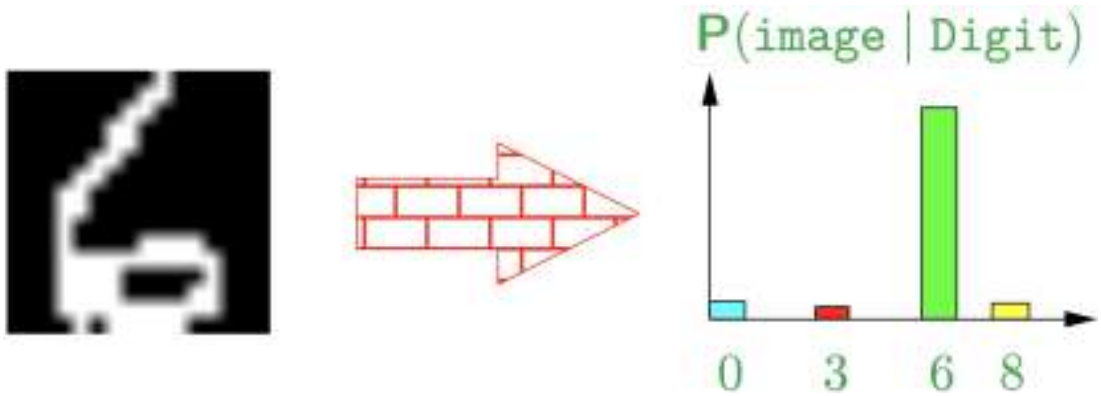


$$\begin{aligned}\mathbf{P}(X_{10}|\mathbf{e}_{1:2}) &= \sum_{x_9} \mathbf{P}(X_{10}|x_9) \cdot P(x_9|\mathbf{e}_{1:2}) \\ &= \langle 0.7, 0.3 \rangle \cdot 0.501 + \langle 0.3, 0.7 \rangle \cdot 0.499 \\ &= \langle 0.500, 0.500 \rangle\end{aligned}$$

$\lim_{k\to\infty} \mathbf{P}(X_{t+k}|\mathbf{e}_{1:t}) = \langle \frac{1}{2}, \frac{1}{2} \rangle$ for this transition model.

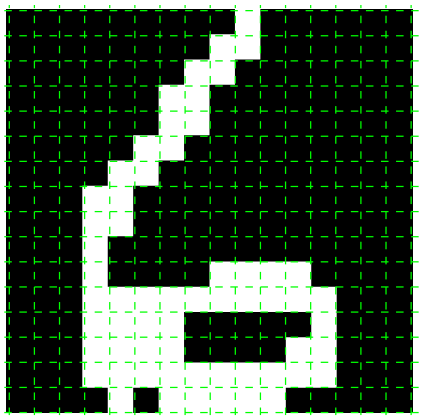
Example: Automatic recognition of hand-written digits

We have this system that can "recognise" hand-written digits:



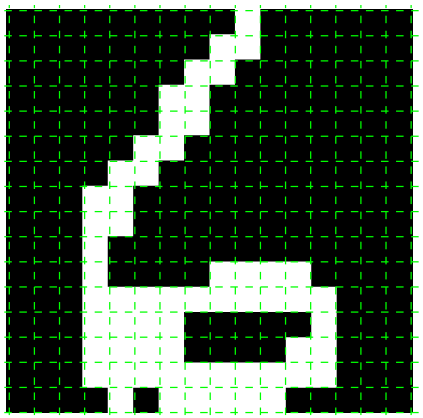
- Takes a binary image of a handwritten digit as input
- Returns **P**(image | Digit).
  - We are really interested in **P**(Digit | image), but will get that using Bayes' rule.
- (The system we will consider is not very good)

Internals of recogniser – Naïve Bayes



- An image is a 16×16 matrix of binary variables $\texttt{Image}_{i,j}$: $\texttt{Image}_{i,j} = \texttt{true}$ if pixel (i, j) is white, $\texttt{false}$ otherwise.
- **How should we proceed?** We need a model for $\mathbf{P}(\textbf{image} \mid \texttt{Digit})$. Note that **image** is 256-dimensional.
- **Idea:** The different digits distribute white spots differently in the image $\Rightarrow$ combine single-pixel information to find digit.

Internals of recogniser – Naïve Bayes



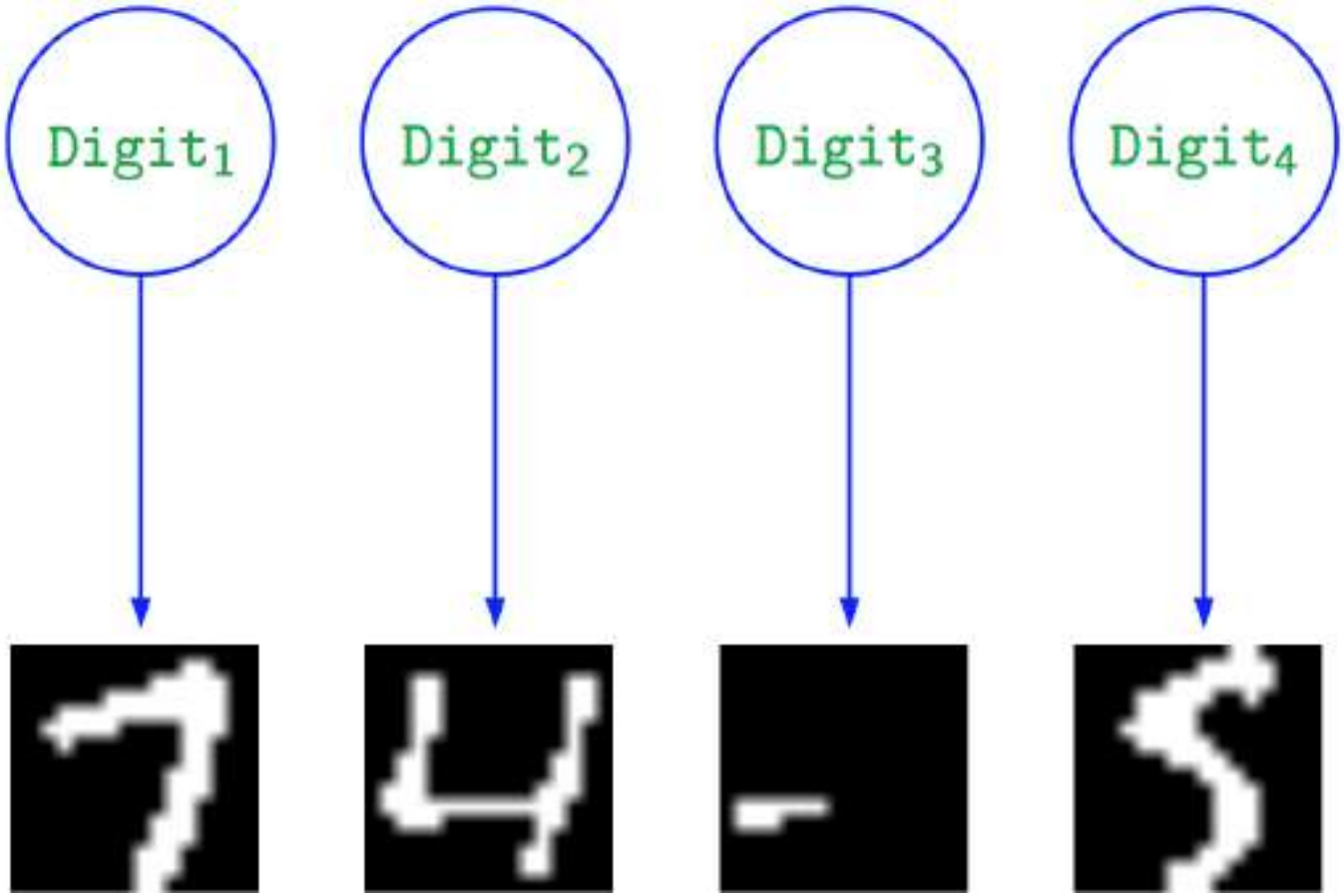
- An image is a 16×16 matrix of binary variables $\texttt{Image}_{i,j}$: $\texttt{Image}_{i,j} = \texttt{true}$ if pixel (i, j) is white, $\texttt{false}$ otherwise.
- **In this example** we assume that each location contribute independently (Naïve Bayes model):

$$\mathbf{P}(\mathbf{image} \mid \texttt{Digit}) = \prod_i \prod_j P(\texttt{image}_{i,j} \mid \texttt{Digit}).$$

- Bayes rule gives us the classification:
$\mathbf{P}(\texttt{Digit} \mid \mathbf{image}) = \alpha \cdot \mathbf{P}(\mathbf{image} \mid \texttt{Digit}) \cdot \mathbf{P}(\texttt{Digit})$.

Scaling up: ZIP-codes

We want to build a system that can decode hand-written ZIP-codes for letters to Norway.



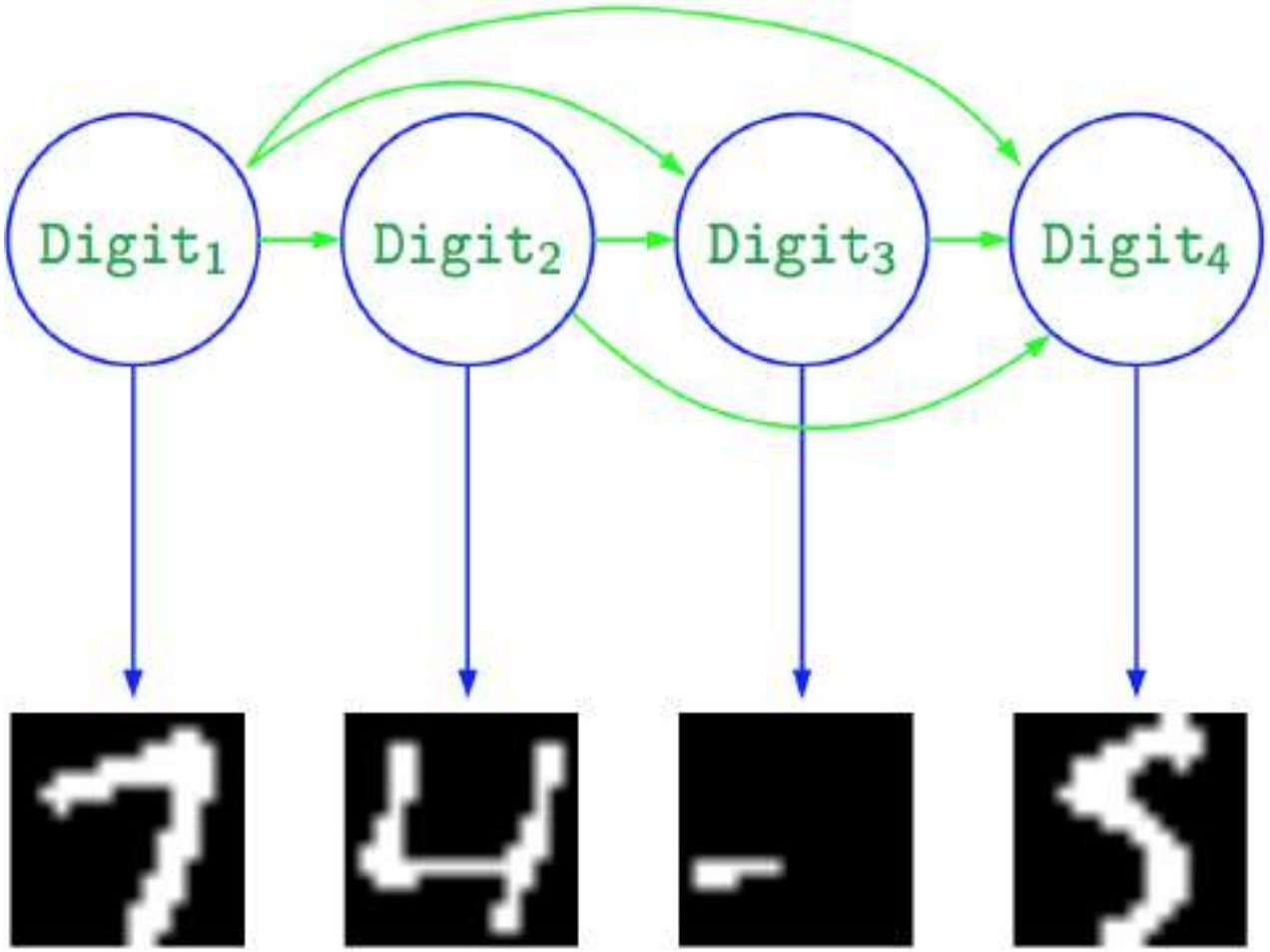
Scaling up: ZIP-codes

We want to build a system that can decode hand-written ZIP-codes for letters to Norway.

- There is a **structure** in this:
  - ZIP-codes always have 4 digits
  - Some ZIP-codes more frequent than others (e.g., `0xxx` – `13xx` for Oslo, `50xx` for Bergen, `70xx` for Trondheim)
  - Some ZIP-codes are not used, e.g. 5022 does not exist
  - . . . but some illegal numbers are often used, e.g. 7000 meaning "Wherever in Trondheim"
- Can we utilise the internal structure to improve the digits-recogniser?

How to model the internal structure of ZIP-codes

Take 1: Full model



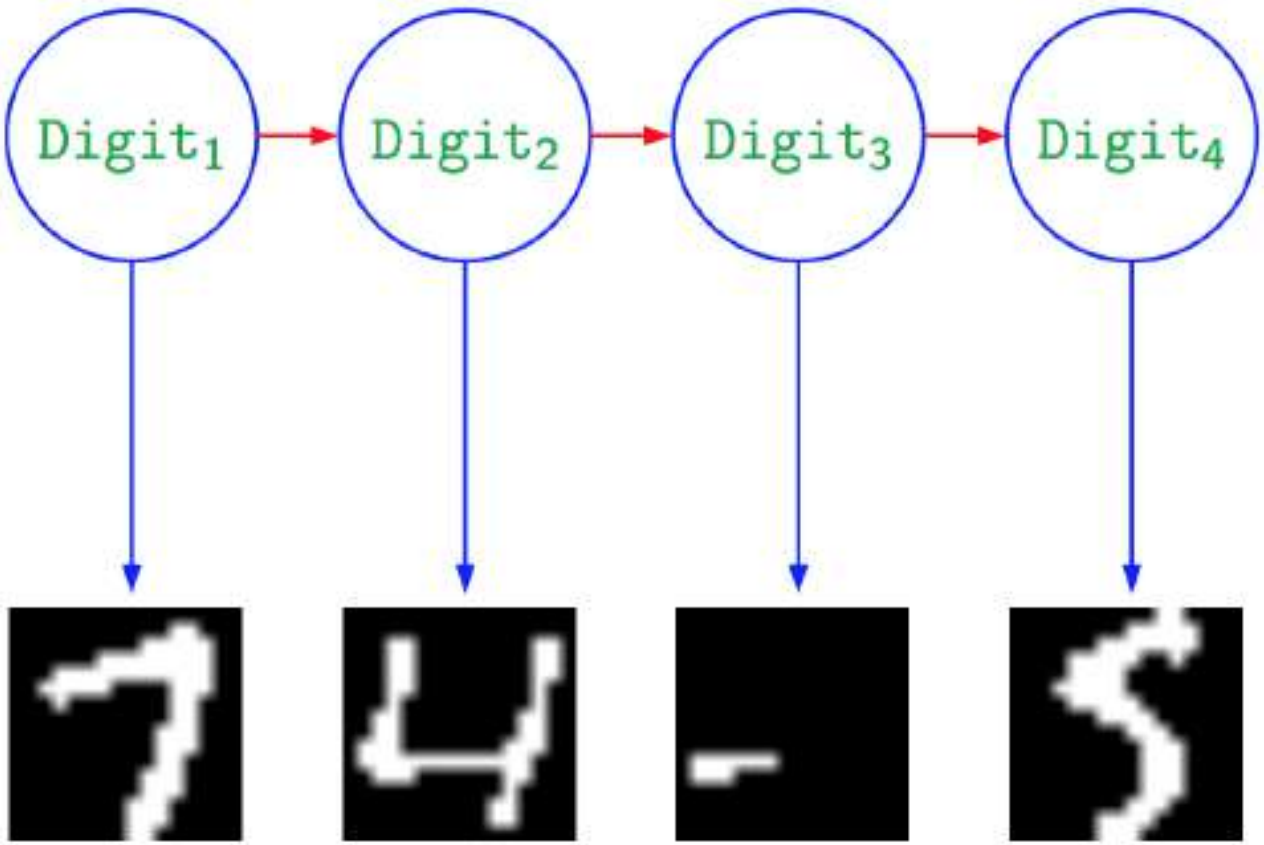
How to model the internal structure of ZIP-codes

Take 1: Full model

- The **full model** includes all relations between digits:
  - 7465 is commonly used, 7365 is not
- The problem is related to size of CPTs:
  - How many numbers to represent $\mathbf{P}(\mathtt{Digit}_4 \mid \mathtt{Pa}(\mathtt{Digit}_4))$?
  - What if we want to use this system to recognise KID numbers (often more than ten digits)?

How to model the internal structure of ZIP-codes

Take 2: Markov model



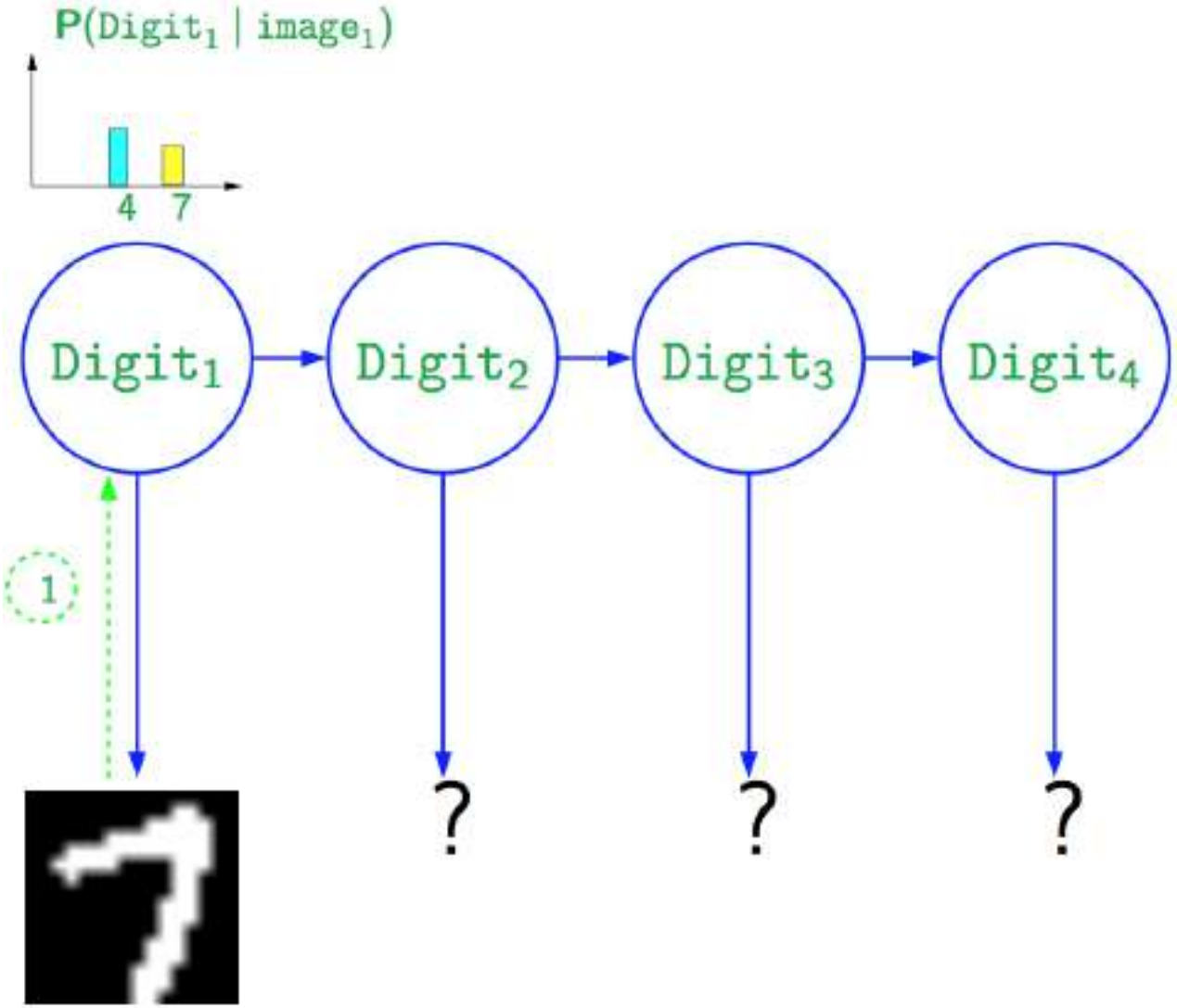
How to model the internal structure of ZIP-codes

Take 2: Markov model

- The **reduced model** includes only some relations between digits:
  - Can represent "If start with 7 and digit number three is 6, then the second one is probably 4"
  - Cannot represent "If start with 9 then digit number four is probably not 7"
- What about making the model **stationary**?
  - Does not seem appropriate here.
  - Might be necessary and/or reasonable for KID, though.

Inference (filtering)

Step 1: First digit classified as a 4! (Not good! I told you!)



Inference (filtering)

Step 1: First digit classified as a 4! (Not good! I told you!)

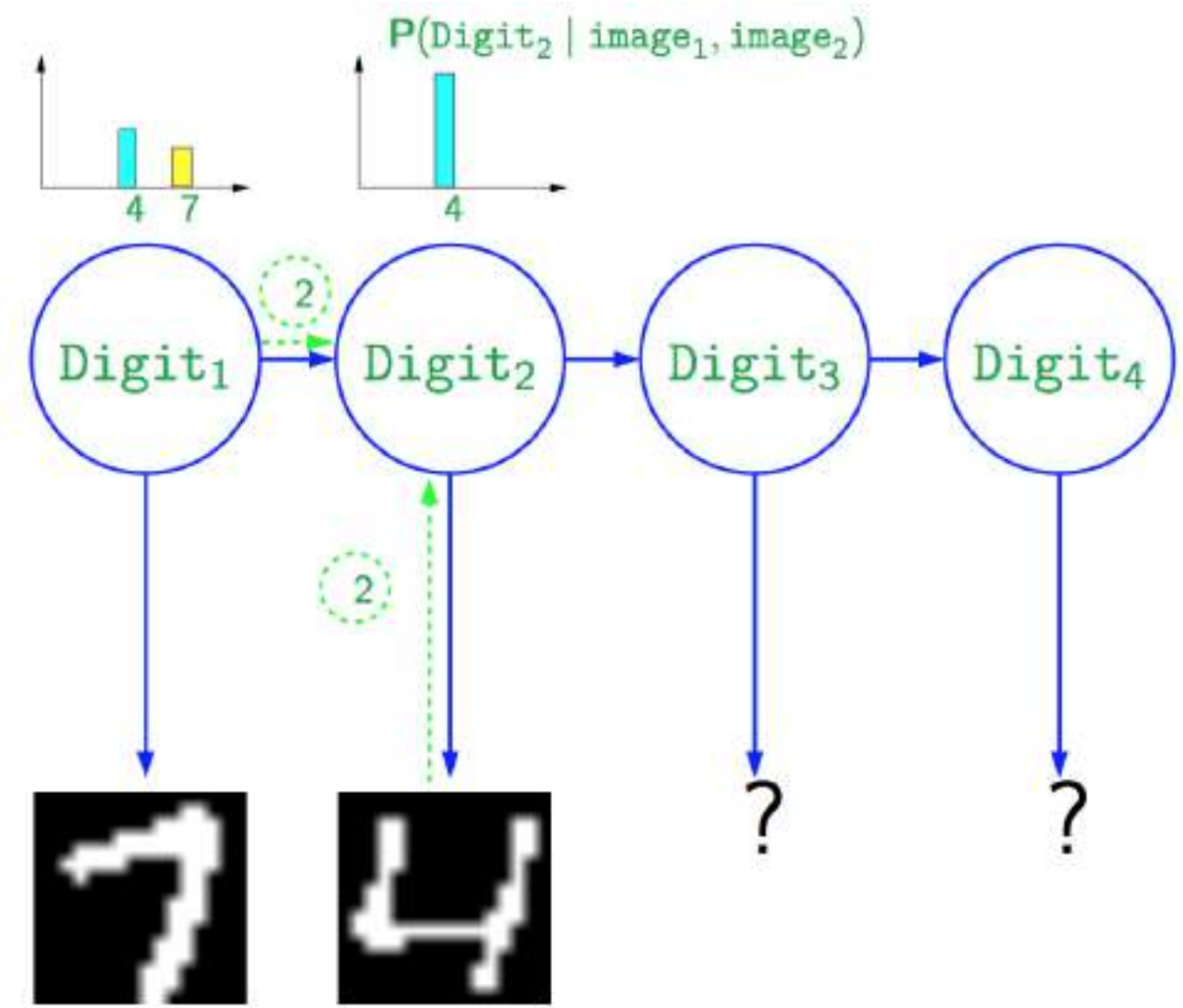
So what happened?

- The Naive Bayes method supplies $\mathbf{P}(\mathtt{image}_1 \mid \mathtt{Digit}_1)$
- Using the calculation rule, the system finds

$$\mathbf{P}(\mathtt{Digit}_1 \mid \mathtt{image}_1) = \alpha \cdot \mathbf{P}(\mathtt{image}_1 \mid \mathtt{Digit}_1) \cdot \mathbf{P}(\mathtt{Digit}_1)$$

Inference (filtering)

Step 2: Second digit classified as a 4.



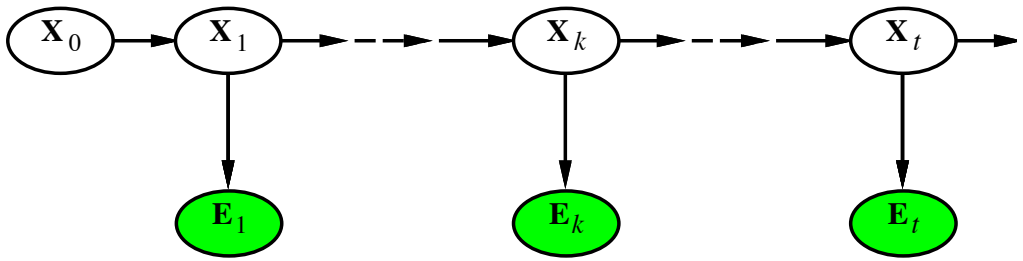
Inference (filtering)

Step 2: Second digit classified as a 4.

So what happened?

- The Naive Bayes method supplies $\mathbf{P}(\mathtt{image}_2 \mid \mathtt{Digit}_2)$
- Using the calculation rule, the system finds
  $\mathbf{P}(\mathtt{Digit}_2 \mid \mathtt{image}_1, \mathtt{image}_2) =$
  $\alpha \cdot \mathbf{P}(\mathtt{image}_2 \mid \mathtt{Digit}_2) \cdot$
  $\sum_{\mathtt{digit}_1} \mathbf{P}(\mathtt{Digit}_2 \mid \mathtt{digit}_1) P(\mathtt{digit}_1 \mid \mathtt{image}_1)$
- To do the classification, the system used the information that
  - The image is a very typical "4"
  - $7 \rightarrow 4$ is probable
  - $4 \rightarrow 4$ is not very probable, but possible
- **Can this structural information also be used "backwards"?**
  - If the 2nd digit looks like 4, then 1st digit is probably a 7, not 4
  - This is called **smoothing**

Smoothing

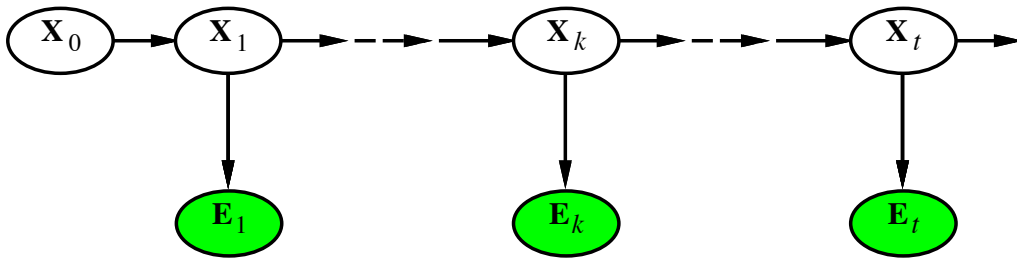


Calculate $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ by dividing evidence $\mathbf{e}_{1:t}$ into $\mathbf{e}_{1:k}$, $\mathbf{e}_{k+1:t}$:

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

where $\mathbf{b}_{k+1:t} = \mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k)$.

Smoothing



Backward message computed by a backwards recursion:

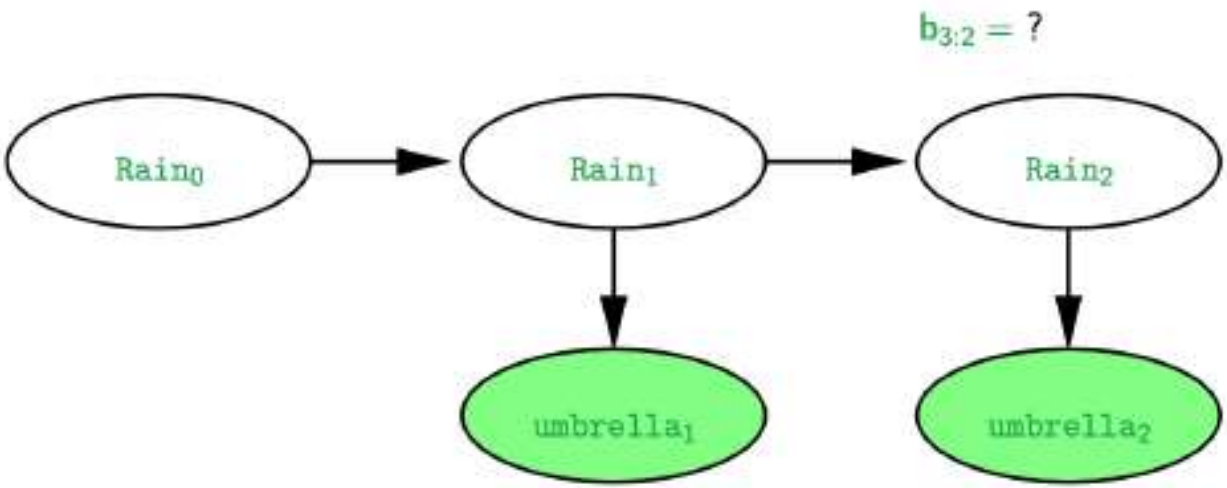
$$\begin{aligned}\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) &= \Sigma_{\mathbf{x}_{k+1}}\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k,\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)\\ &= \Sigma_{\mathbf{x}_{k+1}}P(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)\\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

So...

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

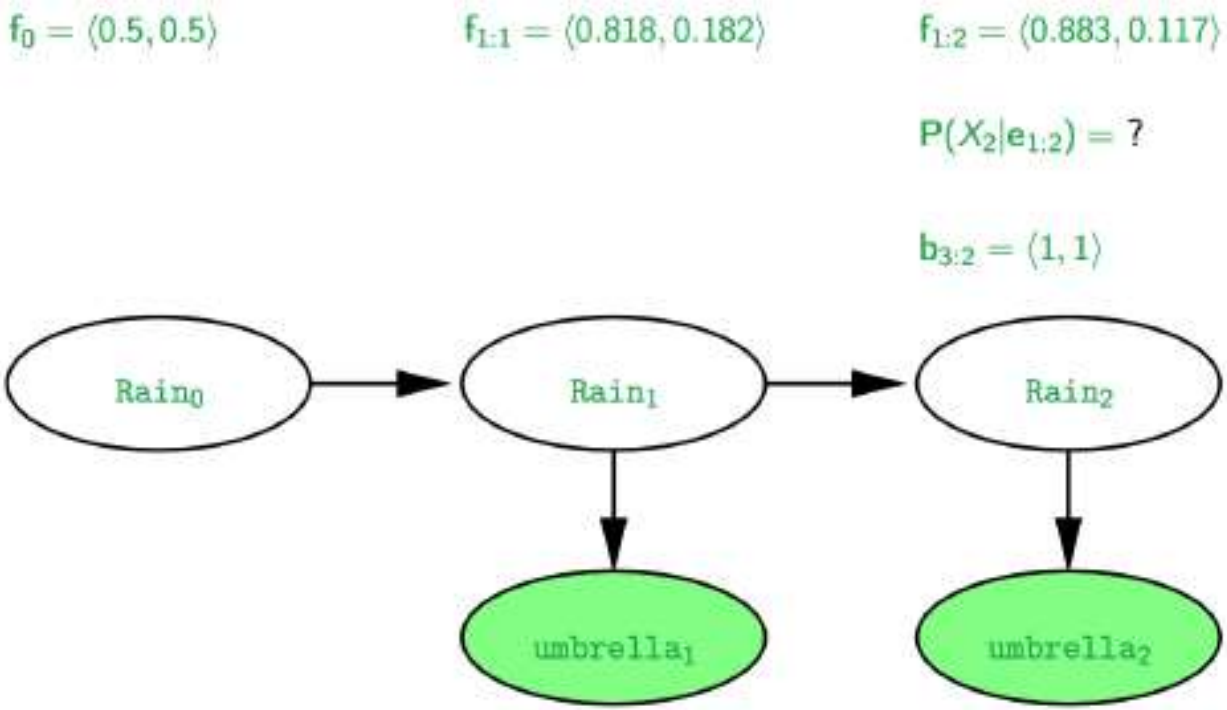
Smoothing example

$\mathbf{f}_0 = \langle 0.5, 0.5 \rangle$ $\mathbf{f}_{1:1} = \langle 0.818, 0.182 \rangle$ $\mathbf{f}_{1:2} = \langle 0.883, 0.117 \rangle$



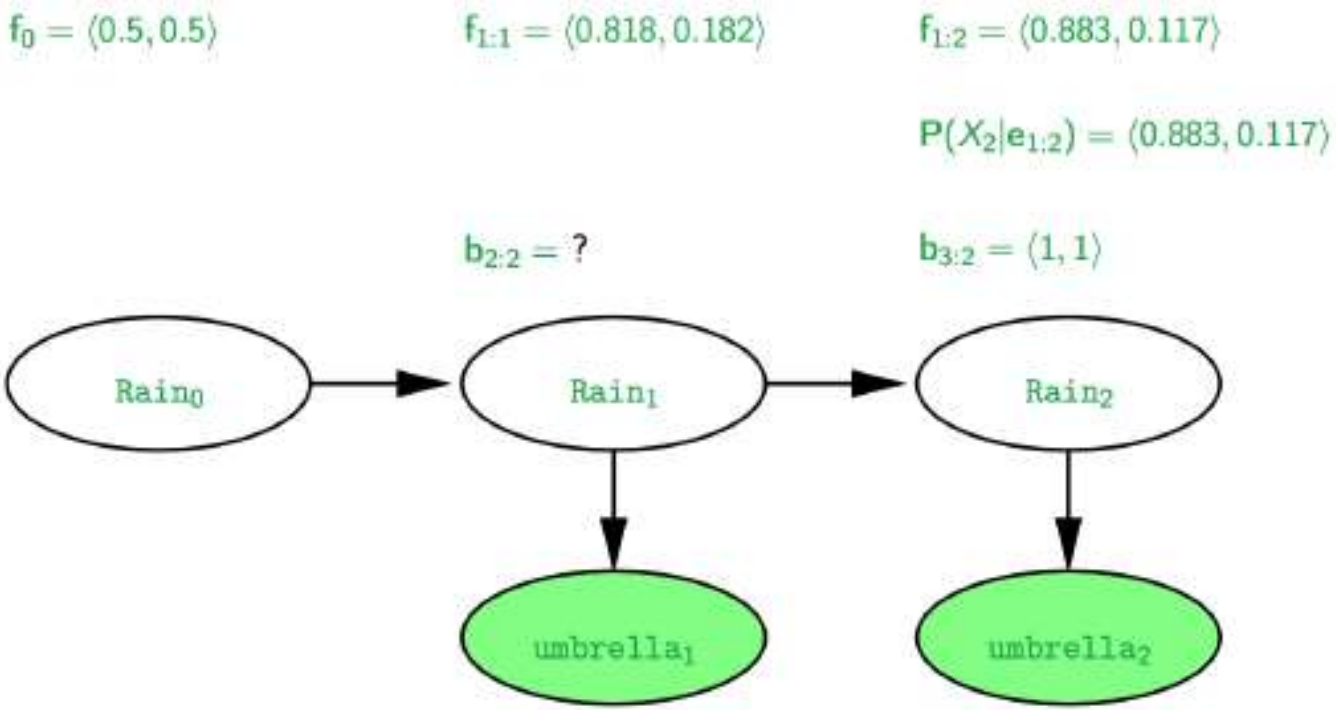
$$\mathbf{b}_{3:2} = \mathbf{P}(\mathbf{e}_{3:2}|\mathbf{X}_2)$$
$$= \langle 1, 1 \rangle \quad \text{(void)}$$

Smoothing example



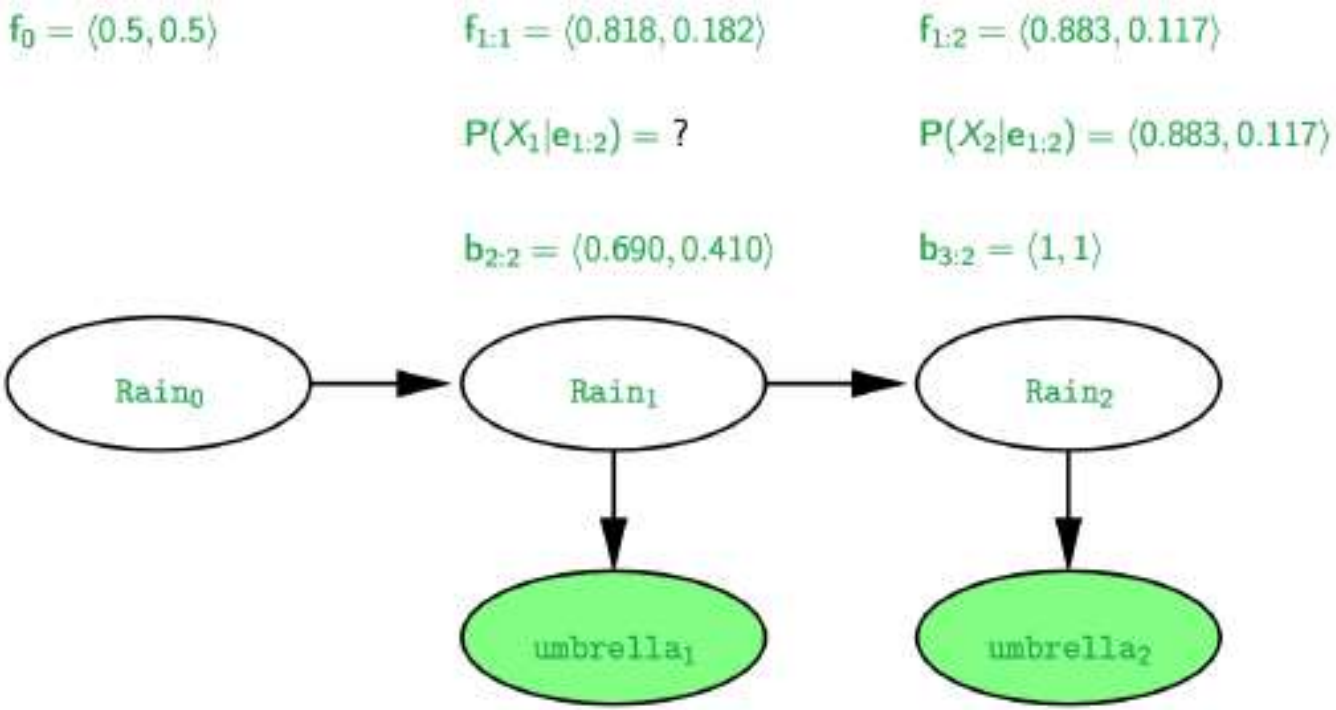
$$\begin{aligned}\mathbf{P}(X_2|\mathbf{e}_{1:2}) &= \alpha \cdot \mathbf{f}_{1:2} \cdot \mathbf{b}_{3:2} \\ &= \alpha \cdot \langle 0.883, 0.117 \rangle \cdot \langle 1, 1 \rangle \\ &= \langle 0.883, 0.117 \rangle\end{aligned}$$

Smoothing example



$$
\begin{aligned}
\mathbf{b}_{2:2} &= \mathbf{P}(\mathbf{e}_{2:2}|\mathbf{X}_1) \\
&= \sum_{\mathbf{x}_2} P(\mathbf{e}_2|\mathbf{x}_2) \cdot \mathbf{b}_{3:2}(x_2) \cdot \mathbf{P}(x_2|\mathbf{X}_1) \\
&= (0.9 \cdot 1 \cdot \langle 0.7, 0.3 \rangle) + (0.2 \cdot 1 \cdot \langle 0.3, 0.7 \rangle) = \langle 0.690, 0.410 \rangle
\end{aligned}
$$

Smoothing example



$$\begin{aligned}
\mathbf{P}(X_1|\mathbf{e}_{1:2}) &= \alpha \mathbf{f}_{1:1} \cdot \mathbf{b}_{2:2} \\
&= \alpha \cdot \langle 0.818, 0.182 \rangle \cdot \langle 0.690, 0.410 \rangle \\
&= \langle 0.883, 0.117 \rangle
\end{aligned}$$

Smoothing example

$\mathbf{f}_0 = \langle 0.5, 0.5 \rangle$

$\mathbf{f}_{1:1} = \langle 0.818, 0.182 \rangle$

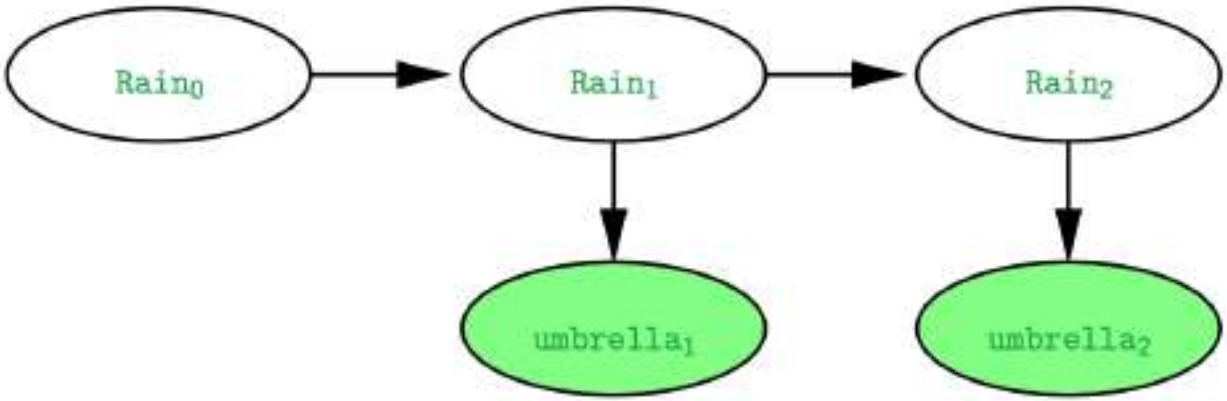
$\mathbf{f}_{1:2} = \langle 0.883, 0.117 \rangle$

$\mathbf{P}(X_1|\mathbf{e}_{1:2}) = \langle 0.883, 0.117 \rangle$

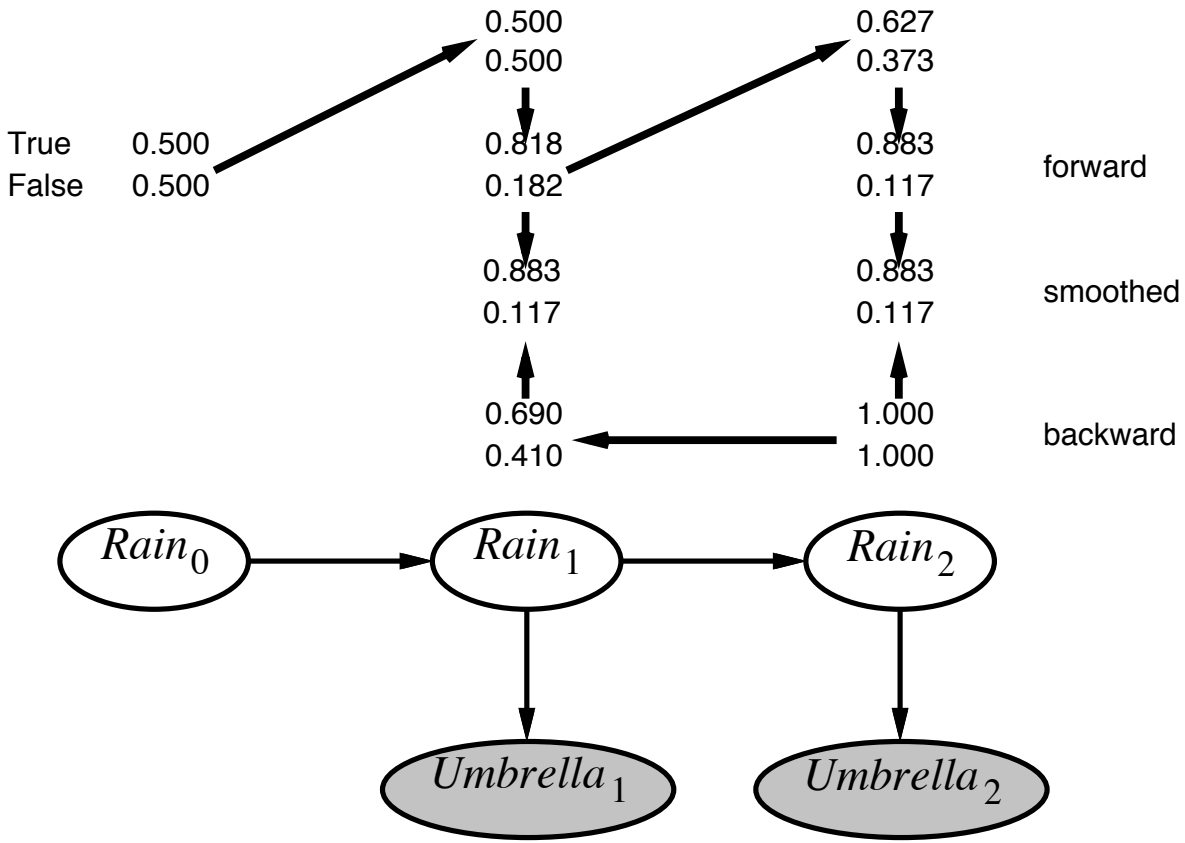
$\mathbf{P}(X_2|\mathbf{e}_{1:2}) = \langle 0.883, 0.117 \rangle$

$\mathbf{b}_{2:2} = \langle 0.690, 0.410 \rangle$

$\mathbf{b}_{3:2} = \langle 1, 1 \rangle$



Smoothing example — conclusion



Forward–backward algorithm: cache $\mathbf{f}_t$-messages as we move
Time linear in t (polytree inference), space $O(t \cdot |\mathbf{f}|)$

Simplifications for Hidden Markov models – complete

$\mathbf{X}_t$ is a single, discrete variable (as is $\mathbf{E}_t$ usually, too)
Domain of X_t is $\{1, \ldots, S\}$

Transition matrix $\mathbf{T}_{ij} = P(X_t = j | X_{t-1} = i)$, e.g., $\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$

Sensor matrix $\mathbf{O}_t$ for each t, diagonal elements $P(e_t | X_t = i)$.
For instance, with $U_1 = \texttt{true}$ we get

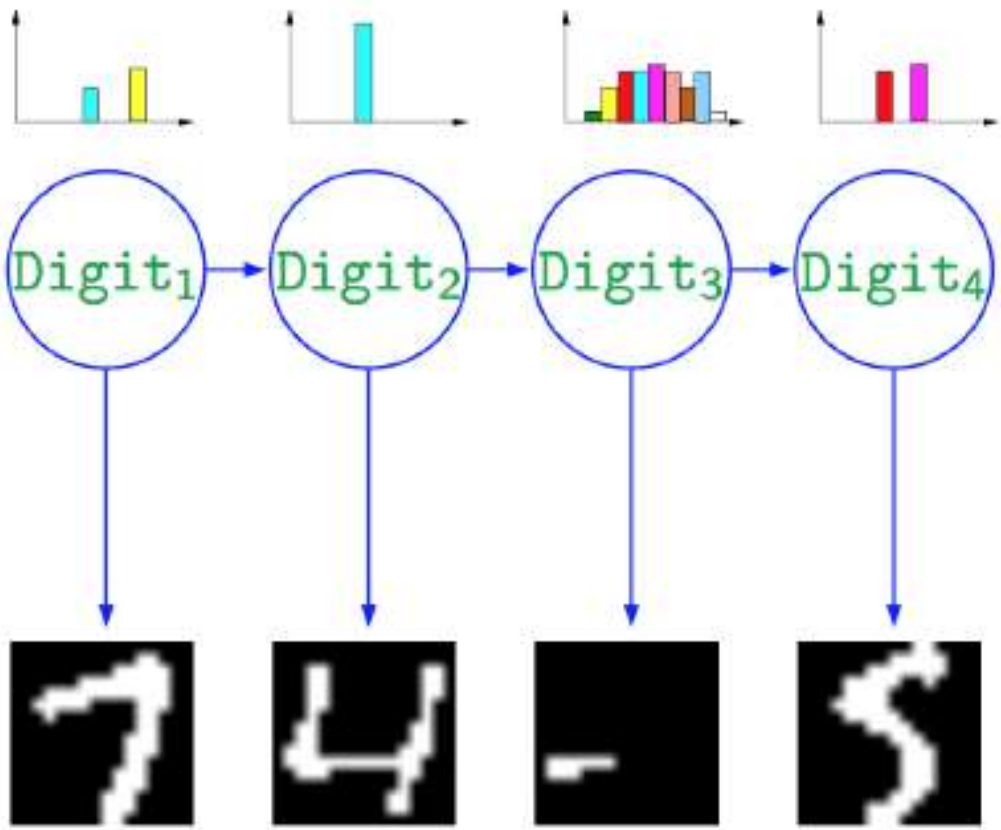
$$\mathbf{O}_1 = \begin{pmatrix} P(u_1|x_1) & 0 \\ 0 & P(u_1|\neg x_1) \end{pmatrix} = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$$

Forward and backward messages as column vectors:

$$\begin{aligned} \mathbf{f}_{1:t+1} &= \alpha \mathbf{O}_{t+1} \mathbf{T}^\top \mathbf{f}_{1:t} \\ \mathbf{b}_{k+1:t} &= \mathbf{T} \mathbf{O}_{k+1} \mathbf{b}_{k+2:t} \end{aligned}$$

The FB-algorithm needs time $O(S^2 \cdot t)$ and space $O(S \cdot t)$

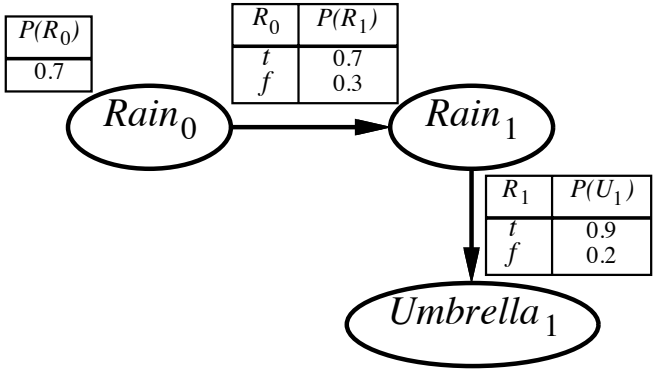
How to classify ZIP-codes?

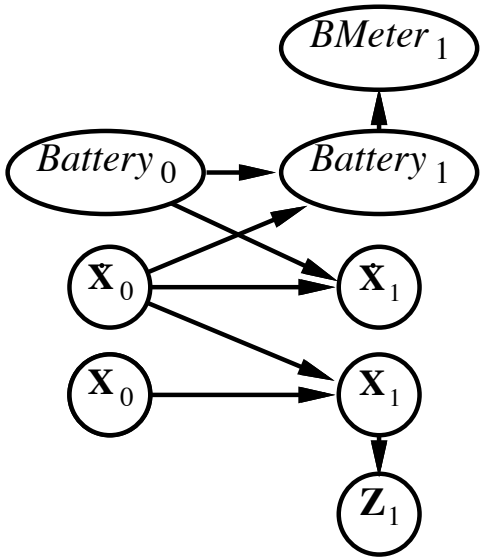


- Can we take the most probable digit *per image* and use for classification?
- **NO! Most likely sequence IS NOT the sequence of most likely states!** For details, see Sec 14.2.3 in the book.

Dynamic Bayesian networks

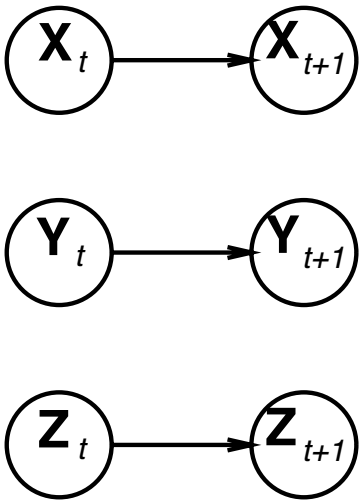
$\mathbf{X}_t$, $\mathbf{E}_t$ contain arbitrarily many variables in a replicated Bayes net

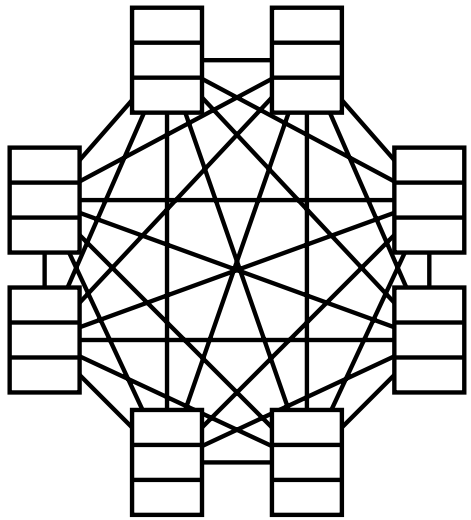




DBNs vs. HMMs

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





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

Summary

- **Temporal models** use state and sensor variables replicated over time
- **Markov** and **stationarity** assumptions, 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, prediction, 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; exact update intractable; approximations exist