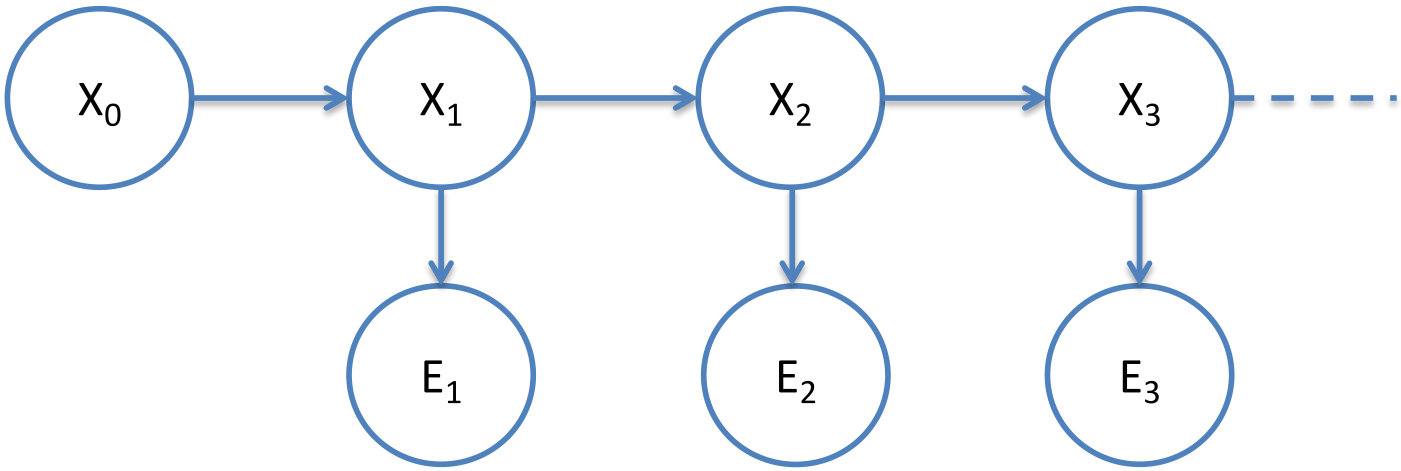
- Markov chains are pretty easy!
- But sometimes they aren't realistic…

- What if we can't directly know the states of the model, but we can see some indirect evidence resulting from the states?

# Weather

- Regular Markov chain
  - Each day the weather is rainy or sunny.
  - $P(X_t = \text{rain} \mid X_{t-1} = \text{rain}) = 0.7$
  - $P(X_t = \text{sunny} \mid X_{t-1} = \text{sunny}) = 0.9$
- Twist:
  - Suppose you work in an office with no windows. All you can observe is weather your colleague brings their umbrella to work.

# Hidden Markov Models

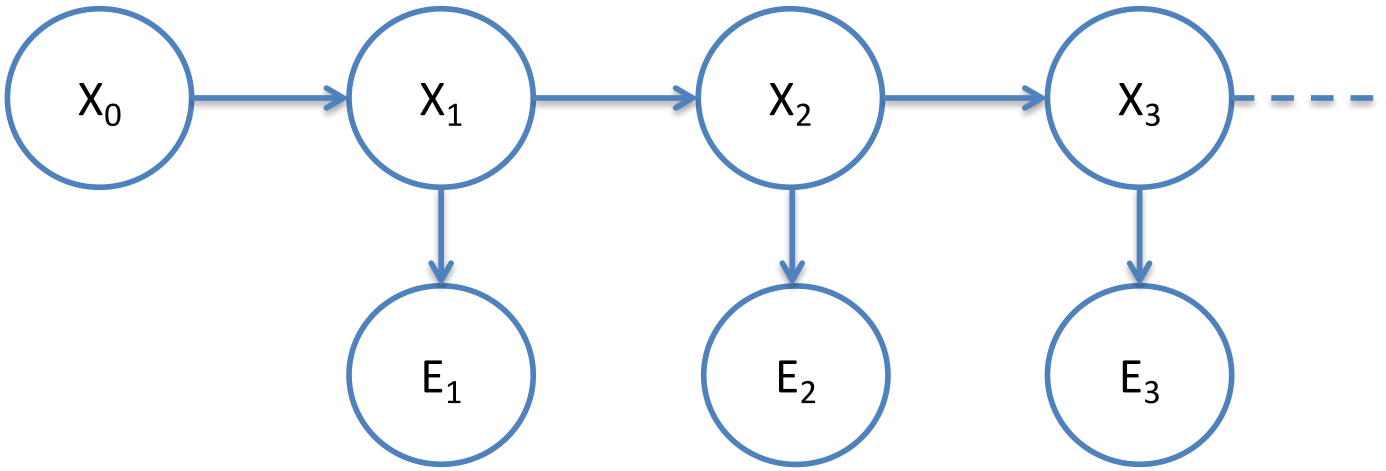



- The X's are the state variables (never directly observed).
- The E's are evidence variables.

# Common real-world uses

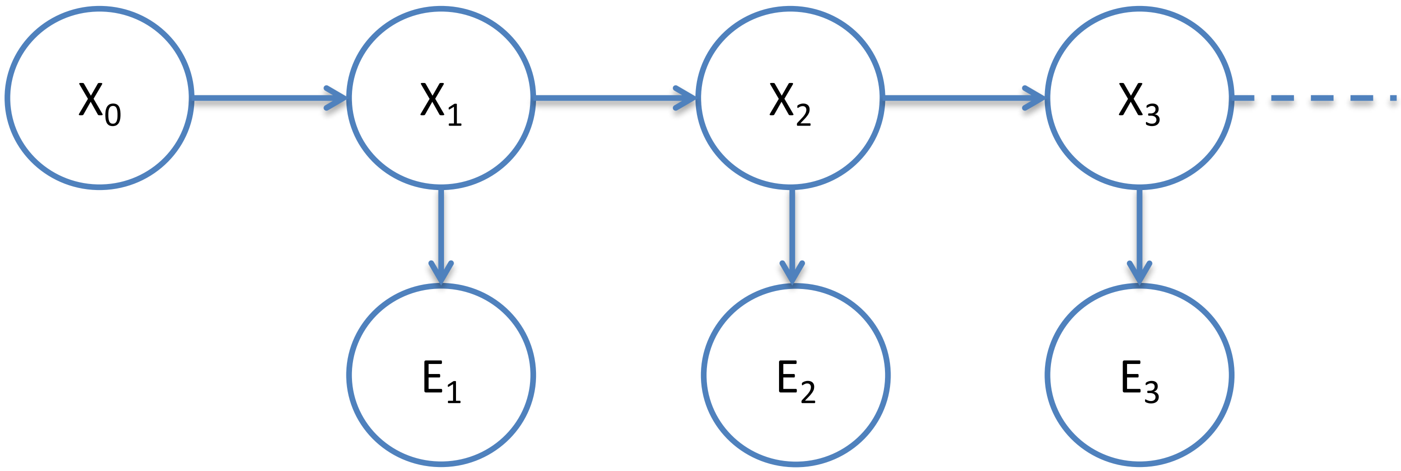
- Speech processing:
  - Observations are sounds, states are words.
- Localization:
  - Observations are inputs from video cameras or microphones, state is the actual location.
- Video processing (example):
  - Extracting a human walking from each video frame. Observations are the frames, states are the positions of the legs.

# Hidden Markov Models




- $P(X_t \mid X_{t-1}, X_{t-2}, X_{t-3}, \ldots) = P(X_t \mid X_{t-1})$
- $P(X_t \mid X_{t-1}) = P(X_1 \mid X_0)$
- $P(E_t \mid X_{0:t}, E_{0:t-1}) = P(E_t \mid X_t)$
- $P(E_t \mid X_t) = P(E_1 \mid X_1)$

# Hidden Markov Models




- What is $P(X_{0:t}, E_{1:t})$?

$$P(X_0)\prod_{i=1}^{t} P(X_i \mid X_{i-1}) P(E_i \mid X_i)$$

# Common questions

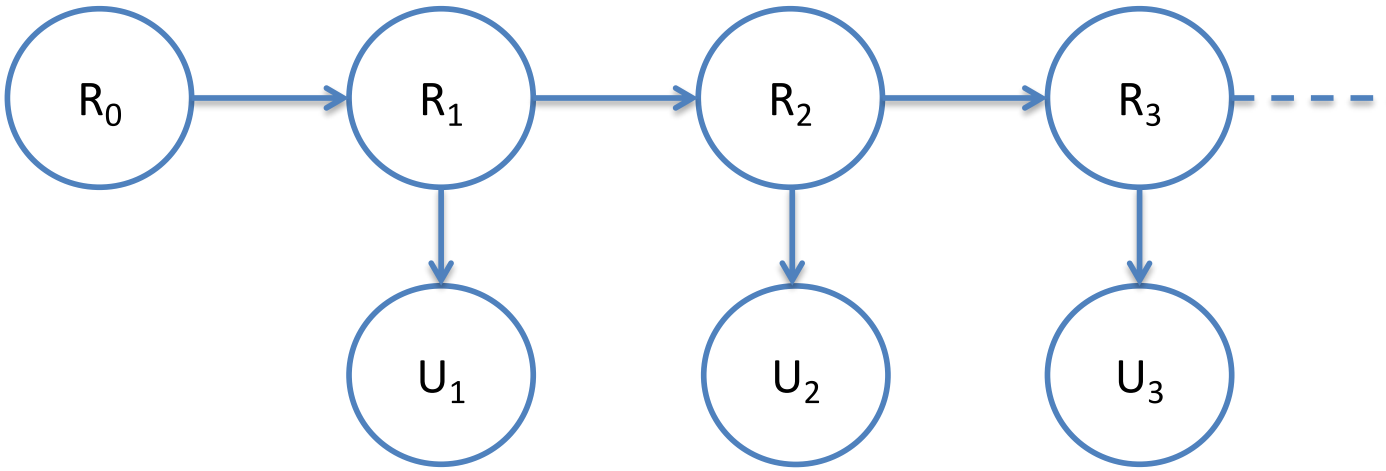
- **Filtering**: Given a sequence of observations, what is the most probable *current* state?
  - Compute $P(X_t \mid e_{1:t})$
- **Prediction**: Given a sequence of observations, what is the most probable *future* state?
  - Compute $P(X_{t+k} \mid e_{1:t})$ for some $k > 0$
- **Smoothing**: Given a sequence of observations, what is the most probable *past* state?
  - Compute $P(X_k \mid e_{1:t})$ for some $k < t$

# Common questions

- **Most likely explanation:** Given a sequence of observations, what is the most probable sequence of states?
  – Compute $\underset{x_{1:t}}{\operatorname{argmax}} P(x_{1:t} \mid e_{1:t})$

- **Learning**: How can we estimate the transition and sensor models from real-world data? (Future machine learning class?)

# Hidden Markov Models




- $P(R_t = yes \mid R_{t-1} = yes) = 0.7$
  $P(R_t = yes \mid R_{t-1} = no) = 0.1$
- $P(U_t = yes \mid R_t = yes) = 0.9$
  $P(U_t = yes \mid R_t = no) = 0.2$

# Filtering

- Filtering is concerned with finding the most probable "current" state from a sequence of evidence.
- Let's compute this.

# Recall the "mini-forward algorithm"

For Markov chains:

$$P(X_{t+1}) = \sum_{x_t} P(X_{t+1} \mid x_t) P(x_t)$$

with matrices: $v_{t+1} = v_t * T$, with $v_0 = P(X_0)$

For HMM's:

$$P(X_{t+1} \mid e_{1:t+1}) =$$

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

# Forward algorithm

- Today is Day 2, and I've been pulling all-nighters for two days!
- My colleague brought their umbrella on days 1 and 2.
- What is the probability it is raining today?
  - that is, find $P(X_t \mid e_{1:t})$ [*filtering*]
- Assume initial distribution of rain/not-rain for Day 0 is 50-50.

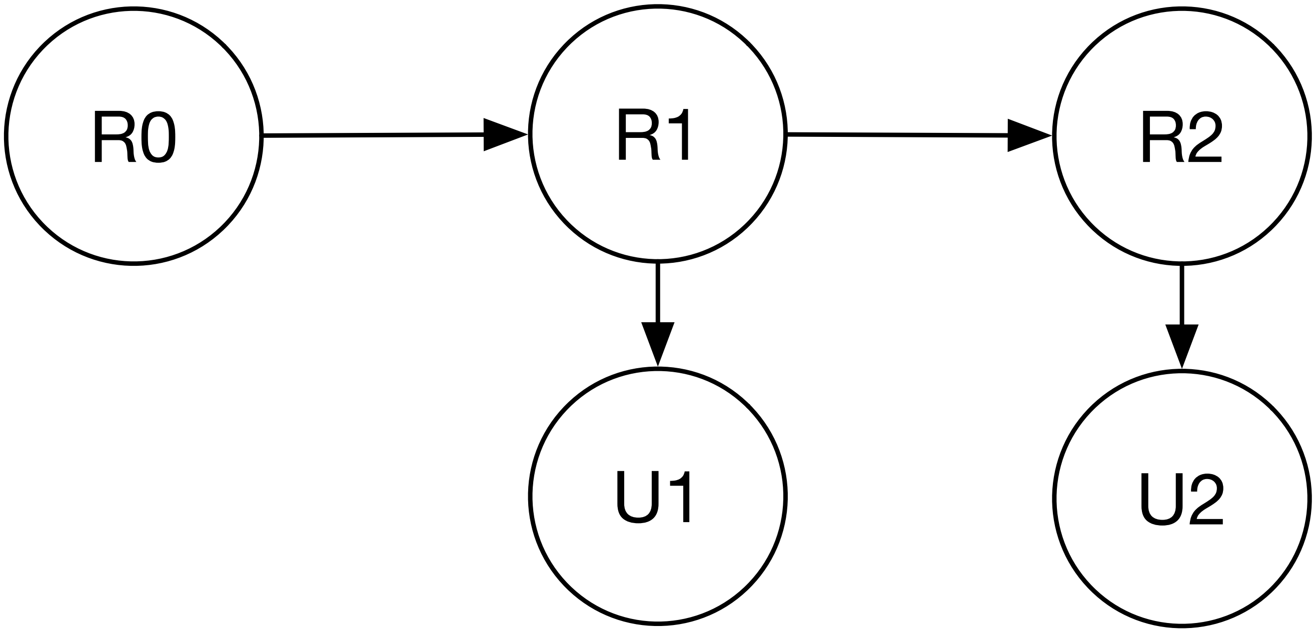
# Matrices to the rescue!

- Define a transition matrix T as normal.
- Define a sequence of observation matrices $O_1$ through $O_t$.
- Each O matrix is a diagonal matrix with the entries corresponding to observation at time t given each state.

$$f_{1:t+1} = \alpha f_{1:t} \cdot T \cdot O_{t+1}$$

where each f is a row vector containing the probability distribution at timestep t.

f1:0=[0.5, 0.5]    f1:1=[0.75, 0.25]    f1:2=[0.846, 0.154]




```
T  = [0.7, 0.3]
     [0.1, 0.9]

O1 = [0.9, 0.0]
     [0.0, 0.2]

O2 = [0.9, 0.0]
     [0.0, 0.2]
```

f1:0 = P(R0) = [0.5, 0.5]

f1:1 = P(R1 | u1) = $\boldsymbol{\alpha}$ * f1:0 * T * O1 = $\boldsymbol{\alpha}$[0.36, 0.12] = [0.75, 0.25]

f1:2 = P(R2 | u1, u2) = $\boldsymbol{\alpha}$ * f1:1 * T * O2 = $\boldsymbol{\alpha}$[0.495, 0.09] = [.846, .154]

# Forward algorithm

- Note that the forward algorithm only gives you the probability of $X_t$ taking into account evidence at times 1 through t.
- In other words, say you calculate $P(X_1 \mid e_1)$ using the forward algorithm, then you calculate $P(X_2 \mid e_1, e_2)$.
  - Knowing e2 changes your calculation of X1.
  - That is, $P(X_1 \mid e_1) \mathrel{!=} P(X_1 \mid e_1, e_2)$