IT5005 Artificial Intelligence

Sirigina Rajendra Prasad
AY2025/2026: Semester 1

Markov Chains and Hidden Markov Models

Agenda

- Recap
- Markov Models
- Hidden Markov Models
  - Bayesian Filtering
- Conclusion

Recap

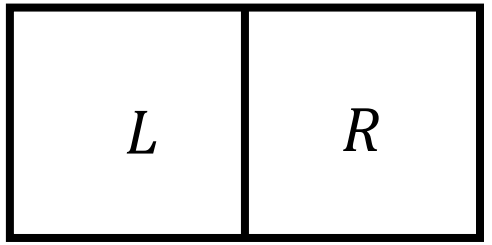
Random Variables

Probability Distributions

Bayesian Networks

Discrete Random Variables: Recap

- X: Discrete random variable with finite domain
  - Examples:
    - X: State of stock market
      - $dom(X) = \{high, low\}$
    - X: Word in a corpus
      - $dom(X) = \{'Chains', 'Markov', 'Rock'\}$
    - X: Location of the agent in Two-Room Vacuum World
      - $dom(X) = \{L, R\}$

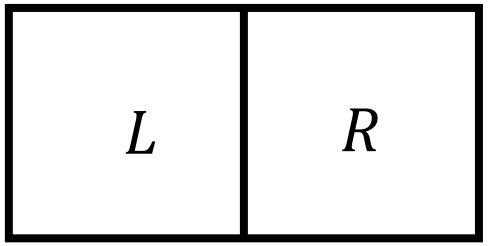


- We also need to define probability distributions for discrete random variables

Probability Distributions: Recap

Bold Font for distribution

- $\boldsymbol{P}(X)$: Probability distribution
  - Vector with size same as that of domain of X
  - Values in the vector indicates the probabilities of X taking a value in that domain
  - Example:
    - State of stock market with $dom(X) = \{high, low\}$
      - $\boldsymbol{P}(X) = \begin{bmatrix} P(X = high) \\ P(X = low) \end{bmatrix}$
    - Text corpus with vocabulary [“$Chains$”, “$Markov$”, “$Rock$”]
      - $\boldsymbol{P}(X) = \begin{bmatrix} P(X = "Chains") \\ P(X = "Markov") \\ P(X = "Rock") \end{bmatrix}$
    - Two-room vacuum world
      - $\boldsymbol{P}(X) = \begin{bmatrix} P(X = L) \\ P(X = R) \end{bmatrix}$



Probability Distributions: Recap

- Properties of Probability Distribution:
  - $P(X = x) \geq 0, \ \forall\, x \in dom(X)$
  - $\sum_{x \in dom(X)} P(X = x) = 1$

  - **Example**: Let $\mathrm{dom}(X) = \{\text{‘}Chains\text{’}, \text{‘}Markov\text{’}, \text{‘}Rocks\text{’}\}$.

    - $\boldsymbol{P}(X) = \begin{bmatrix} P(X = \text{"}Chains\text{"}) \\ P(X = \text{"}Markov\text{"}) \\ P(X = \text{"}Rocks\text{"}) \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.4 \end{bmatrix}$

Conditional Probability Distribution $P(X|Y = y)$

- Properties of Conditional Probability Distributions
  - $P(X = x|Y = y) \geq 0 \;\; \forall\, x \in dom(X)$
  - $\sum_{x \in dom(X)} P(X = x|Y = y) = 1$

- **Example**: Let $\mathrm{dom}(X) = dom(Y) = \{"Chains","Markov","Rocks"\}$.
  - $\boldsymbol{P}(X|Y = \text{“}Markov\text{”}) = \begin{bmatrix} P(X = \text{“}Chains\text{”}|Y = "Markov") \\ P(X = \text{“}Markov\text{”}|Y = \text{“}Markov\text{”}) \\ P(X = \text{“}Rocks\text{”}|Y = "Markov") \end{bmatrix}$

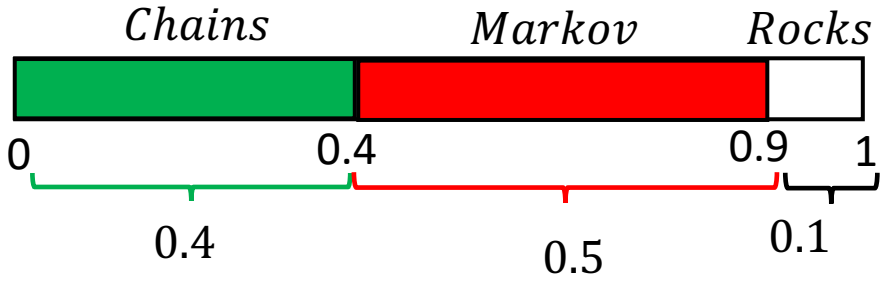
$$= \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \end{bmatrix}$$

Sampling from Probability Distributions

- If distribution of data is known, samples (data) can be generated
- Example:
  - $\boldsymbol{P}(X) = \begin{bmatrix} P(X = \text{“}Chains\text{”}) \\ P(X = \text{“}Markov\text{”}) \\ P(X = \text{“}Rocks\text{”}) \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.5 \\ 0.1 \end{bmatrix}$

Most Probable Word: $Markov$

```
>>> import numpy as np
>>> vocabulary = ["Chains","Markov","Rocks"]
>>> np.random.choice(vocabulary, p=[0.4,0.5,0.1])
'Chains'
```




```
>>> import random
>>> random.uniform(0,1)
0.2894913983065621
```

Generated Sample: $Chains$

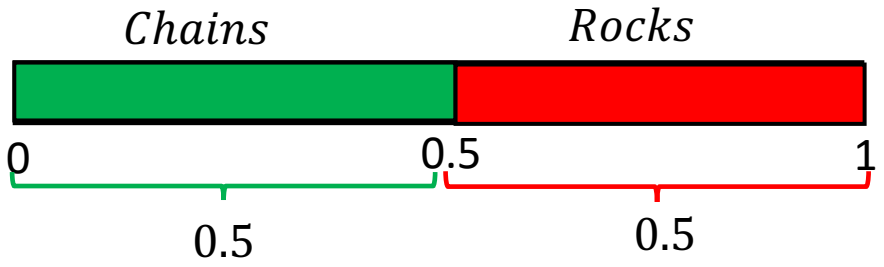
Sampling from Probability Distributions

- If distribution of data is known, samples (data) can be generated
- Example:

$$\begin{bmatrix} P(X = \text{“}Chains\text{”} | Y = \text{“}Markov\text{”}) \\ P(X = \text{“}Markov\text{”} | Y = \text{“}Markov\text{”}) \\ P(X = \text{“}Rocks\text{”} | Y = \text{“}Markov\text{”}) \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \end{bmatrix}$$

Most Probable Word:
$Chains$ and $Rocks$ are equally probable

```
>>> import numpy as np
>>> vocabulary = ["Chains","Markov","Rocks"]
>>> np.random.choice(vocabulary, p=[0.5,0,0.5])
'Chains'
```



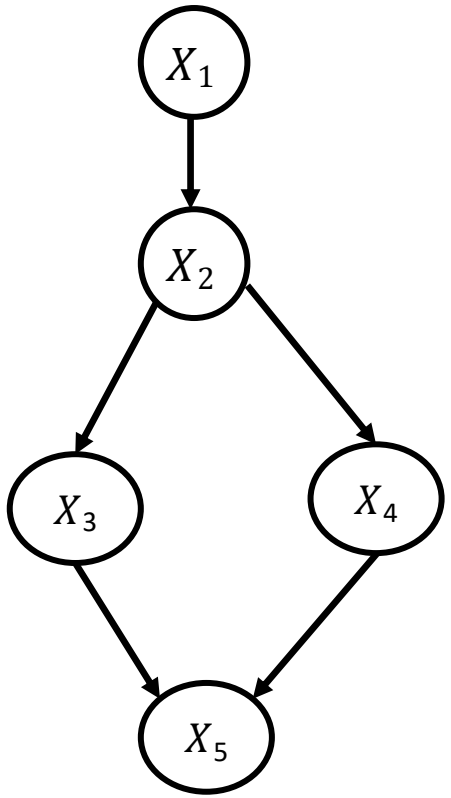

```
>>> import random
>>> random.uniform(0,1)
0.2894913983065621
```

Generated Sample: $Chains$

Bayesian Networks (BNs):

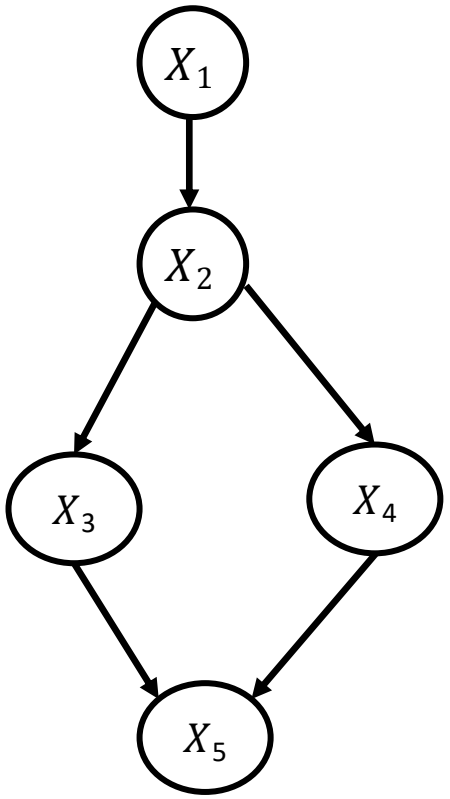
- Directed Acyclic Graphs
  - Nodes represent variables and edges represent relation between two variables
- BNs allows factorization of joint probability distributions
  - Reduces computational complexity
  - Structure allows us to do efficient reasoning
  - Example:

$$P(X_1, X_2, X_3, X_4, X_5) = P(X_1)P(X_2|X_1)P(X_3|X_2)P(X_4|X_2)P(X_5|X_3, X_4)$$



Bayesian Networks (BNs):

- Descendants won't influence the belief about parents unless they are observed
- Examples:
  - Query 1: $\boldsymbol{P}(X_2)$
    - None of the descendants of X_2 are observed
    - Descendants are irrelevant for the query
  - Query 2: $\boldsymbol{P}(X_2|X_5 = x_5)$
    - Descendant X_5 is observed
    - All descendants that lie between the query variable and observed descendant are relevant for the query



- Example: Query is $\boldsymbol{P}(X_2)$
  - We need to marginalize $X_1, X_3, X_4,\ and\ X_5$

$$\boldsymbol{P}(X_2) = \sum_{x_1 \in X_1} \sum_{x_3 \in X_3} \sum_{x_4 \in X_4} \sum_{x_5 \in X_5} P(X_1, X_2, X_3, X_4, X_5)$$

$$= \sum_{x_1 \in X_1} \sum_{x_3 \in X_3} \sum_{x_4 \in X_4} \sum_{x_5 \in X_5} P(x_1)P(X_2|x_1)P(x_3|X_2)P(x_4|X_2)P(x_5|x_3, x_4)$$

$$= \sum_{x_1 \in X_1} P(x_1)P(X_2|x_1) \sum_{x_3 \in X_3} P(x_3|X_2) \sum_{x_4 \in X_4} P(x_4|X_2) \sum_{x_5 \in X_5} P(x_5|x_3, x_4)$$

Evaluates to 1

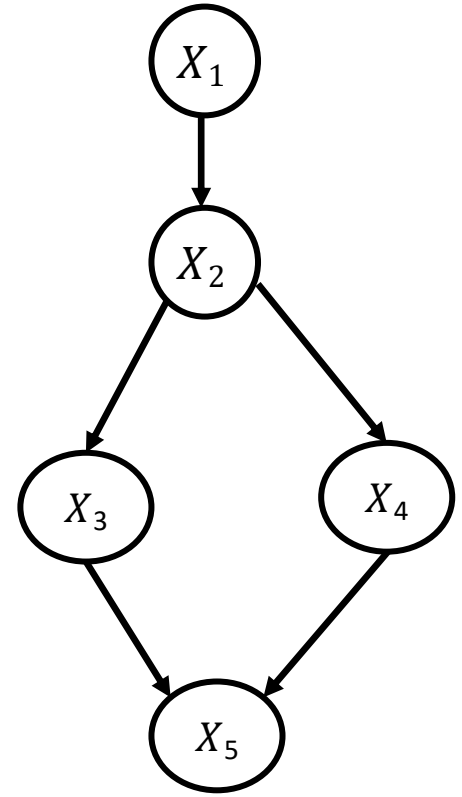
$$= \sum_{x_1 \in X_1} P(x_1)P(X_2|x_1) \sum_{x_3 \in X_3} P(x_3|X_2) \sum_{x_4 \in X_4} P(x_4|X_2)$$

Evaluates to 1

$$= \sum_{x_1 \in X_1} P(x_1)P(X_2|x_1) \sum_{x_3 \in X_3} P(x_3|X_2)$$

Evaluates to 1

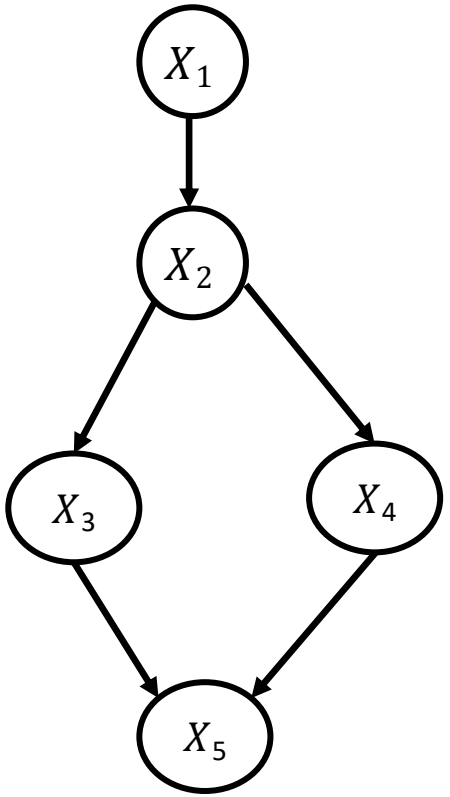
$$= \sum_{x_1 \in X_1} P(x_1)P(X_2|x_1)$$



Notation:

$$P(X_5 = x_5|X_3 = x_3, X_4 = x_4) = P(x_5|x_3, x_4)$$

Example: Query is $\boldsymbol{P}(X_2)$



- After marginalization of descendants

$$\boldsymbol{P}(X_2) = \sum_{x_1 \in X_1} \sum_{x_3 \in X_3} \sum_{x_4 \in X_4} \sum_{x_5 \in X_5} P(X_1, X_2, X_3, X_4, X_5)$$

$$= \sum_{x_1 \in X_1} P(x_1) P(X_2 | x_1)$$

- Ignore the descendants as they don't influence belief about X_2

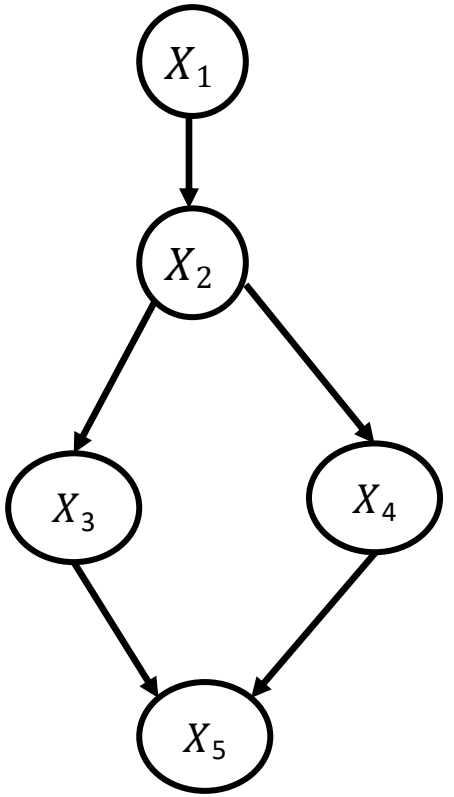
- $X_3, X_4, and\ X_5$ are irrelevant for the query

- Example: Query is $\boldsymbol{P}(X_2|X_5 = x_5)$
  - Marginalize $X_1, X_3, and\ X_4$

$$\boldsymbol{P}(X_2) = \sum_{x_1 \in X_1} \sum_{x_3 \in X_3} \sum_{x_4 \in X_4} P(X_1, X_2, X_3, X_4, X_5)$$

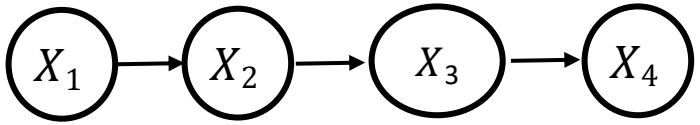
$$= \sum_{x_1 \in X_1} \sum_{x_3 \in X_3} \sum_{x_4 \in X_4} P(x_1)P(X_2|x_1)P(x_3|X_2)P(x_4|X_2)P(x_5|x_3, x_4)$$

    - None of the sums evaluate to 1
  - All descendants contribute to the probability of X_2

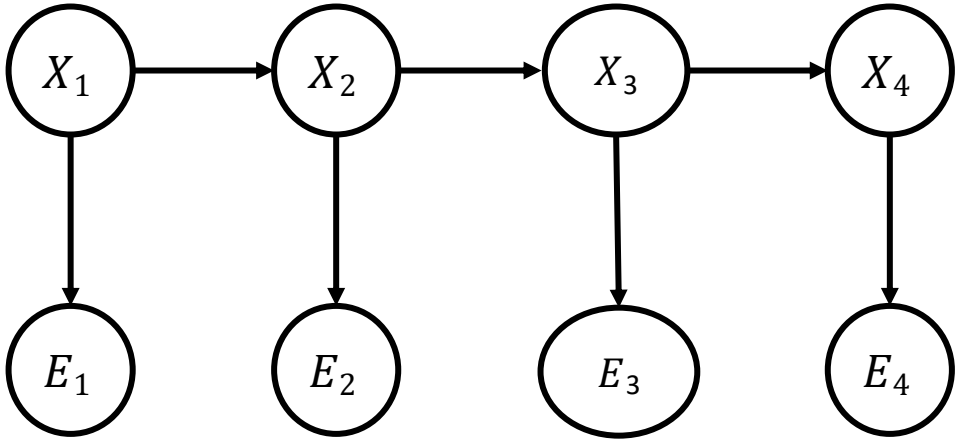


Bayesian Networks (BNs): Examples

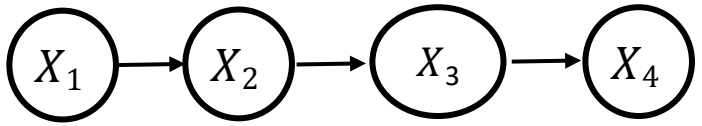
- Consider the following Bayesian Networks
  - **BN 1** with four random variables X_1, X_2, X_3, X_4.



  - **BN 2** with eight random variables $X_1, X_2, X_3, X_4, E_1, E_2, E_3, E_4$.



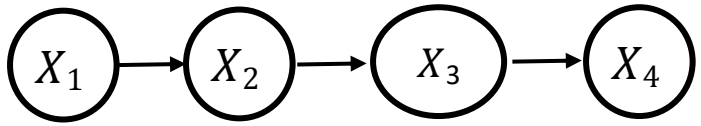
Bayesian Network 1 (BN 1)



- Identify the relevant variables for the following queries
  - $\boldsymbol{P}(X_1)$
  - $\boldsymbol{P}(X_2)$
  - $\boldsymbol{P}(X_2|X_3 = x_3)$
  - $\boldsymbol{P}(X_2|X_4 = x_4)$
- Factorize the joint distributions for the above queries
- Identify the variables that need to be marginalized

BN 1: Irrelevant Variables and Factorization

- **Query: $\boldsymbol{P}(X_1)$**

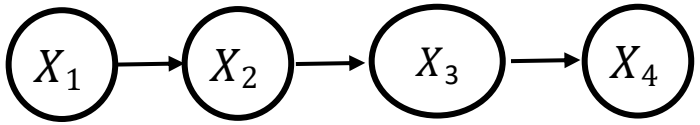


- Variables X_2, X_3, and X_4 are descendant variables
  - They won't influence the belief about X_1
  - $\therefore X_2$, X_3, and X_4 are irrelevant for the query

- Factorization: $P(X_1)$.
- Marginalization is not needed

BN 1: Irrelevant Variables and Factorization

- **Query: $\boldsymbol{P}(X_2)$**



- Variables X_3 and X_4 are descendant variables
  - They won't influence the belief about X_2
  - $\therefore$ X_3 and X_4 are irrelevant for the query

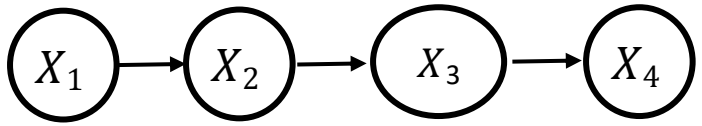
- X_2 is not independent of X_1
  - Therefore, X_1 is a relevant variable

- Factorization: $P(X_1, X_2) = P(X_1)P(X_2|X_1)$

- Marginalize X_1 because it is not given

BN 1: Irrelevant Variables and Factorization

- **Query:** $\boldsymbol{P}(X_2|X_3 = x_3)$



- X_4 is descendant variable to X_2 and X_3
  - $\therefore X_4$ is an irrelevant variable for this query

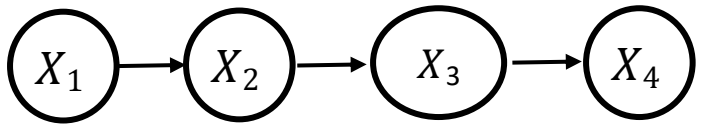
- X_2 is not guaranteed to be independent of X_1 given X_3
  - $\therefore X_1$ is a relevant variable for this query

- Factorization: $P(X_1, X_2, X_3 = x_3) = P(X_1)P(X_2|X_1)\,P(X_3 = x_3|X_2)$

- Marginalize X_1 because it is not given

BN 1: Irrelevant Variables and Factorization

- **Query:** $\boldsymbol{P}(X_2|X_4 = x_4)$



- X_2 is not guaranteed to be independent of X_1 and X_3 given X_4
  - ∴ Variables X_1 and X_3 are relevant for the query

- Factorization: $P(X_1, X_2, X_3, X_4 = x_4) = P(X_1)P(X_2|X_1)\,P(X_3|X_2)\,P(X_4 = x_4|X_3)$

- Marginalize X_1 and X_3 because they are not given

Dynamic Bayesian Networks

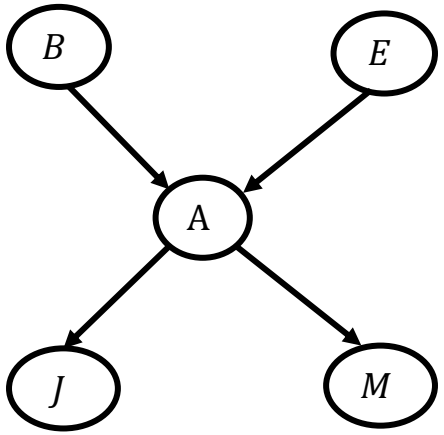
Motivation

Markov Chains

Hidden Markov Models

So far: Bayesian Networks

- Probabilistic reasoning in static world

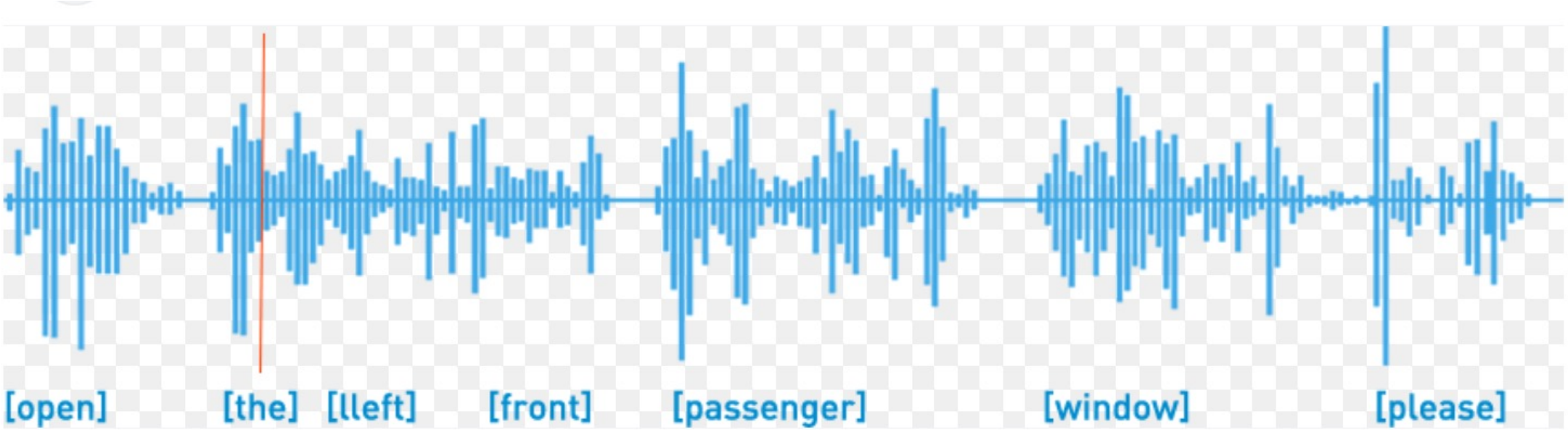


Observed:
John calls

Query:
Burglary?

- Given partial state information and network structure:
  - Answers queries related to unobserved states
  - Leverage conditional independencies
- Binary random variables
  - Variables take only two values: $true$ or $false$

However, Real World is Dynamic



https://glas.ai/speech-understanding.html

Dynamic Bayesian Networks (DBNs)

- Probabilistic reasoning in dynamic world
  - Relies on temporal modelling of tasks

- Applications
  - Speech Signal Processing
    - Speech recognition, etc.
  - Natural Language Processing
    - POS tagging, etc.
  - Robotics
    - Localization, path planning, navigation, etc.
  - Finance
    - Risk management and portfolio management
  - Medicine
    - Disease progression modelling and patient monitoring
  - Forecasting
    - Weather forecasting, economic forecasting, traffic forecasting, etc.
  - Recommender systems
    - Models user behavior and preferences over time to provide recommendations in streaming, e-commerce, etc.

Dynamic Bayesian Networks: Assumptions

- Assumption 1:
  - Markov chains
    - Evolution of random variables (states) over time form Markov chains
    - Markov models inherently incorporate time

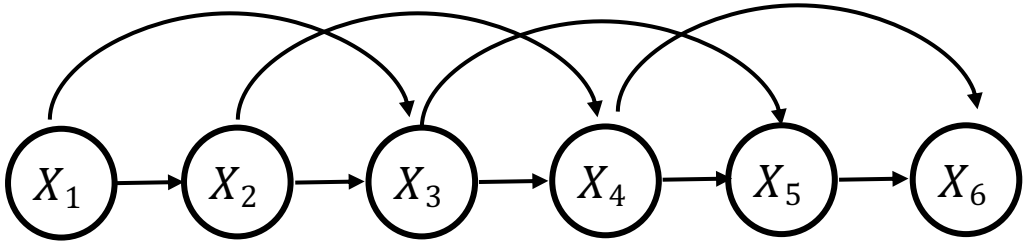
- Assumption 2:
  - Stationarity
    - Transition probabilities between states are always the same

Assumption 1: Markov Chains

Andrey Andreyevich Markov
(1856-1922)

Markov Chain of Order n:
Probability of outcome of state X depends on the state of the previous n outcomes.

Markov chain of order 2



$$P(X_5|X_4, X_3, X_2, X_1) = P(X_5|X_4, X_3)$$

Markov Chain of First-Order

- Value or the state of random variable (X_t) in each time step only depends on the previous time step, i.e., future evolution is independent of its history given the present

- $P(X_1, X_2, X_3, \dots, X_t) = P(X_t | X_1, X_2, X_3, \dots, X_{t-1})\, P(X_1, X_2, X_3, \dots, X_{t-1})$

$$= P(X_t | X_{t-1})\, P(X_1, X_2, X_3, \dots, X_{t-1})$$

$$= P(X_t | X_{t-1})\, P(X_{t-1} | X_{t-2}) P(X_1, X_2, X_3, \dots, X_{t-2})$$

$$= P(X_t | X_{t-1})\, P(X_{t-1} | X_{t-2}) \dots P(X_2 | X_1) P(X_1)$$

Markov Chains: Examples

Stock Market:

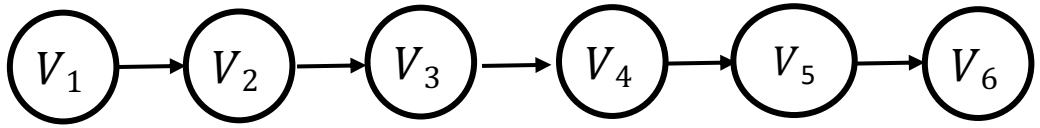
State: Stock Value (V)

$V = \{high, low\}$

Sequence:

$high, low, low, high, high, high, \ldots.$

$V_1, \quad V_2, \quad V_3, \quad V_4, \quad V_5, \quad V_6$



NLP:

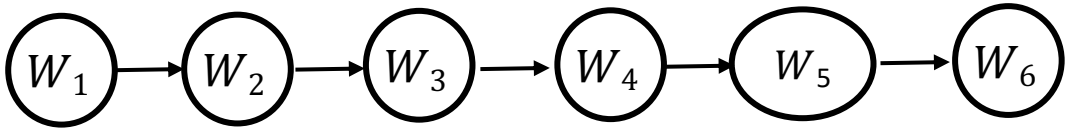
State: Word (W)

$W = \{\text{“}Markov\text{”}, \text{“}Chains\text{”}, \text{“}Rock\text{”}\}$

Sequence:

$Markov \ \ Chains. \ Markov \ Chains \ Rock.$

$W_1, \quad W_2, \quad W_3, \quad W_4, \quad W_5$



Robotics:

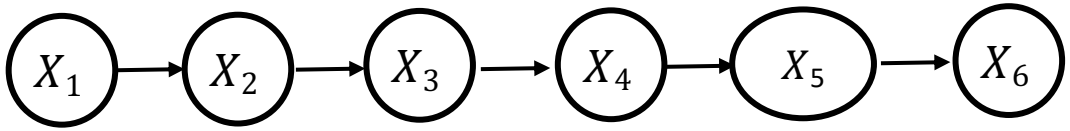
State: Location of the Robot (X)

$X = \{left, right\}$

Sequence:

$left, left, right, left, left, right$

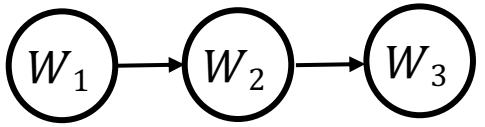
$X_1, \quad X_2, \quad X_3, \quad X_4, \quad X_5, \quad X_6$



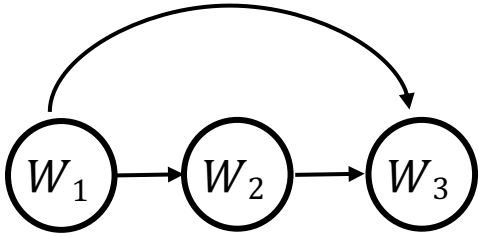
Markov Chains in NLP

- Unigram Model:
  - Each word is independent
  - Ex: $P(\text{"Markov Chains Rocks"}) = P(\text{"Markov"})P(\text{"Chains"})P(\text{"Rocks"})$

- Bigram Model (First-order MC)
  - Current word depends only on previous word
  - Ex: $P(\text{"Markov Chains Rocks"}) = P(\text{"Markov"})P(\text{"Chains"}|\text{"Markov"})P(\text{"Rocks"}|\text{"Chains"})$



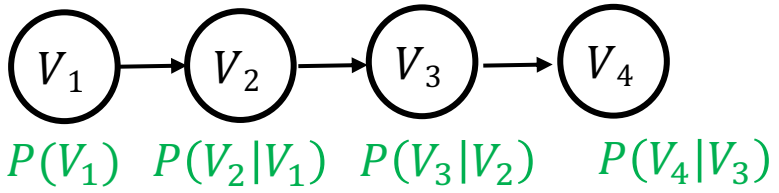
- Trigram Model (Second-order MC)
  - Current word depends on previous two words
  - Ex: $P(\text{"Markov Chains Rocks"}) = P(\text{"Markov"})P(\text{"Chains"}|\text{"Markov"})$
    $P(\text{"Rocks"}|\text{"Chains"},\text{"Markov"})$

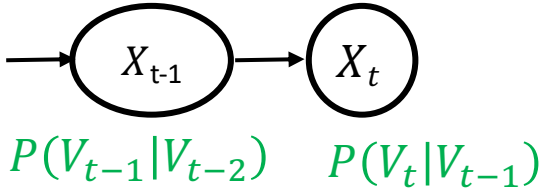


Question: How to calculate the probabilities on the RHS?

Assumption 2: Stationarity of Transition Probabilities

- Transition probabilities between the states are always the same
  - $P(X_2|X_1) = P(X_3|X_2) = \cdots = P(X_t|X_{t-1})$
  - Eg: $P(V_2 = high|V_1 = high) = P(V_{11} = high|V_{10} = high)$
    - $P(V_2 = high|V_1 = high)$ is independent of time





- Non-stationary Markov models
  - Transition probabilities change over time
  - Out-of-scope for this course

Markov Chains are Bayesian Networks

- First-Order MC
  - Treat $X_t, t = \{1, \dots, T\}$, as T variables

- Semantics of Bayesian networks apply
  - Local Semantics
    - Conditional independence
    - Markov Blanket
  - Global Semantics
    - d-Separation Properties

Markov Chains: Joint Distribution

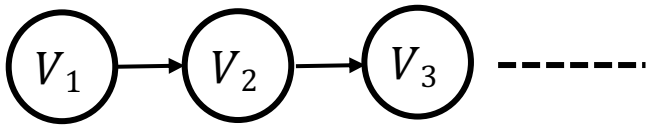
- Joint distribution of first t variables:
  - $P(X_1, X_2, X_3, \ldots, X_t) = \mathrm{P}(X_1) \prod_{i=2}^{t} P(X_i|X_{i-1})$

- We only need two types of parameters to specify a Markov model:
  - Initial state probability: $P(X_1)$
  - Transition Probability: $P(X_t|X_{t-1})$
    - Known as conditional probability table (CPT)
    - Specifies how state evolves over time

Markov Chains

- Given the initial distribution and transition probabilities, we can predict the distribution of state at future time instants

Example: Stock Value of a Firm



- Stock value (V) is encoded as two states: $\{high,\ low\}$
- Parameters:
  - $P(V_1 = high) = 1.0$
  - CPT:

Four different ways of representing state transition models

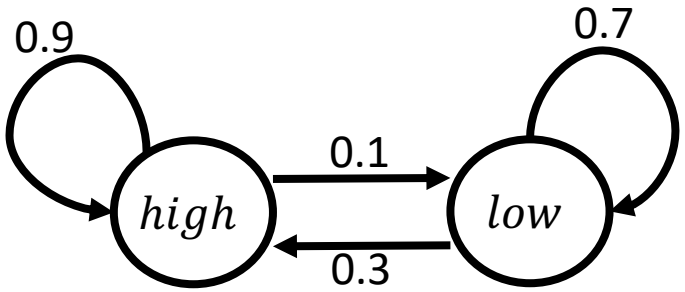
1 Conditional Probability Table

V_{t-1}	V_t	$P(V_t \mid V_{t-1})$
$high$	$high$	0.9
$high$	low	0.1
low	$high$	0.3
low	low	0.7

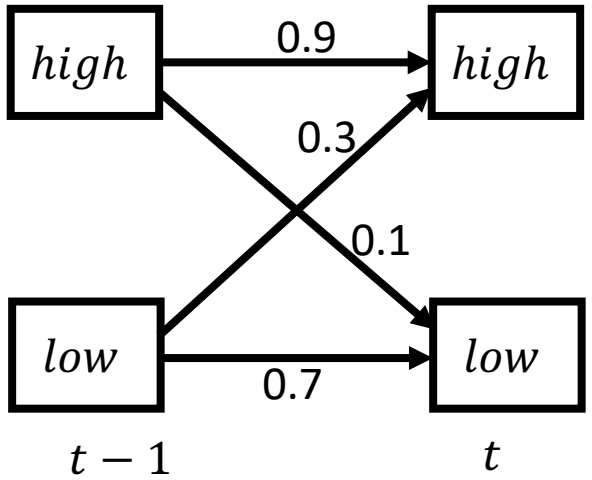
2 State Transition Matrix

$$V_{t-1} \begin{matrix} \\ high \\ low \end{matrix} \overset{V_t}{\begin{matrix} high & low \end{matrix}} \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} = \boldsymbol{T}$$

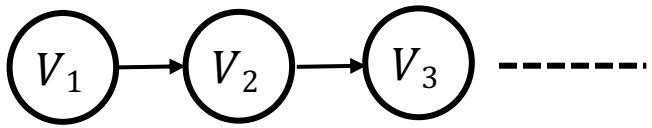
3 State Transition Diagram



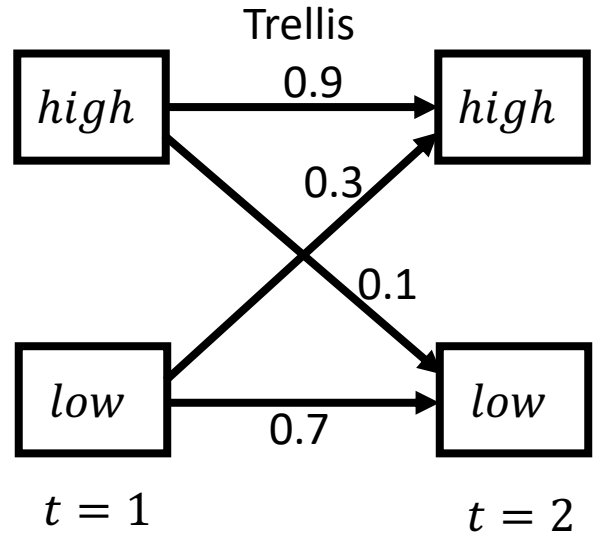
4 Trellis Diagram



Example: Stock Value of a Firm

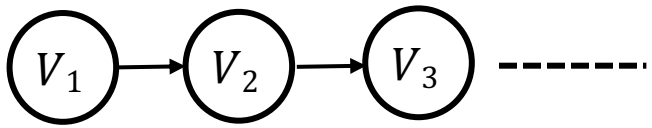


- Given $\boldsymbol{P}(V_1)$ and $\boldsymbol{P}(V_t|V_{t-1})$, What is $\boldsymbol{P}(V_2)$?
- Query Variable: V_2
- Evidence Variables: None
- Relevant Variables: V_1 and V_2
- Irrelevant Variables: $V_t, t > 2$
- Factorization: $P(V_1, V_2) = P(V_1)P(V_2|V_1)$
- Variables to Marginalize: V_1

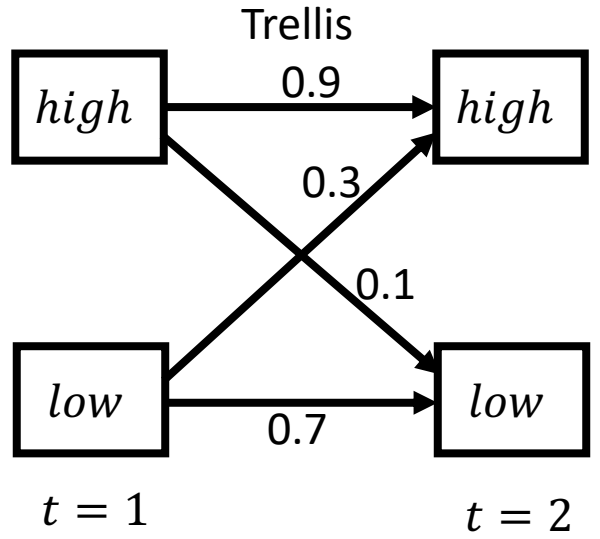


$$V_{t-1} \begin{matrix} \\ \end{matrix} \overset{V_t}{\begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix}}$$

Example: Stock Value of a Firm



- Given $\boldsymbol{P}(V_1)$ and $\boldsymbol{P}(V_t|V_{t-1})$, What is $\boldsymbol{P}(V_2)$?



$$P(V_2 = high) = \sum_{v_1} P(V_2 = high, V_1 = v_1)$$

$$= \sum_{v_1} P(V_2 = high|V_1 = v_1)P(V_1 = v_1)$$

$$= P(V_2 = high|V_1 = low)P(V_1 = low) + P(V_2 = high|V_1 = high)P(V_1 = high)$$

$$= 0.3 * 0 + 0.9 * 1.0 = 0.9$$

$$P(V_2 = low) = \sum_{v_1} P(V_2 = low, V_1 = v_1)$$

$$= \sum_{v_1} P(V_2 = low|V_1 = v_1)P(V_1 = v_1)$$

$$= P(V_2 = low|V_1 = low)P(V_1 = low) + P(V_2 = low|V_1 = high)P(V_1 = high)$$

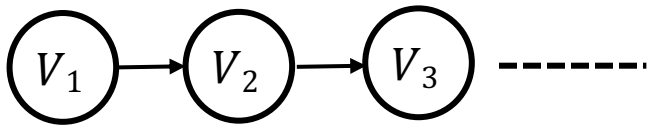
$$= 0.7 * 0 + 0.1 * 1.0 = 0.1$$

$$P(V_2 = low) = 1 - P(V_2 = high)$$
$$= 0.1$$

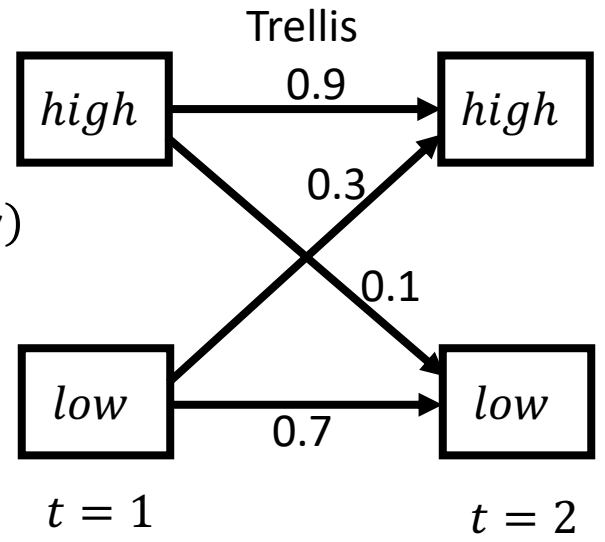
$$\begin{matrix} & V_t \\ V_{t-1} & \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} \end{matrix}$$

In matrix-vector product form: $\boldsymbol{P}(V_2) = \boldsymbol{T}^T \boldsymbol{P}(V_1)$ (derivation in next page)

Example: Stock Value of a Firm



- Given $\boldsymbol{P}(V_1)$ and $\boldsymbol{P}(V_t|V_{t-1})$, What is $\boldsymbol{P}(V_2)$?



$$P(V_2 = high) = P(V_2 = high|V_1 = high)P(V_1 = high) + P(V_2 = high|V_1 = low)P(V_1 = low)$$

$$P(V_2 = low) = P(V_2 = low|V_1 = high)P(V_1 = high) + P(V_2 = low|V_1 = low)P(V_1 = low)$$

$$\begin{bmatrix} P(V_2 = high) \\ P(V_2 = low) \end{bmatrix} = \begin{bmatrix} P(V_2 = high|V_1 = high) & P(V_2 = high|V_1 = low) \\ P(V_2 = low|V_1 = high) & P(V_2 = low|V_1 = low) \end{bmatrix} \begin{bmatrix} P(V_1 = high) \\ P(V_1 = low) \end{bmatrix}$$

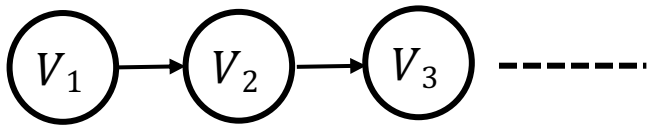
$$\boldsymbol{P}(V_2) = \boldsymbol{T}^T \boldsymbol{P}(V_1)$$

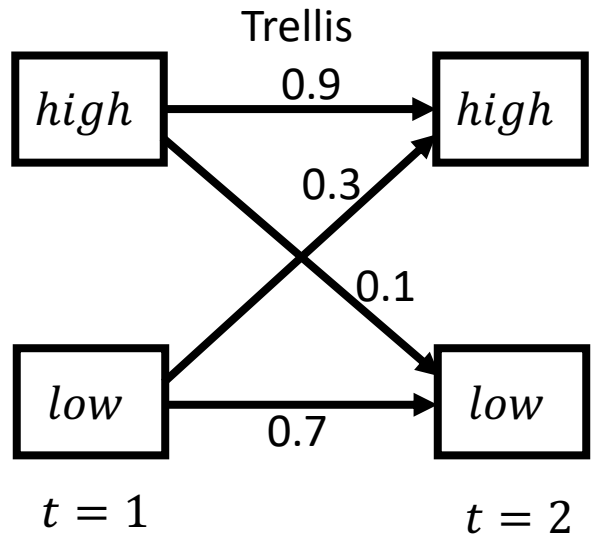
$$where\ \boldsymbol{P}(V_2) = \begin{bmatrix} P(V_2 = high) \\ P(V_2 = low) \end{bmatrix}$$

$$\boldsymbol{P}(V_1) = \begin{bmatrix} P(V_1 = high) \\ P(V_1 = low) \end{bmatrix}$$

$$\boldsymbol{T} = \begin{bmatrix} P(V_2 = high|V_1 = high) & P(V_2 = low|V_1 = high) \\ P(V_2 = high|V_1 = low) & P(V_2 = low|V_1 = low) \end{bmatrix}$$

Example: Stock Value of a Firm



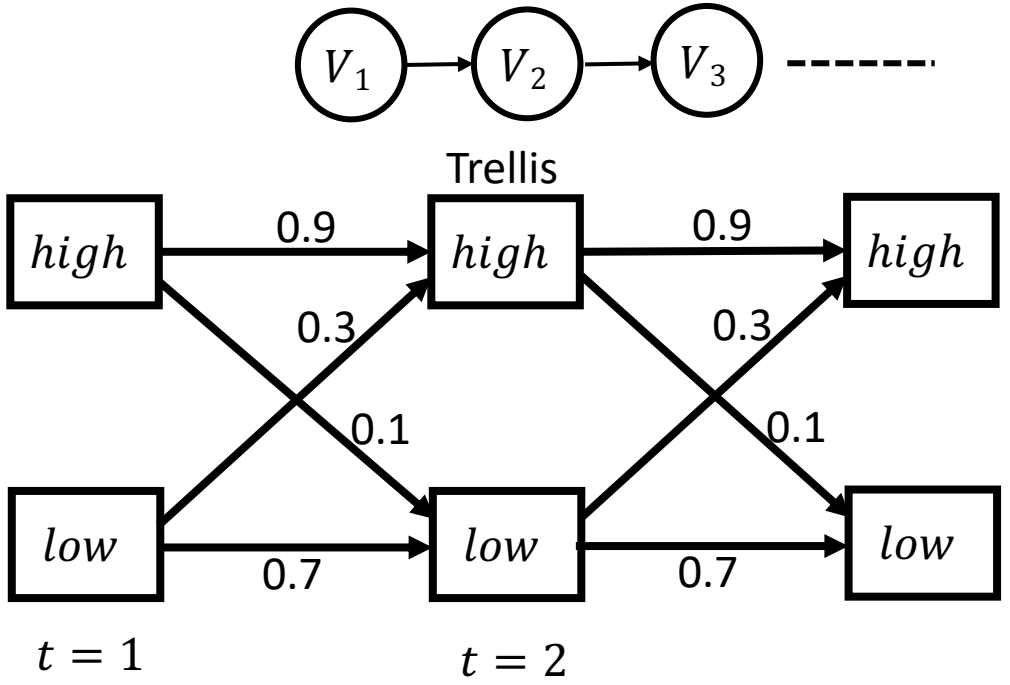


- Given $\boldsymbol{P}(V_1)$ and $\boldsymbol{P}(V_t|V_{t-1})$, What is $\boldsymbol{P}(V_3)$?
- Query Variable: V_3
- Evidence Variables: None
- Relevant Variables: V_1 , V_2, V_3
- Irrelevant Variables: $V_t, t > 3$
- Factorization: $P(V_1, V_2, V_3) = P(V_1)P(V_2|V_1)P(V_3|V_2)$
- Variables to Marginalize: V_1, V_2

$$\begin{matrix} & V_t \\ V_{t-1} & \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} \end{matrix}$$

Example: Stock Value

- What is $\boldsymbol{P}(V_3)$?
  - We know $\boldsymbol{P}(V_1)$ and $\boldsymbol{P}(V_t|V_{t-1})$



$$P(V_3 = h) = \sum_{v_1}\sum_{v_2} P(V_3 = h, V_2 = v_2, V_1 = v_1)$$

$$= \sum_{v_1}\sum_{v_2} P(V_1 = v_1)P(V_2 = v_2|V_1 = v_1)P(V_3 = h|V_2 = v_2)$$

$$= \sum_{v_2} P(V_3 = h|V_2 = v_2) \sum_{v_1} P(V_1 = v_1)P(V_2 = v_2|V_1 = v_1)$$

$$= \sum_{v_2} P(V_3 = h|V_2 = v_2)\, P(V_2 = v_2)$$

$$= 0.9 * 0.9 + 0.3 * 0.1 = 0.84$$

$$P(V_3 = low) = 1 - P(V_3 = high) = 0.16$$

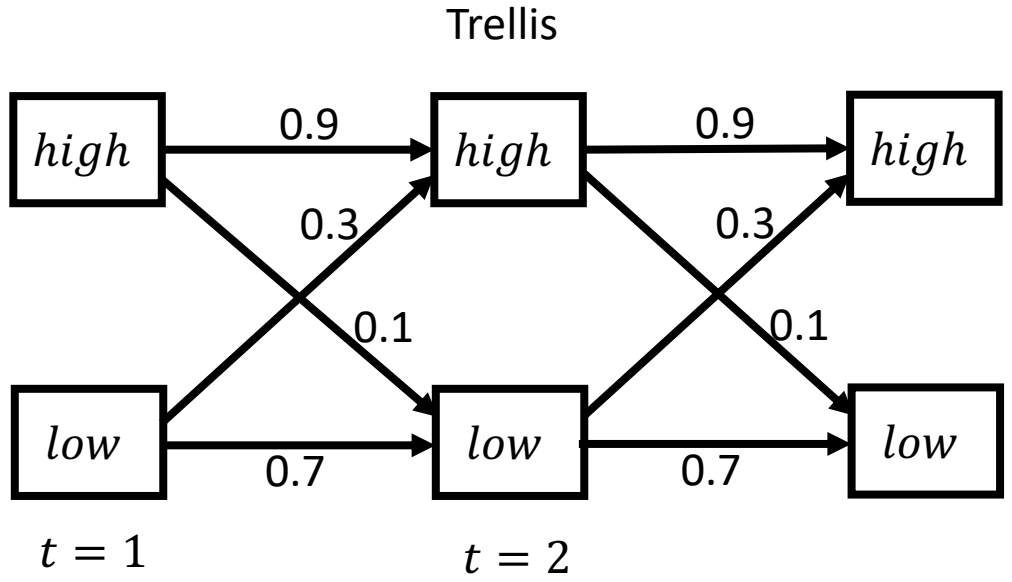
$$V_{t-1} \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} \quad V_t$$

$$P(V_2 = v_2) = \sum_{v_1} P(V_1 = v_1)P(V_2 = v_2|V_1 = v_1)$$

We can write this in Matrix-Vector product form

Example: Stock Value

- What is $\boldsymbol{P}(V_3)$?
  - We know $\boldsymbol{P}(V_1)$ and $\boldsymbol{P}(V_t|V_{t-1})$



$$P(V_3 = h) = P(V_3 = h|V_2 = h)P(V_2 = h) + P(V_3 = h|V_2 = l)P(V_2 = l)$$

$$P(V_3 = l) = P(V_3 = l|V_2 = h)P(V_2 = h) + P(V_3 = l|V_2 = l)P(V_2 = l)$$

$$\begin{bmatrix} P(V_3 = high) \\ P(V_3 = low) \end{bmatrix} = \begin{bmatrix} P(V_3 = h|V_2 = h) & P(V_3 = h|V_2 = l) \\ P(V_3 = l|V_2 = h) & P(V_3 = l|V_2 = l) \end{bmatrix} \begin{bmatrix} P(V_2 = h) \\ P(V_2 = l) \end{bmatrix}$$

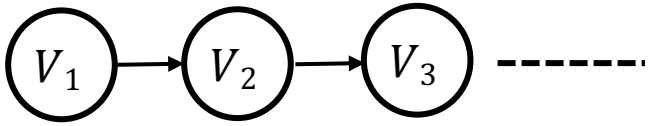
$$\boldsymbol{P}(V_3) = \boldsymbol{T}^T \boldsymbol{P}(V_2)$$

$$= \boldsymbol{T}^T \boldsymbol{T}^T \boldsymbol{P}(V_1)$$

$$= (\boldsymbol{T}^T)^2 \boldsymbol{P}(V_1)$$

$$V_{t-1} \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} \quad V_t$$

Example: Stock Value



- Given $\boldsymbol{P}(V_1)$, what is $\boldsymbol{P}(V_n)$?

$$\boldsymbol{P}(V_n) = \boldsymbol{T}^T \boldsymbol{P}(V_{n-1})$$

$$= \boldsymbol{T}^T \boldsymbol{T}^T \boldsymbol{P}(V_{n-2})$$

$$\vdots$$

$$= \underbrace{\boldsymbol{T}^T \ \boldsymbol{T}^T \ \ldots \boldsymbol{T}^T}_{(n-1) \text{ terms}} \ \boldsymbol{P}(V_1)$$

$$= (\boldsymbol{T}^T)^{n-1} \boldsymbol{P}(V_1)$$

Example: Stock Value

- From initial observation of $high$:

$$\underset{\boldsymbol{P}(V_1)}{\begin{bmatrix}1.0\\0.0\end{bmatrix}} \quad \underset{\boldsymbol{P}(V_2)}{\begin{bmatrix}0.9\\0.1\end{bmatrix}} \quad \underset{\boldsymbol{P}(V_3)}{\begin{bmatrix}0.84\\0.16\end{bmatrix}} \quad \underset{\boldsymbol{P}(V_4)}{\begin{bmatrix}0.804\\0.196\end{bmatrix}} \longrightarrow \underset{\boldsymbol{P}(V_\infty)}{\begin{bmatrix}0.75\\0.25\end{bmatrix}}$$

- From initial observation of low:

$$\underset{\boldsymbol{P}(V_1)}{\begin{bmatrix}0.0\\1.0\end{bmatrix}} \quad \underset{\boldsymbol{P}(V_2)}{\begin{bmatrix}0.3\\0.7\end{bmatrix}} \quad \underset{\boldsymbol{P}(V_3)}{\begin{bmatrix}0.48\\0.52\end{bmatrix}} \quad \underset{\boldsymbol{P}(V_4)}{\begin{bmatrix}0.588\\0.412\end{bmatrix}} \longrightarrow \underset{\boldsymbol{P}(V_\infty)}{\begin{bmatrix}0.75\\0.25\end{bmatrix}}$$

- From some arbitrary initial distribution:

$$\underset{\boldsymbol{P}(V_1)}{\begin{bmatrix}p\\1-p\end{bmatrix}} \ \ldots \longrightarrow \underset{\boldsymbol{P}(V_\infty)}{\begin{bmatrix}0.75\\0.25\end{bmatrix}}$$

Matrix Representation

- State Transition Matrix $\boldsymbol{T} = \begin{bmatrix} T_{11} & \cdots & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & \cdots & T_{nn} \end{bmatrix}$, where $T_{ij} = P(X_n = j | X_{n-1} = i)$

- Prior (or) Initial Distribution: $\boldsymbol{P}(X_1)$

$$\boldsymbol{P}(X_2) = \boldsymbol{T}^T \boldsymbol{P}(X_1)$$

$$\boldsymbol{P}(X_3) = \boldsymbol{T}^T \boldsymbol{P}(X_2) = (\boldsymbol{T}^T)^2 \boldsymbol{P}(X_1)$$

$$\boldsymbol{P}(X_4) = \boldsymbol{T}^T \boldsymbol{P}(X_3) = (\boldsymbol{T}^T)^3 \boldsymbol{P}(X_1)$$

$$\vdots$$

$$\boldsymbol{P}(X_{n+1}) = \boldsymbol{T}^T \boldsymbol{P}(X_n) = (\boldsymbol{T}^T)^n \boldsymbol{P}(X_1)$$

Stationary Distribution: $\boldsymbol{P}(X_\infty)$ or $\boldsymbol{P}_\infty(X)$

- Stationary distribution is independent of initial distribution*
  - Describes the long-term behavior of the state
- Properties of stationary distribution:
  - $\boldsymbol{P}(X_\infty) = \boldsymbol{T}^T \boldsymbol{P}(X_\infty)$ $-\ Equation\ I$
  - $\sum_x P_\infty(X) = 1$ $-\ Equation\ II$

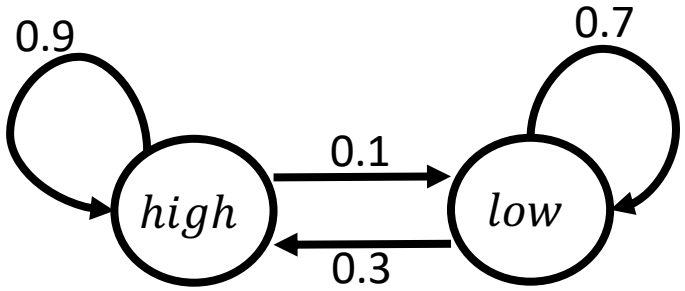
$Equation\ I$

$$\boldsymbol{P}(X_\infty) = \boldsymbol{T}^T \boldsymbol{P}(X_\infty)$$

Stationary distribution is an eigen vector of transpose of transition matrix ($\boldsymbol{T}^T$)

*There are Markov chains which do not have this property, but this is out of scope

Example: Stationary Distribution



From Equation I:

- $P_\infty(high) = P(high|high)P_\infty(high) + P(high|low)P_\infty(low)$

$$= 0.9 * P_\infty(high) + 0.3 * P_\infty(low)$$

From Equation II:

- $\sum_v P_\infty(V) = 1 \Longrightarrow P_\infty(high) + P_\infty(low) = 1$

$$\Longrightarrow P_\infty(high) = \frac{3}{4} \; and \; P_\infty(low) = \frac{1}{4}$$

$$P_\infty(low) = P(X_\infty = low)$$
$$P_\infty(high) = P(X_\infty = high)$$

Applications of Stationary Distribution

- Diffusion Models
  - Simulates Markov chain to reach stationary distribution
  - Eg: Text-based Image/Art Generation (Dall E-2, Stable Diffusion)

- Page Rank
  - Stationary distributions over the webpages (states) is used to rank them
    - Higher ranks for pages with high reachability
    - Since more links go to important pages, it is robust
    - Eg: Google 0.01 returned pages that contain all your keywords in a decreasing rank
  - Now a days all search engines use link analysis along with many other factors
    - rank is less significant

Dynamic Bayesian Networks

- Two architectures
  - Hidden Markov Models (HMM) – today
  - Markov Decision Processes (MDP) – later

Hidden Markov Models

Introduction

Applications

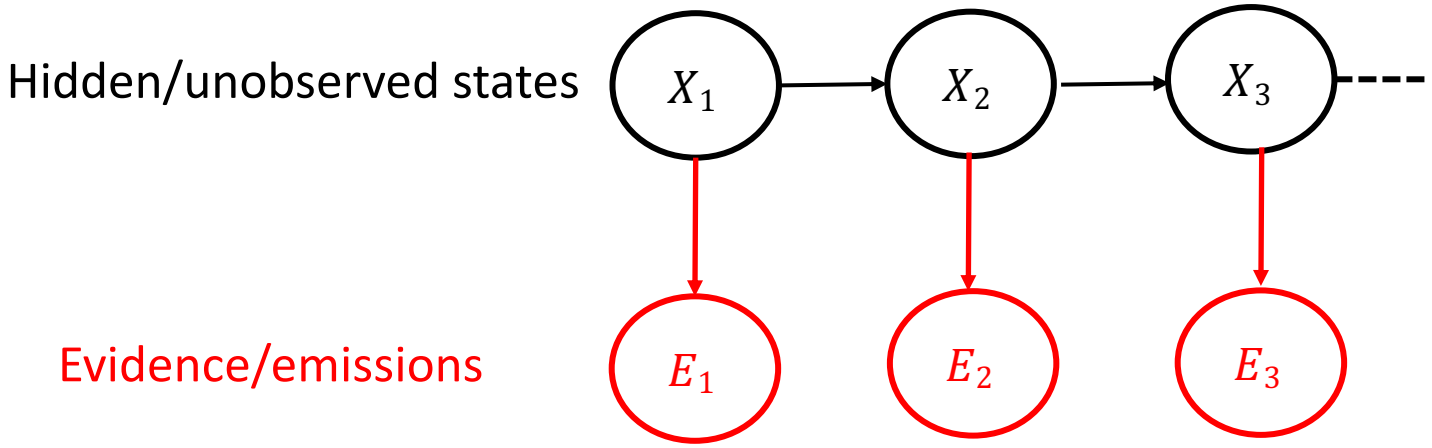
Examples

Joint Probability Distribution

Filtering

Hidden Markov Models

- Assumption:
  - Noisy version of state is observed



X_t: State of the environment at time t
E_t: Evidence (observation) of the environment at time t
$X_{1:T}$: States from $t = 1$ to $t = T$
$E_{1:T}$: Evidence from $t = 1$ to $t = T$

HMM Applications

Application	Observation	Hidden State
Language Translation	Source language	Target language
Speech-to-Text	Audio samples	Phonemes or words
Music Transcription	Audio sample of music	Sequence of musical notes or chords
Handwritten Text Recognition	Images of text	Characters in handwritten text
Parts-of-Speech Tagging	Words in a sentence	POS Tag (Verb, noun, etc.)
Gene Prediction	DNA sequence	Gene states (coding regions, non-coding regions, etc.)
Stock Market Analysis	Stock writer or Finance indices	Market conditions or states (bull market, bear market, etc.)
Localization	Sensor readings	Actual position in the environment
Object Tracking in Videos	Positional coordinates and Object features	Actual coordinates of the object and object identity

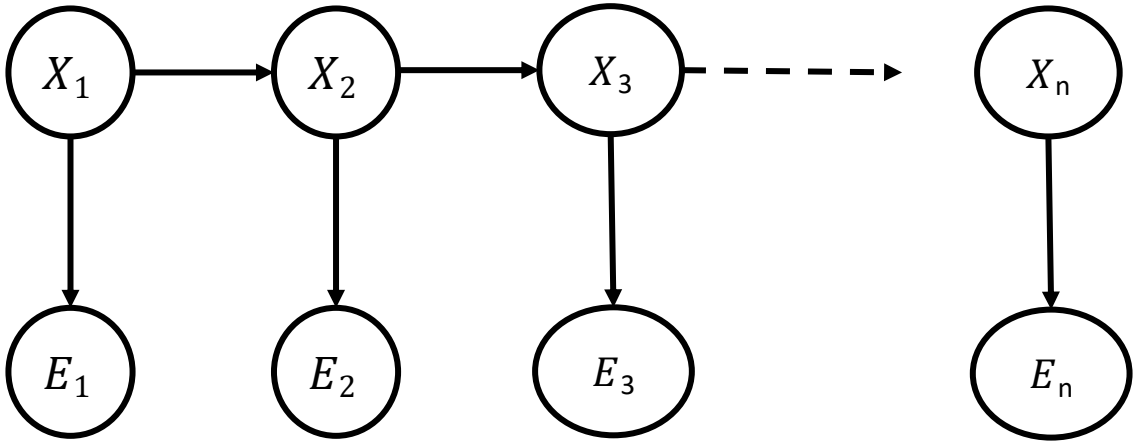
HMMs as Bayes Nets

- Bayesian networks represent the world at a given point in time or one snapshot of the world

- HMMs model the evolution over time by factoring in additional evidence
  - A special case of Bayesian networks that evolve over time

- All the semantic properties of BNs apply in HMMs as well
  - Local Semantics
    - Conditional independence
    - Markov Blanket
  - Global Semantics
    - D-Separation Properties

HMM: Notations



- Let $\mathcal{S} = \{1,2, \ldots, |S|\}$ be the set of states and $X_t \in \mathcal{S}$
- Let $\mathcal{O}$ be the set of output symbols and $E_t \in \mathcal{O}$
- Let $T_{ij} = P(X_t = j | X_{t-1} = i)$ be the transition probability
- Let $O_{io} = P(E_t = o | X_t = i)$ indicates the probability of output o in state i.

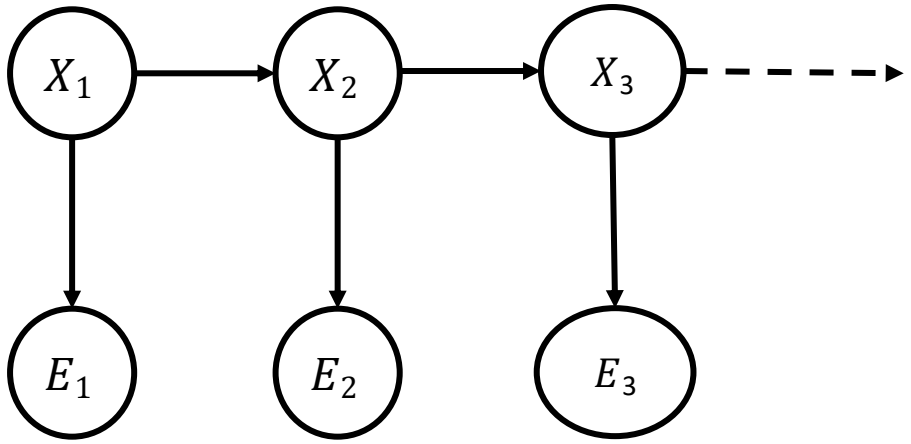
HMM: Stock Market Example

Set of States ($\mathcal{S}$): Stock Value

$$X \in \mathcal{S} = \{high, low\}$$

Evidence ($\mathcal{O}$): Buy or Sell indicator

$$E \in \mathcal{O} = \{buy, sell\}$$



Initial:

$$P(X_1 = high) = 1$$

$$\boldsymbol{P}(X_1) = \begin{bmatrix} P(X_1 = high) \\ P(X_1 = low) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Transition

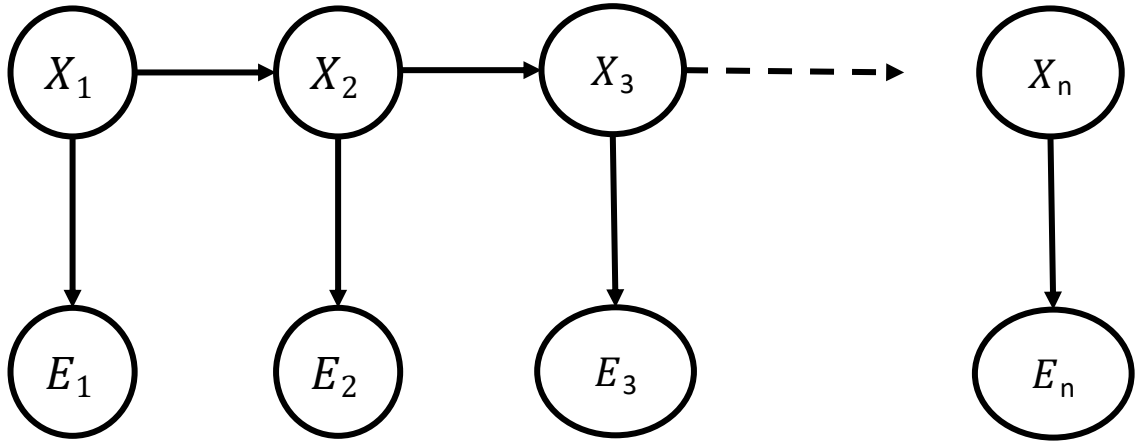
X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
high	*high*	0.9
high	*low*	0.1
low	*high*	0.3
low	*low*	0.7

Emission

X_t	E_t	$P(E_t \mid X_t)$
high	*buy*	0.1
high	*sell*	0.9
low	*buy*	0.8
low	*sell*	0.2

HMM Example: POS Tagging in NLP

- Set of states ($\mathcal{S}$): Parts-of-Speech
  - Eg: $X \in \mathcal{S} = \{N, V, A\}$
- Evidence: Vocabulary
  - $Eg: E \in \mathcal{O} = \{amazing, is, Markov\}$
- Parameters:
  - $T_{ij}, O_{io}, \pi_j = T_{0j}$



π_j: Initial Probability of State j

I run every morning
verb
I went for a morning run
noun

HMM: Applications

- HMM-like models are found to perform the named entity recognition (NER) task better than GPT-like models in financial tasks

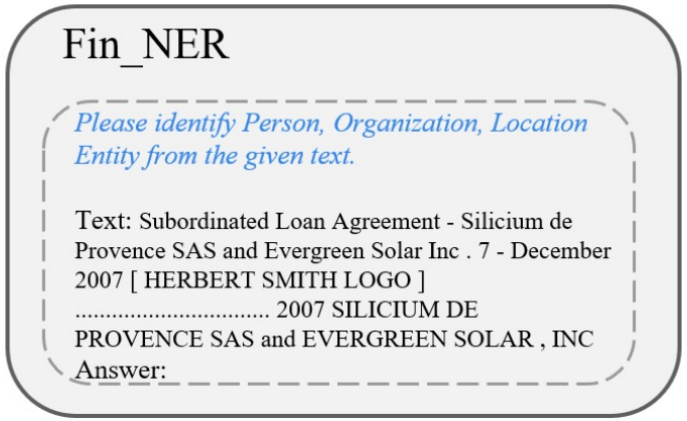


Figure 9: prompt for NER dataset

Model	Entity F1
ChatGPT(0)	29.21
ChatGPT(20)	51.52
GPT-4(0)	36.08
GPT-4(20)	56.71
BloombergGPT(20)	60.82
GPT-NeoX(20)	60.98
OPT66B(20)	57.49
BLOOM176B(20)	55.56
CRF(CoNLL)	17.20
CRF(FIN5)	**82.70**

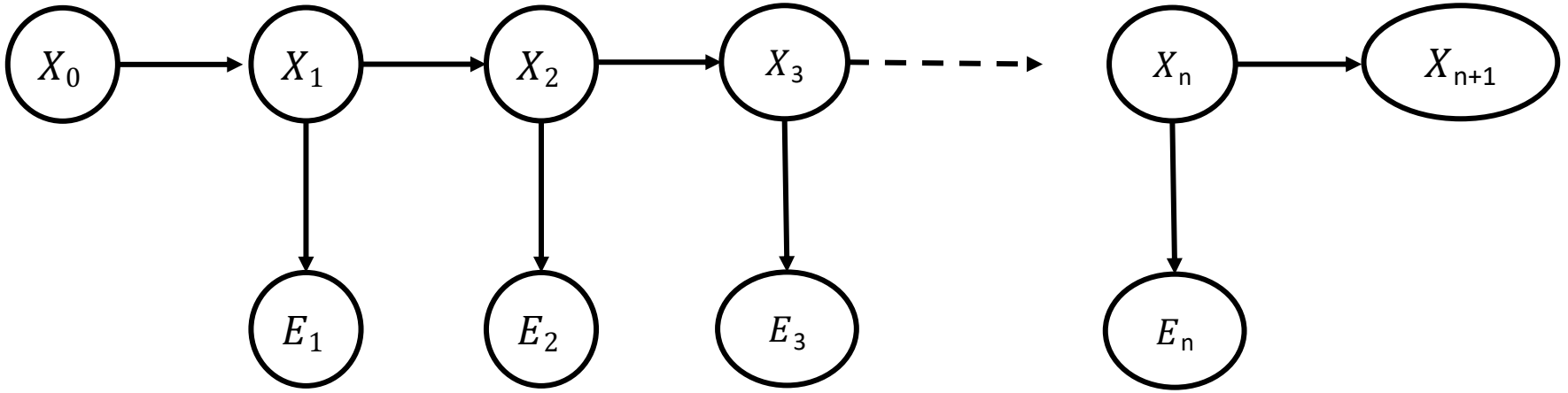
Table 6: Results of few-shot performance on the NER dataset. CRF(CoNLL) refers to CRF model that is trained on general CoNLL data, CRF(FIN5) refers to CRF model that is trained on FIN5 data. Again, we choose the same shot as BloombergGPT for fair comparison. More detailed experiments using 5 to 20 shots can be found in Appendix C.

CRFConditional Random Fileds

https://aclanthology.org/2023.emnlp-industry.39.pdf

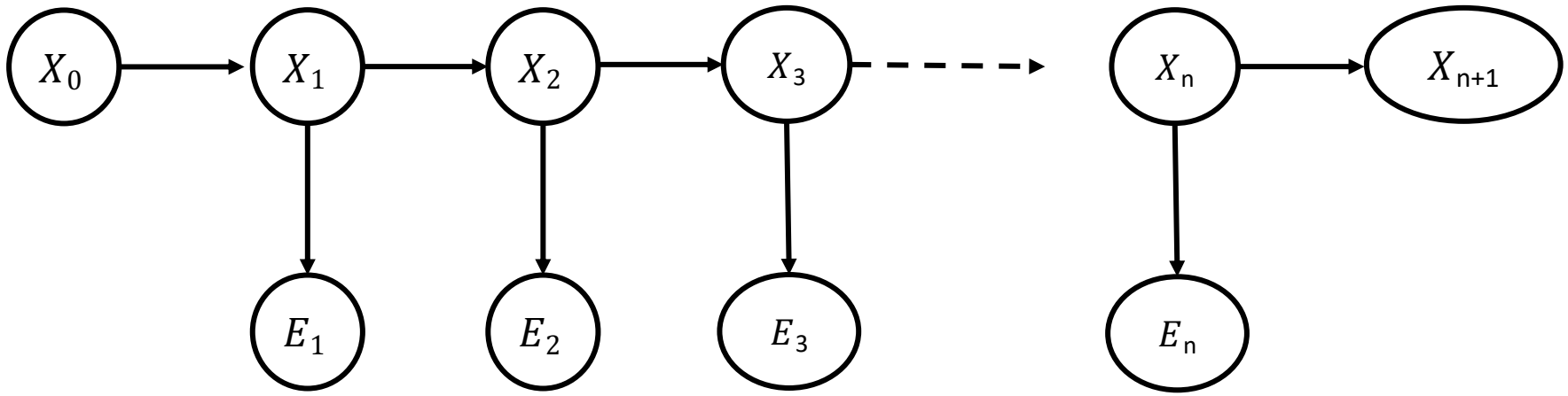
Joint Probability Distribution of HMM

- Include extra RVs $X_0 = START$ and $X_{n+1} = STOP$ for completeness
  - Note: We can also start with X_1 and end with X_n. We are just using these as a placeholders for better clarity



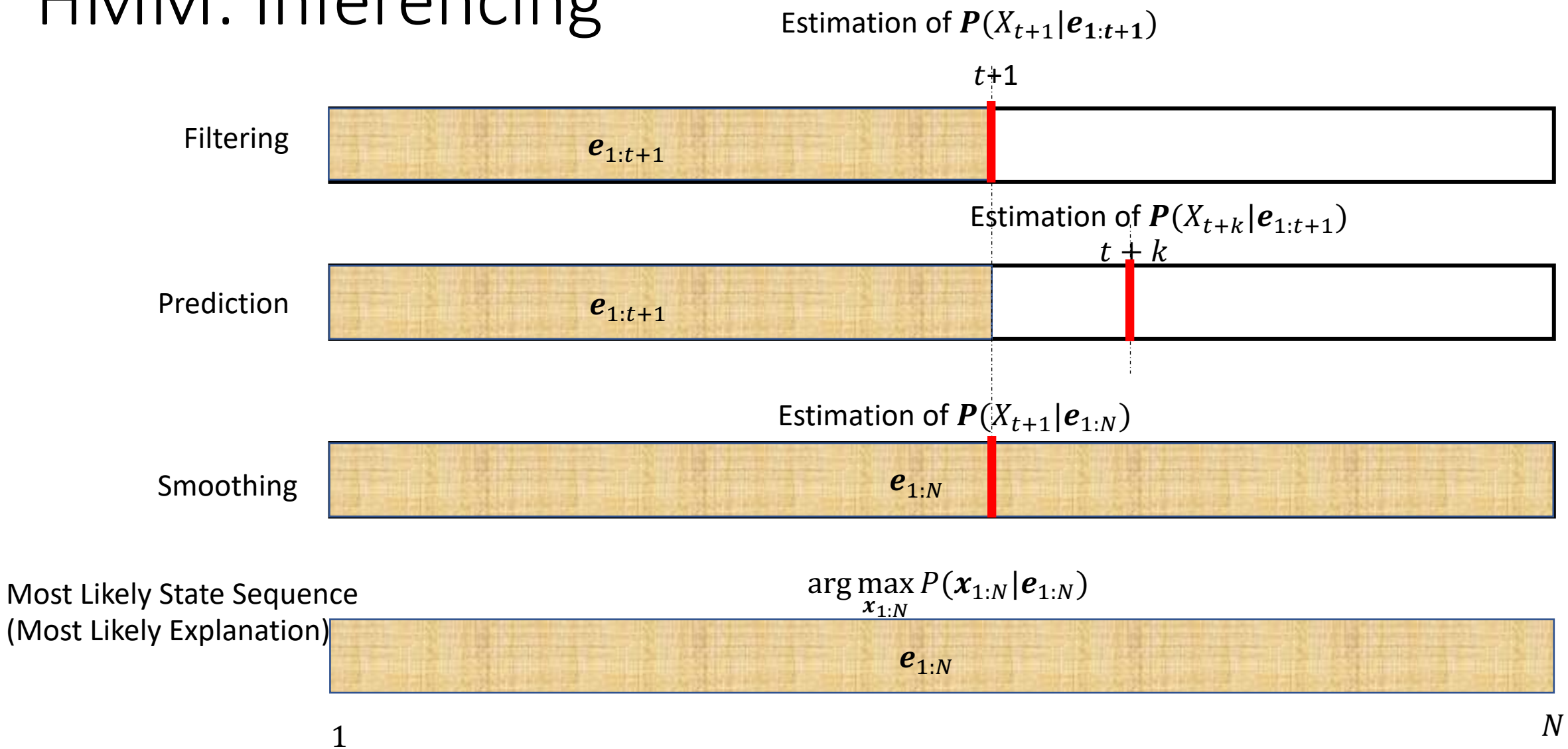
  - We are interested in deriving $\boldsymbol{P}(E_1, \ldots, E_n, X_0, X_1, \ldots, X_n, X_{n+1})$

Joint Probability Distribution of HMM



$$\boldsymbol{P}(E_1, \ldots, E_n, X_0, X_1, \ldots, X_n, X_{n+1}) = \underbrace{\boldsymbol{P}(X_0)}_{\text{Initial Probability}} \underbrace{\prod_{i=1}^{n+1} \boldsymbol{P}(X_i|X_{i-1})}_{\text{Transition Distribution}} \underbrace{\prod_{i=1}^{n} \boldsymbol{P}(E_i|X_i)}_{\text{Emission Distribution}}$$

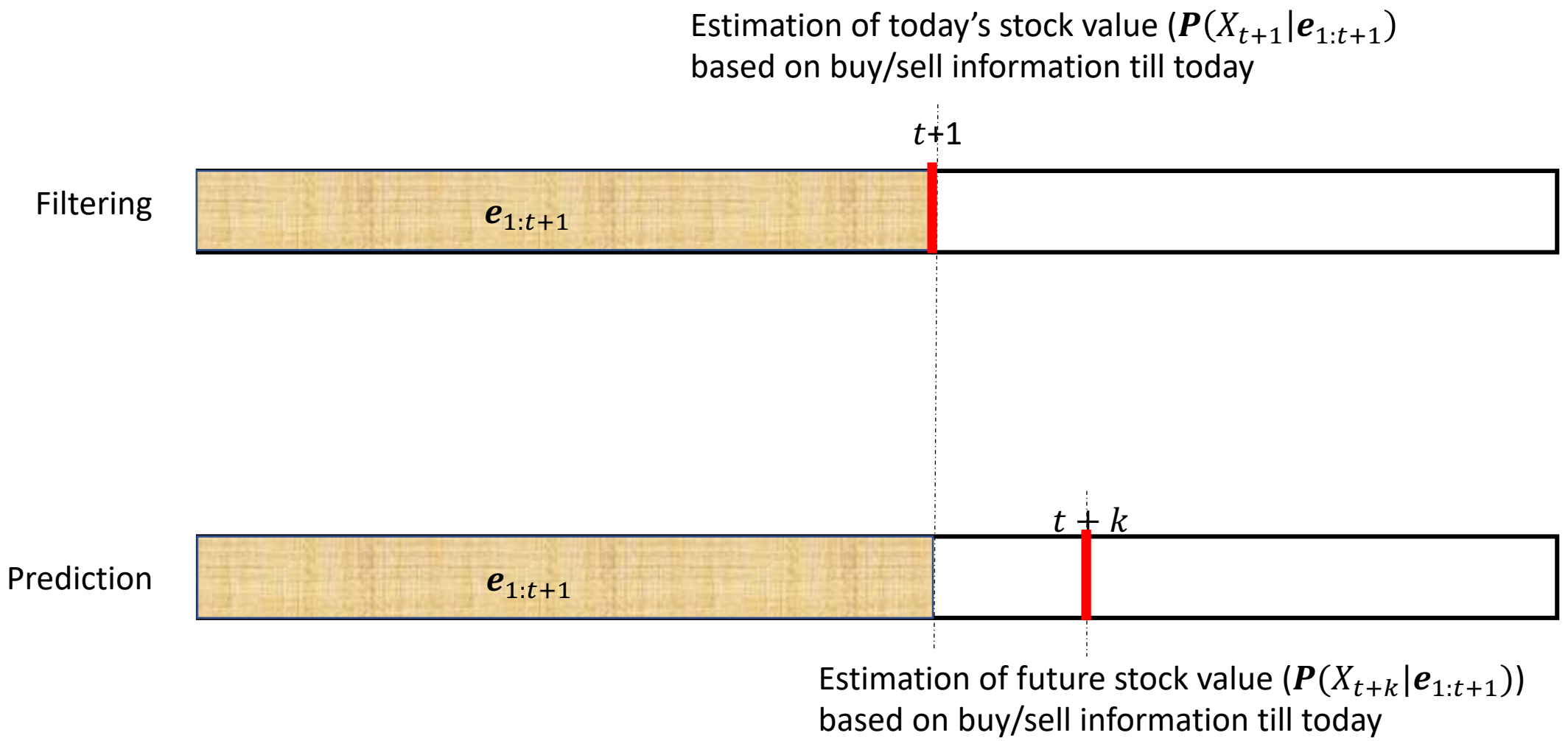
HMM: Inferencing



$\boldsymbol{x}_{1:N}$: States from $t = 1$ to $t = N$

$\boldsymbol{e}_{1:N}$: Evidence from $t = 1$ to $t = N$

HMM Inferencing: Stock Market Example



$\boldsymbol{x}_{1:N}$: States from $t = 1$ to $t = N$
$\boldsymbol{e}_{1:N}$: Evidence from $t = 1$ to $t = N$

HMM Inferencing: Filtering

Proof in subsequent slides

- **Objective**:
  - Calculation of $B(X_{t+1}) = P(X_{t+1}|\boldsymbol{e}_{1:t+1})\ \forall t$
- **Solution:** Recursive Filtering

$$\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

α : Normalization constant

$$\boldsymbol{f}_{1:t+1} = \begin{bmatrix} B(X_{t+1}=1) \\ \vdots \\ B(X_{t+1}=|S|) \end{bmatrix} = \boldsymbol{P}(X_{t+1}|\boldsymbol{e}_{1:t+1})$$

Probability distribution of X_{t+1}, given the evidence $\boldsymbol{e}_{1:t+1}$

$$\boldsymbol{f}_{1:t} = \begin{bmatrix} B(X_t=1) \\ \vdots \\ B(X_t=|S|) \end{bmatrix} = \boldsymbol{P}(X_t|\boldsymbol{e}_{1:t})$$

Probability distribution of X_t, given the evidence $\boldsymbol{e}_{1:t}$

$$\boldsymbol{O}_{t+1} = \boldsymbol{diag}([\mathbf{P}(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}=1) \quad \dots \quad \mathbf{P}(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}=|S|)])$$

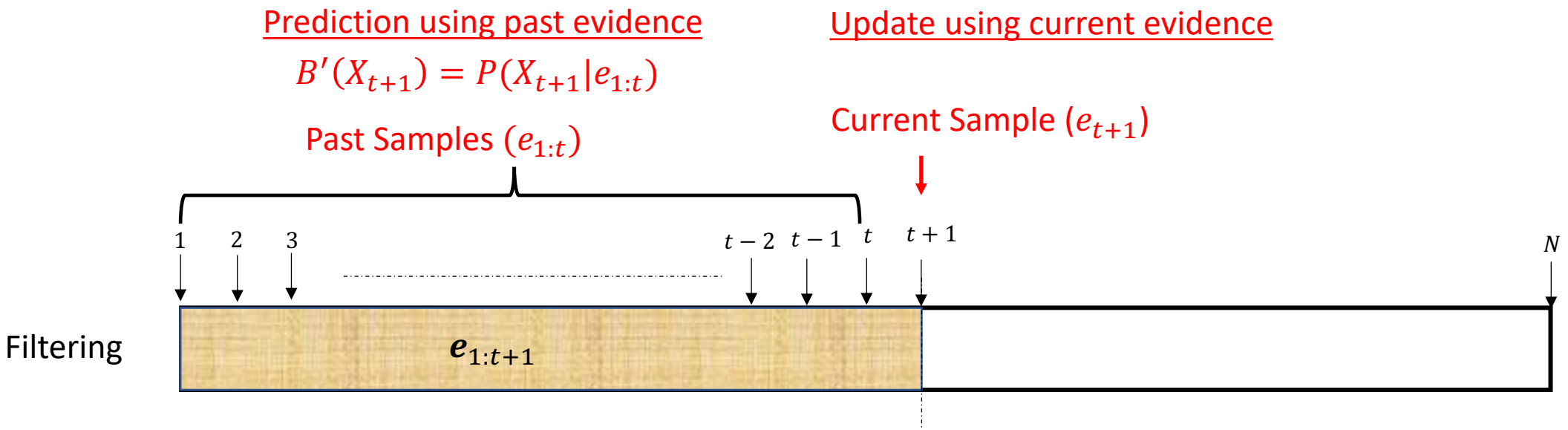
Observation matrix

$$\boldsymbol{T} = \begin{bmatrix} T_{11} & \dots & T_{1|S|} \\ \vdots & \ddots & \vdots \\ T_{|S|1} & \dots & T_{|S||S|} \end{bmatrix}, T_{ij} = \mathbf{P}(\boldsymbol{X}_{t+1}=j|\boldsymbol{X}_t=i)$$

Transition Matrix

Bayesian Recursive Filtering

- **Objective**:
  - Calculation of $B(X_{t+1}) = P(X_{t+1}|e_{1:t+1})\ \forall t$
- **Key Idea**: Prediction + Update/Correction
  - Predict X_{t+1} using the evidence up to t, i.e., $P(X_{t+1}|e_{1:t})$. Let us call this $B'(X_{t+1})$
    - $B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$
  - Update the prediction with the evidence at $t+1$ (i.e., e_{t+1})



Bayesian Recursive Filtering

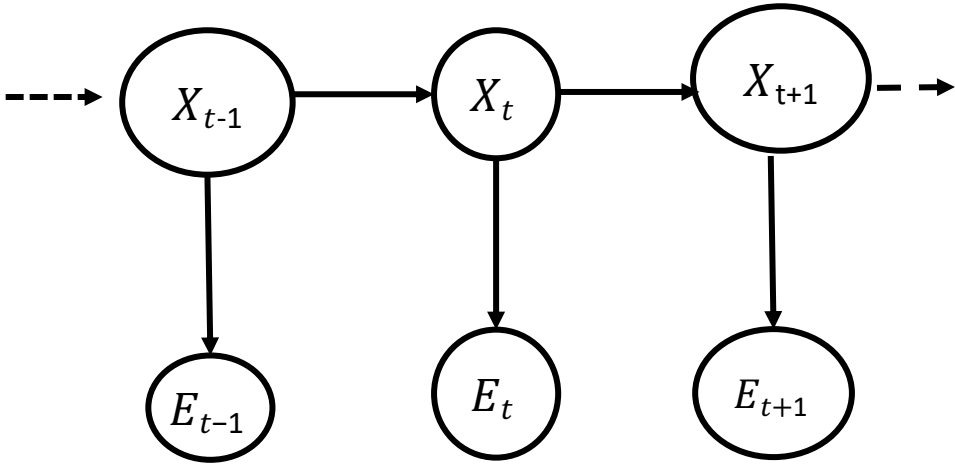
- **Objective**: Derive the following recursive relation
  - $\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$

- Recursion: $\mathbf{P}(X_{t+1}|\boldsymbol{e}_{1:t+1}) = \mathrm{f}(\boldsymbol{e}_{t+1}, \mathbf{P}(X_t|\boldsymbol{e}_{1:t}))$
  - Base case: $\boldsymbol{P}(X_0)$
  - We assume that $\boldsymbol{P}(X_t|e_{1:t})$ is available and use this fact to calculate $\boldsymbol{P}(X_{t+1}|e_{1:t+1})$

- Use the Bayes Rule
  - $P(A, B|C) = P(A|B, C)P(B|C)$

- HMM Provides the following parameters:
  - $P(X_0)$
  - $P(e_t|x_t)$
  - $P(X_t|X_{t-1})$

Bayesian Recursive Filtering



$$\mathbf{P}(X_{t+1}|\boldsymbol{e}_{1:t+1}) = \boldsymbol{P}(X_{t+1}|e_{t+1}, \boldsymbol{e}_{1:t})$$

$$= \frac{\boldsymbol{P}(X_{t+1}, e_{t+1}|\boldsymbol{e}_{1:t})}{P(e_{t+1}|\boldsymbol{e}_{1:t})}$$

$\because P(A|B,C) = \frac{P(A,B|C)}{P(B|C)}, where\ A = X_{t+1}, B = e_{t+1,}\ C = \boldsymbol{e}_{1:t}$

$$= \alpha\, \boldsymbol{P}(X_{t+1}, e_{t+1}|\boldsymbol{e}_{1:t})$$

$\alpha = \frac{1}{P(\boldsymbol{e}_{t+1}|\boldsymbol{e}_{1:t})}$ is the normalization constant

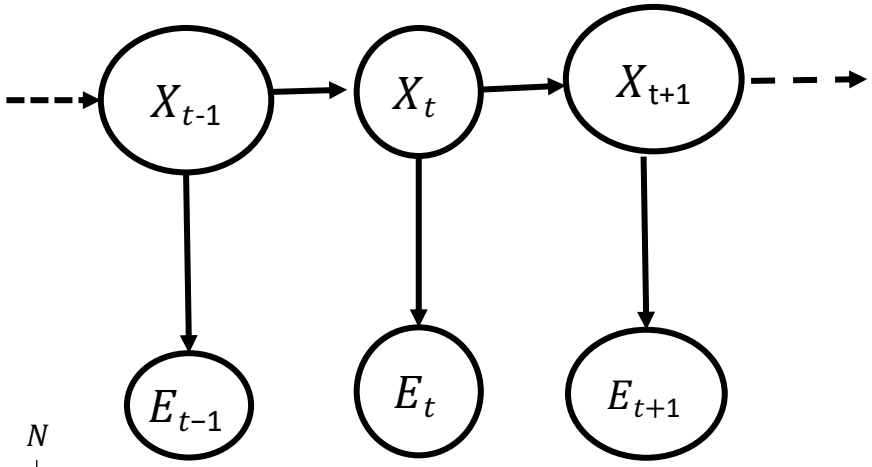
$$= \alpha\, \boldsymbol{P}(e_{t+1}|X_{t+1}, \boldsymbol{e}_{1:t})\boldsymbol{P}(X_{t+1}|\boldsymbol{e}_{1:t})$$

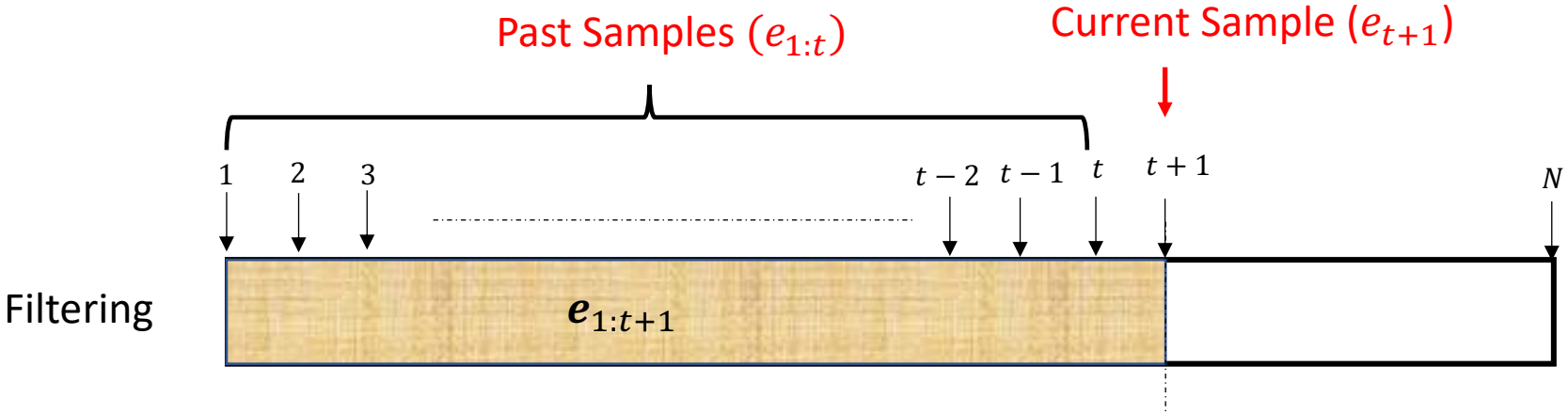
$\because P(A,B|C) = P(B|A,C)P(A|C); where\ A = X_{t+1}, B = e_{t+1,}\ C = \boldsymbol{e}_{1:t}$

$$= \alpha\, \boldsymbol{P}(e_{t+1}|X_{t+1})\boldsymbol{P}(X_{t+1}|\boldsymbol{e}_{1:t})$$

$\because e_{t+1}$ is independent of $\boldsymbol{e}_{1:t}$, given X_{t+1}

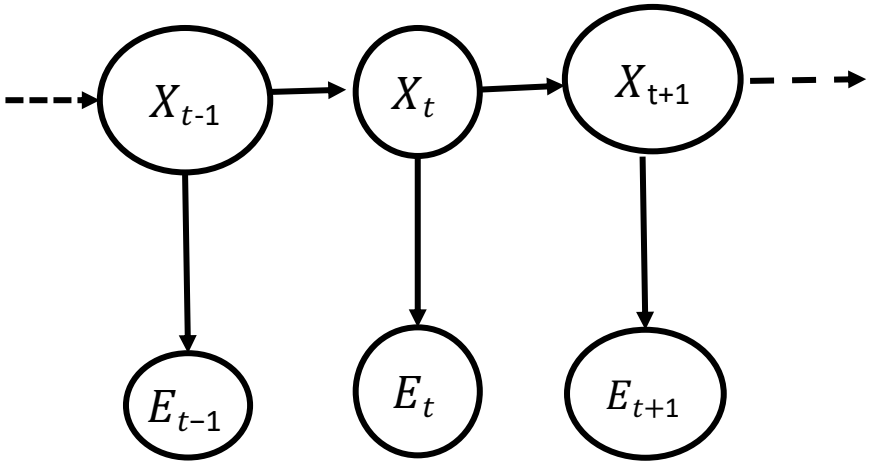
Bayesian Recursive Filtering





$$\mathbf{P}(X_{t+1}|\boldsymbol{e}_{1:t+1}) = \alpha\, \underbrace{\boldsymbol{P}(e_{t+1}|X_{t+1})}_{\text{Update using current sample (provided in the model)}} \underbrace{\boldsymbol{P}(X_{t+1}|\boldsymbol{e}_{1:t})}_{\text{Prediction using past samples}}$$

Bayesian Recursive Filtering



- **Calculation of $\boldsymbol{P}(X_{t+1}|\boldsymbol{e}_{1:t})$:**
  - Query Variable: X_{t+1}
  - Evidence: $E_1, E_2, \dots, E_{t-1}, E_t$
  - Relevant Variables for the Query:
    - $X_1, X_2, X_3, \dots, X_{t-1}, X_t, X_{t+1}, E_1, E_2, \dots, E_{t-1}, E_t$
  - Variables to Marginalize:
    - $X_1, X_2, X_3, \dots, X_{t-1}, X_t$

Bayesian Recursive Filtering

$$\mathbf{P}(X_{t+1}|\boldsymbol{e}_{1:t+1}) = \alpha\, \boldsymbol{P}(\boldsymbol{e}_{t+1}|X_{t+1})\boldsymbol{P}(X_{t+1}|\boldsymbol{e}_{1:t})$$

Marginalize unobserved relevant variables

$$= \alpha\, \boldsymbol{P}(e_{t+1}|\boldsymbol{X}_{t+1}) \sum_{x_t} \boldsymbol{P}(X_{t+1}, X_t = x_t|\boldsymbol{e}_{1:t})$$

Why is only X_t marginalized?
What about other relevant variables ($X_{1:t-1}$)?
Homework!!

$$= \alpha\, \boldsymbol{P}(e_{t+1}|X_{t+1}) \sum_{x_t} \boldsymbol{P}(X_{t+1}|X_t = x_t, \boldsymbol{e}_{1:t})\, \boldsymbol{P}(X_t = x_t|\boldsymbol{e}_{1:t})$$

$\because P(A, B|C) = P(A|B, C)P(B|C); where\ A = X_{t+1}, B = x_t, C = \boldsymbol{e}_{1:t}$

$$= \alpha\, \boldsymbol{P}(e_{t+1}|X_{t+1}) \sum_{x_t} \boldsymbol{P}(X_{t+1}|X_t = x_t)\, \boldsymbol{P}(X_t = x_t|\boldsymbol{e}_{1:t})$$

Sensor model | Transition model | Recursion

Message/belief propagated from state estimation at time t

$$P(A, B|C) = P(A|B, C)P(B|C)$$

Bayesian Recursive Filtering

- Belief/Message propagated from state estimation at time t:

$$\boldsymbol{P}(X_{t+1}|\boldsymbol{e}_{1:t+1}) = \alpha\, P(e_{t+1}|X_{t+1}) \sum_{\boldsymbol{x_t}} \boldsymbol{P}(X_{t+1}|X_t = x_t)\, P(X_t = x_t|\boldsymbol{e}_{1:t})$$

- Bayesian recursion in vector format:

$$\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

where $\boldsymbol{O}_{t+1} = \boldsymbol{diag}([P(e_{t+1}|X_{t+1} = 1) \quad \dots \quad P(e_{t+1}|X_{t+1} = |S|)])$

$$\boldsymbol{f}_{1:t+1} = \begin{bmatrix} B(X_{t+1} = 1) \\ \vdots \\ B(X_{t+1} = |S|) \end{bmatrix}, \boldsymbol{f}_{1:t} = \begin{bmatrix} B(X_t = 1) \\ \vdots \\ B(X_t = |S|) \end{bmatrix}, \boldsymbol{T} = \begin{bmatrix} T_{11} & \dots & T_{1|S|} \\ \vdots & \ddots & \vdots \\ T_{|S|1} & \dots & T_{|S||S|} \end{bmatrix}, T_{ij} = P(X_{t+1} = j|X_t = i)$$

$B(X_t = x_t) = P(X_t = x_t|\boldsymbol{e}_{1:t})$ is the belief that the state is x_t given the evidence up to t
$B(X_{t+1} = x_{t+1}) = P(X_{t+1} = x_{t+1}|\boldsymbol{e}_{1:t+1})$ is the belief that the state is x_{t+1} given the evidence up to $t+1$

Filtering: Example

$$\boldsymbol{P}(X_{t+1}|\boldsymbol{e}_{1:t+1}) = \alpha\, P(e_{t+1}|X_{t+1}) \sum_{\boldsymbol{x_t}} \boldsymbol{P}(X_{t+1}|X_t = x_t)\, P(X_t = x_t|\boldsymbol{e}_{1:t})$$

$$\boldsymbol{f}_{1:t+1} = \alpha\, \boldsymbol{O}\, \boldsymbol{T}^T\, \boldsymbol{f}_{1:t}$$

- Consider an HMM with two states, i.e., $\mathcal{S} = \{1,2\}$
- Message propagated from state estimation at time t:

$$B(X_{t+1} = 1) = \mathrm{P}(X_{t+1} = 1|\boldsymbol{e}_{1:t+1})$$

$$= \alpha\, P(e_{t+1}|X_{t+1} = 1) \sum_{\boldsymbol{x_t}\in\{1,2\}} P(X_{t+1} = 1|X_t = x_t)\, P(X_t = x_t|\boldsymbol{e}_{1:t})$$

$$= \alpha\, P(e_{t+1}|X_{t+1} = 1)\big(P(X_{t+1} = 1|X_t = 1)\, P(X_{\boldsymbol{t}} = 1|\boldsymbol{e}_{1:t}) + P(X_{t+1} = 1|X_t = 2)\, P(X_t = 2|\boldsymbol{e}_{1:t})\big)$$

$$= \alpha\, O_{\boldsymbol{11}}\big(T_{\boldsymbol{11}}\, P(X_t = 1|\boldsymbol{e}_{1:t}) + T_{\boldsymbol{21}}\, P(X_t = 2|\boldsymbol{e}_{1:t})\big)$$

$$= \alpha\, O_{\boldsymbol{11}}\big(T_{\boldsymbol{11}} B(X_t = 1) + T_{\boldsymbol{21}} B(X_t = 2)\big) \quad (1)$$

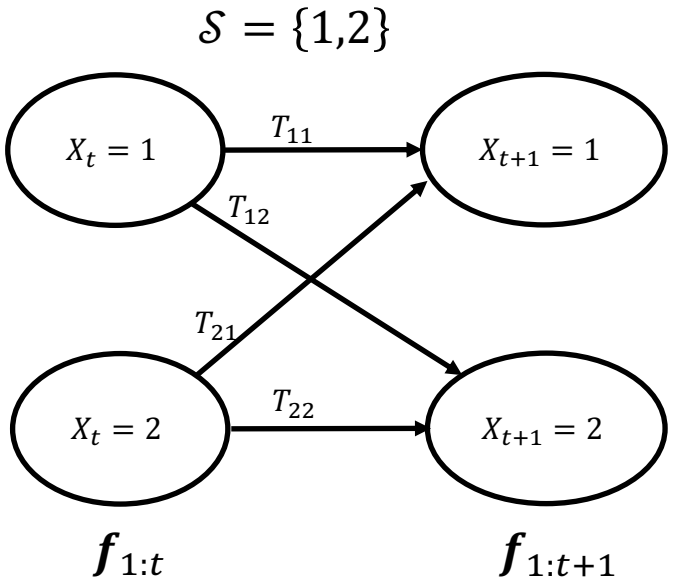
$$B(X_{t+1} = 2) = \mathrm{P}(X_{t+1} = 2|\boldsymbol{e}_{1:t+1})$$

$$= \alpha\, P(e_{t+1}|X_{t+1} = 2) \sum_{\boldsymbol{x_t}\in\{1,2\}} P(X_{t+1} = 2|X_t = x_t)\, P(X_t = x_t|\boldsymbol{e}_{1:t})$$

$$= \alpha\, P(e_{t+1}|X_{t+1} = 2)\big(P(X_{t+1} = 2|X_t = 1)\, P(X_{\boldsymbol{t}} = 1|\boldsymbol{e}_{1:t}) + P(X_{t+1} = 2|X_t = 2)\, P(X_t = 2|\boldsymbol{e}_{1:t})\big)$$

$$= \alpha\, O_{22}\big(T_{12}\, P(X_t = 1|\boldsymbol{e}_{1:t}) + T_{22}\, P(X_t = 2|\boldsymbol{e}_{1:t})\big)$$

$$= \alpha\, O_{22}\big(T_{12} B(X_t = 1) + T_{\boldsymbol{22}} B(X_t = 2)\big) \quad (2)$$



Bayesian Recursion: Example

$$B(X_{t+1}=1) = \alpha\, O_{11}\big(T_{11}B(X_t=1) + T_{21}B(X_t=2)\big) \quad (1)$$
$$B(X_{t+1}=2) = \alpha\, O_{22}\big(T_{12}B(X_t=1) + T_{22}B(X_t=2)\big) \quad (2)$$

$$\begin{bmatrix} B(X_{t+1}=1) \\ B(X_{t+1}=2) \end{bmatrix} = \alpha \begin{bmatrix} O_{11} & 0 \\ 0 & O_{22} \end{bmatrix} \begin{bmatrix} T_{11} & T_{21} \\ T_{12} & T_{22} \end{bmatrix} \begin{bmatrix} B(X_t=1) \\ B(X_t=2) \end{bmatrix}$$

$$\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

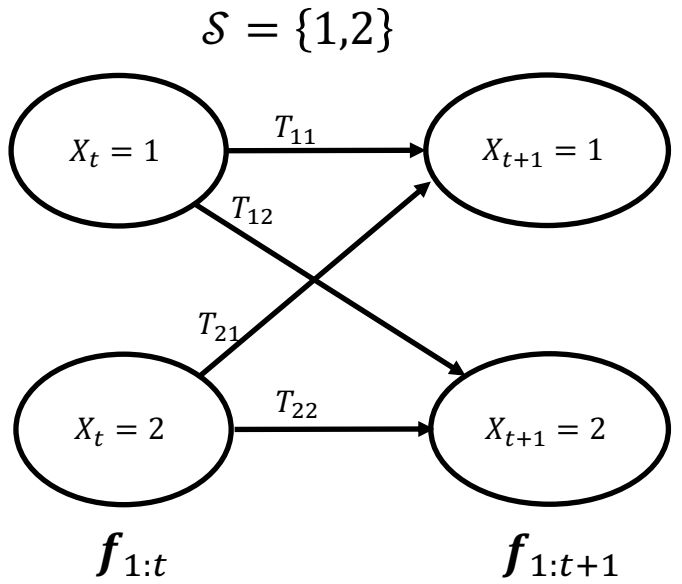
where $\boldsymbol{f}_{1:t+1} = \begin{bmatrix} B(X_{t+1}=1) \\ B(X_{t+1}=2) \end{bmatrix}$, $\boldsymbol{f}_{1:t} = \begin{bmatrix} B(X_t=1) \\ B(X_t=2) \end{bmatrix}$,

$$\boldsymbol{O}_{t+1} = \begin{bmatrix} O_{11} & 0 \\ 0 & O_{22} \end{bmatrix}$$

$$\boldsymbol{T} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}$$

$$O_{ii} = P(e_{t+1}|X_{t+1}=i)$$

$$T_{ij} = P(X_{t+1}=j|X_t=i)$$



Conditional probability distribution of state at time t given evidence up to time t

Conditional probability distribution of state at time $t+1$ given evidence up to time $t+1$

Bayesian Recursive Filtering: Forward Algorithm

- Let $\boldsymbol{f}_{1:t} = \boldsymbol{P}(X_t|e_{1:t})$ be the belief vector that is propagated along the sequence

$$\boldsymbol{f}_{1:t+1} = FORWARD(\boldsymbol{f}_{1:t}, e_{1:t+1})$$

where $\boldsymbol{f}_{1:0} = \boldsymbol{P}(X_o)$

$$\boldsymbol{f}_{1:t+1} = \alpha\, \boldsymbol{O_{t+1}} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

$\boldsymbol{O}_{t+1}$: Diagonal matrix with $P(e_{t+1}|X_{t+1} = i)$ as the i-th diagonal element

$\boldsymbol{T}$: State Transition Matrix

https://github.com/aimacode/aima-python/blob/master/probability.py

Only for your reading

Bayesian Recursive Filters

- Real-world systems may contain large number of discrete states
  - May not be feasible to do real-time exact inference
  - Alternative: Approximate inference
    - Eg: Particle filtering for robot localization

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7826670/

- Continuous States
  - Kalman filtering
    - Assumes Gaussian distribution for state transition model and sensor noise
    - Widely used in object tracking in videos, real-time target tracking in radar, etc.

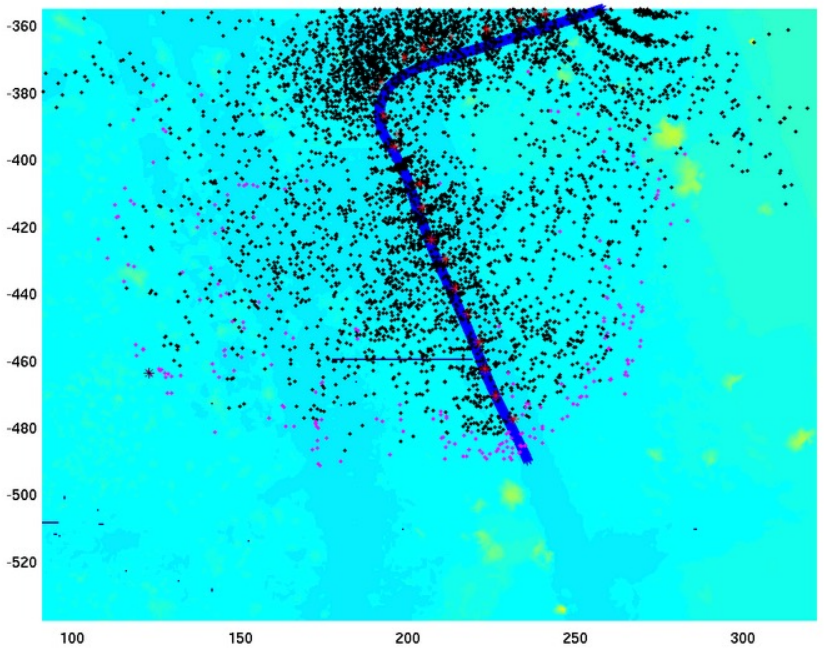


Image Credit: https://www.cs.cmu.edu/~pkv/movies/

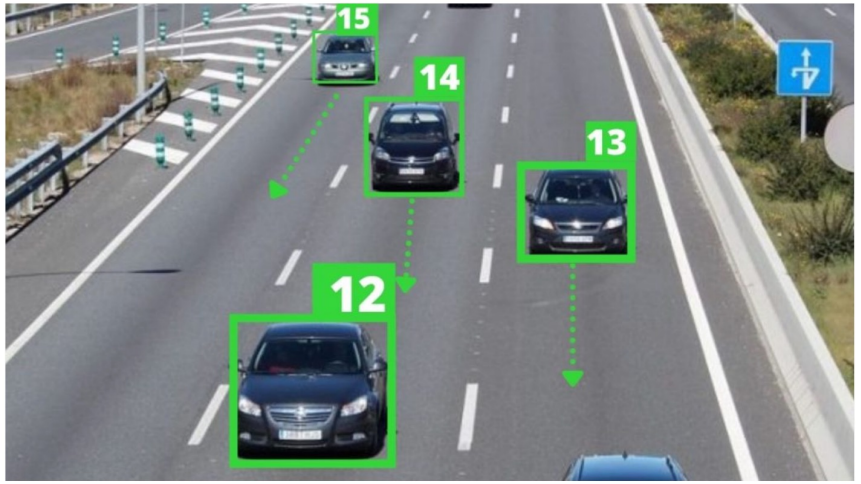


Image Credit: https://pysource.com/2021/11/02/kalman-filter-predict-the-trajectory-of-an-object/