REASONING UNDER UNCERTAINTY

REASONING UNDER UNCERTAINTY

the process of making logical inferences and decisions when information is incomplete, ambiguous, or subject to variability. It is a critical aspect of artificial intelligence (AI), decision-making systems, and knowledge representation, as many real-world scenarios involve uncertainty about facts, outcomes, or relationships.

KEY SOURCES OF UNCERTAINTY

- **Incomplete Information:**
  - Not all relevant information is available.
  - Example: A doctor diagnosing a patient without knowing all symptoms.
- **Ambiguity:**
  - Multiple interpretations of the same information are possible.
  - Example: A word in a sentence having multiple meanings.
- **Stochastic Nature:**
  - The outcome of an event is inherently probabilistic.
  - Example: Rolling a die or predicting weather conditions.
- **Measurement Noise:**
  - Observations or data may be imprecise or noisy.
  - Example: Sensor readings in robotics.

TECHNIQUES FOR REASONING UNDER UNCERTAINTY

Several frameworks and models are used to handle uncertainty, each suited to different types of problems.

1. Probability Theory
2. Fuzzy Logic
3. Belief Networks (Bayesian Networks)
4. Dempster-Shafer Theory
5. Non-Probabilistic Approaches
6. Markov Models and Hidden Markov Models (HMMs)
7. Machine Learning Techniques

TECHNIQUES FOR REASONING UNDER UNCERTAINTY

1. Probability Theory

- **Bayesian Reasoning**:
    - Models uncertainty using probabilities.
    - Uses **Bayes' Theorem** to update beliefs based on evidence:

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)}$$

    - Example: Diagnosing a disease based on symptoms and prior probabilities of diseases.
- Strengths:
    - Quantifies uncertainty explicitly.
    - Supports sequential updates with new evidence.

TECHNIQUES FOR REASONING UNDER UNCERTAINTY

2. Fuzzy Logic

- Models uncertainty using degrees of truth rather than binary true/false logic.
- Example: Classifying temperature as "hot" or "cold" with varying degrees of membership.
- Strengths:
    - Handles vagueness and imprecision well.
    - Useful for control systems (e.g., fuzzy controllers in appliances).

3. Belief Networks (Bayesian Networks)

- A graphical model that represents probabilistic relationships between variables.
- Example: Predicting the likelihood of rain based on cloud cover and atmospheric pressure.
- Strengths:
    - Handles complex dependencies between variables.
    - Efficient for reasoning in large systems.

TECHNIQUES FOR REASONING UNDER UNCERTAINTY

4. Dempster-Shafer Theory

- Generalizes probability theory to model uncertainty and ignorance explicitly.
- Combines evidence from multiple sources to estimate degrees of belief.
- Example: Combining expert opinions in risk assessment.

5. Non-Probabilistic Approaches

- **Possibility Theory**:
  - Focuses on modeling what is possible, rather than probable.
  - Example: Analyzing worst-case scenarios.
- **Rule-Based Systems**:
  - Use heuristics or expert-defined rules for decision-making under uncertainty.
  - Example: Diagnosing faults in machinery using if-then rules.

TECHNIQUES FOR REASONING UNDER UNCERTAINTY

6. Markov Models and Hidden Markov Models (HMMs)

- Useful for reasoning about systems that evolve over time with probabilistic state transitions.
- Example: Speech recognition, where the actual phoneme sequence is uncertain but inferred from audio signals.

7. Machine Learning Techniques

- Probabilistic models (e.g., Naive Bayes, Gaussian Mixture Models) are often employed for reasoning under uncertainty.
- Example: Predicting stock market trends using probabilistic regression models.

APPLICATIONS OF REASONING UNDER UNCERTAINTY

- Healthcare
- Robotics
- Natural Language Processing (NLP)
- Finance
- Environmental Monitoring
- AI Decision-Making

APPLICATIONS OF REASONING UNDER UNCERTAINTY

1. **Healthcare**:
   - Diagnosing diseases based on symptoms and test results.
   - Example: Bayesian inference in medical expert systems.
2. **Robotics**:
   - Localization and navigation in uncertain environments.
   - Example: Markov localization to estimate a robot's position.
3. **Natural Language Processing (NLP)**:
   - Disambiguating meanings of words or phrases in ambiguous sentences.
   - Example: Hidden Markov Models for part-of-speech tagging.

APPLICATIONS OF REASONING UNDER UNCERTAINTY

4. **Finance**:
   - Assessing risks and predicting market trends.
   - Example: Probabilistic models for credit scoring.
5. **Environmental Monitoring**:
   - Predicting weather patterns or pollution levels based on uncertain data.
6. **AI Decision-Making**:
   - Automated reasoning in games, recommendation systems, and self-driving cars.

CHALLENGES

- Computational Complexity
- Data Requirements
- Modeling Uncertainty
- Interpreting Results

CHALLENGES

1. **Computational Complexity**:
    - Many models, like Bayesian networks, become computationally expensive for large systems.
2. **Data Requirements**:
    - Probabilistic models often require large datasets for accurate parameter estimation.
3. **Modeling Uncertainty**:
    - Selecting the right framework (e.g., probability vs. fuzzy logic) for a given application is non-trivial.
4. **Interpreting Results**:
    - Communicating and explaining uncertain outcomes to non-experts can be challenging.

EXAMPLE
REASONING UNDER UNCERTAINTY IN A MEDICAL DIAGNOSIS

A system models the likelihood of diseases D (e.g., flu, cold) based on observed symptoms S (e.g., fever, cough). Using **Bayes' Theorem**:

$$P(\text{flu} \mid \text{fever}) = \frac{P(\text{fever} \mid \text{flu}) \cdot P(\text{flu})}{P(\text{fever})}$$

Here, the system updates the probability of flu as more symptoms are observed, enabling it to refine its diagnosis.

MARKOV MODEL

a mathematical framework used to model systems that transition between a finite set of states based on certain probabilities. These models are widely used in various domains, such as signal processing, natural language processing, biology, and finance, to represent and analyze sequential or time-dependent data.

KEY FEATURES OF A MARKOV MODEL

1. **States**:
   - A system is represented by a finite set of states, $S = \{S_1, S_2, \ldots, S_n\}$.
   - The system transitions between these states over time.
2. **Markov Property**:
   - The probability of transitioning to the next state depends only on the current state and not on the sequence of prior states.

$$P(X_{t+1} = S_j \mid X_t = S_i, X_{t-1}, \ldots) = P(X_{t+1} = S_j \mid X_t = S_i)$$

3. **Transition Probabilities**:
   - Defined by a **transition matrix** P, where P_{ij} is the probability of transitioning from state S_i to state S_j:

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix}$$

where $\sum_{j=1}^{n} P_{ij} = 1$, ensuring probabilities sum to 1 for each state.

KEY FEATURES OF A MARKOV MODEL

4. **Initial State Distribution**:
    - Specifies the probability of the system starting in each state:

$$\pi = [\pi_1, \pi_2, \ldots, \pi_n], \quad \text{where } \sum_{i=1}^{n} \pi_i = 1.$$

5. **Stationary Behavior**:
    - Over time, the system may reach a stationary distribution where the state probabilities stabilize and no longer depend on the initial distribution.

TYPES OF MARKOV MODELS

1. Discrete-Time Markov Chain
2. Continuous-Time Markov Chain (CTMC):
3. Hidden Markov Model (HMM):

TYPES OF MARKOV MODELS

1. **Discrete-Time Markov Chain (DTMC):**
   - Transitions occur in discrete time steps.
   - Commonly used for systems like weather modeling, board games, or stock price movement.
2. **Continuous-Time Markov Chain (CTMC):**
   - Transitions can occur at any time, governed by rates instead of probabilities.
   - Widely used in modeling queues or biological processes.
3. **Hidden Markov Model (HMM):**
   - Extends Markov models by introducing hidden states that are not directly observable. Outputs from the system are observable and provide indirect information about the state.
   - Example: Speech recognition, where the observed sound corresponds to hidden phonemes.

APPLICATIONS OF MARKOV MODELS

1. **Natural Language Processing**:
   - Text prediction or part-of-speech tagging using Markov chains or Hidden Markov Model
2. **Bioinformatics**:
   - Gene sequencing and protein structure prediction.
3. **Finance**:
   - Modeling stock price changes and credit ratings.
4. **Signal Processing**:
   - Modeling noise and communication channels.
5. **Behavior Analysis**:
   - Predicting user behavior or activity in recommendation systems.

EXAMPLE: WEATHER PREDICTION

Imagine a simple Markov model with two states:

1. S_1: Sunny
2. S_2: Rainy

The transition matrix might look like this:

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix}$$

- $P_{11} = 0.8$: If it's sunny today, there's an 80% chance it will be sunny tomorrow.
- $P_{12} = 0.2$: If it's sunny today, there's a 20% chance it will rain tomorrow.

The initial state could be $\pi = [1, 0]$ (e.g., starting with a sunny day).

USE OF MARKOV MODEL IN KNOWLEDGE REPRESENTATION AND REASONING

Markov models are widely used in **knowledge representation and reasoning** to model uncertainty, temporal processes, and sequential dependencies in complex systems. Their ability to capture probabilistic transitions between states makes them a valuable tool for representing and reasoning about systems where events unfold over time or involve uncertainty.

APPLICATIONS OF MARKOV MODELS IN KNOWLEDGE REPRESENTATION AND REASONING

1. Temporal Reasoning

- Markov models provide a way to reason about systems that change over time.
- Example:
  - Representing the progression of a disease through different stages.
  - Modeling user behavior transitions in a system (e.g., from browsing to purchasing).

2. Probabilistic Inference

- Markov models allow reasoning about the likelihood of future states given the current or past states.
- Example:
  - Predicting the next action a user might take in a recommendation system.
  - Inferring missing data in sequential observations.

APPLICATIONS OF MARKOV MODELS IN KNOWLEDGE REPRESENTATION AND REASONING

3. Decision Support Systems

- Markov models are used in decision-making processes where outcomes depend on a sequence of events.
- Example:
  - Healthcare: Deciding on treatment plans by modeling patient states as a Markov process.
  - Maintenance: Predicting when a machine is likely to fail and scheduling preventive actions.

4. Natural Language Understanding

- In natural language processing (NLP), Markov models are used for sequence labeling and prediction.
- Example:
  - Part-of-speech tagging: Identifying grammatical categories of words in sentences.
  - Speech recognition: Converting spoken words into text by modeling phoneme transitions.

APPLICATIONS OF MARKOV MODELS IN KNOWLEDGE REPRESENTATION AND REASONING

5. Reasoning Under Uncertainty

- Markov models handle scenarios where knowledge is incomplete or uncertain.
- Example:
  - Robotics: Estimating the robot's location (Markov localization).
  - Environmental monitoring: Predicting pollutant levels based on past observations.

6. Multi-Agent Systems

- In systems with multiple agents, Markov models can represent interactions and state transitions between agents.
- Example:
  - Modeling the behavior of autonomous vehicles in traffic.
  - Representing strategies in game theory.

SPECIFIC TYPES OF MARKOV MODELS FOR KNOWLEDGE REPRESENTATION

1. Markov Decision Processes (MDPs)
2. Hidden Markov Models (HMMs)
3. Dynamic Bayesian Networks (DBNs)

SPECIFIC TYPES OF MARKOV MODELS FOR KNOWLEDGE REPRESENTATION

1. **Markov Decision Processes (MDPs):**
   - Extend Markov models to include actions and rewards, enabling reasoning about optimal strategies in uncertain environments.
   - Example:
     - Planning under uncertainty in AI, such as navigation or resource allocation.
2. **Hidden Markov Models (HMMs):**
   - Represent systems where the underlying states are not directly observable but can be inferred through observed outputs.
   - Example:
     - Activity recognition: Inferring user activities (e.g., walking, running) from sensor data.
     - Gene prediction in bioinformatics.
3. **Dynamic Bayesian Networks (DBNs):**
   - A generalization of Markov models that allow reasoning with richer dependency structures.
   - Example:
     - Predicting equipment failures based on multiple interconnected sensors.

Advantages of Using Markov Models in Knowledge Representation

1. **Simplicity**:
   - The Markov property simplifies the representation and reasoning process by reducing dependencies to the current state.
2. **Probabilistic Framework**:
   - Provides a natural way to model uncertainty and make predictions.
3. **Scalability**:
   - Markov models can handle large-scale systems efficiently when the state space is well-defined.
4. **Flexibility**:
   - Extendable to include actions, observations, and rewards (e.g., MDPs, HMMs).

Challenges

1. **State Space Explosion**:
   - In complex systems, the number of states can grow exponentially, making computation challenging.
2. **Assumption of Markov Property**:
   - The assumption that the future depends only on the current state may not always hold.
3. **Data Requirements**:
   - Requires sufficient data to estimate transition probabilities accurately.

Example: Use in Healthcare

Consider a system for modeling disease progression in patients. States might include "Healthy," "Mild Condition," "Severe Condition," and "Recovery." A Markov model can:

- Represent the likelihood of transitioning between these states based on treatment.
- Help doctors make decisions about the best course of action by predicting future patient conditions.

By incorporating Markov models into knowledge representation, systems can reason effectively in dynamic, uncertain environments, enabling more intelligent decision-making.

8.2 Definitions

The Markov chain is the process $X_0, X_1, X_2, \ldots$.

Definition: The **state** of a Markov chain at time t is the *value of X_t.*

For example, if $X_t = 6$, we say *the process is in state 6 at time t.*

Definition: The **state space** of a Markov chain, S, is the set of values that each X_t can take. For example, $S = \{1, 2, 3, 4, 5, 6, 7\}$.

Let S have size N (possibly infinite).

Definition: A **trajectory** of a Markov chain is *a particular set of values for* $X_0, X_1, X_2, \ldots$.

For example, if $X_0 = 1$, $X_1 = 5$, and $X_2 = 6$, then the trajectory up to time $t = 2$ is $1, 5, 6$.

More generally, if we refer to the trajectory $s_0, s_1, s_2, s_3, \ldots$, we mean that $X_0 = s_0$, $X_1 = s_1$, $X_2 = s_2$, $X_3 = s_3$, ...

'Trajectory' is just a word meaning *'path'.*

Markov Property

The basic property of a Markov chain is that *only the most recent point in the trajectory affects what happens next.*

This is called the *Markov Property.*
It means that X_{t+1} *depends upon* X_t, *but it does not depend upon* $X_{t-1}, \ldots, X_1, X_0$.

We formulate the Markov Property in mathematical notation as follows:

$$\boxed{\mathbb{P}(X_{t+1} = s \,|\, X_t = s_t, X_{t-1} = s_{t-1}, \ldots, X_0 = s_0) \;=\; \mathbb{P}(X_{t+1} = s \,|\, X_t = s_t),}$$

for all $t = 1, 2, 3, \ldots$ and for all states $s_0, s_1, \ldots, s_t, s$.

Explanation:

$$\mathbb{P}(\underset{\substack{\uparrow \\ \textit{distribution} \\ \textit{of } X_{t+1}}}{X_{t+1} = s} \;\Big|\; \underset{\substack{\uparrow \\ \textit{depends} \\ \textit{on } X_t}}{X_t = s_t}, \; \underbrace{\cancel{X_{t-1} = s_{t-1}, X_{t-2} = s_{t-2}, \ldots, X_1 = s_1, X_0 = s_0}}_{\substack{\uparrow \\ \textit{but whatever happened before time } t \\ \textit{doesn't matter.}}})$$

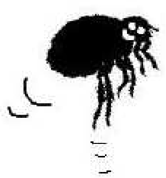
Definition: Let $\{X_0, X_1, X_2, \ldots\}$ be a sequence of discrete random variables. Then $\{X_0, X_1, X_2, \ldots\}$ is a **Markov chain** if *it satisfies the Markov property:*

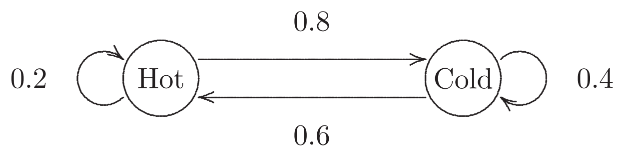
$$\mathbb{P}(X_{t+1} = s \,|\, X_t = s_t, \ldots, X_0 = s_0) \;=\; \mathbb{P}(X_{t+1} = s \,|\, X_t = s_t),$$

for all $t = 1, 2, 3, \ldots$ *and for all states* $s_0, s_1, \ldots, s_t, s$.

8.3 The Transition Matrix

We have seen many examples of **transition diagrams** to describe Markov chains. The transition diagram is so-called because it shows the transitions between different states.





We can also summarize the probabilities in a **matrix:**

$$\begin{array}{cc} & \overbrace{\begin{array}{cc}\text{Hot} & \text{Cold}\end{array}}^{X_{t+1}} \\ X_t \left\{ \begin{array}{c} \text{Hot} \\ \text{Cold} \end{array} \right. & \begin{pmatrix} 0.2 & 0.8 \\ 0.6 & 0.4 \end{pmatrix} \end{array}$$

The matrix describing the Markov chain is called the *transition matrix.* It is the most important tool for analysing Markov chains.

Transition Matrix

$$\begin{array}{cc} & \overbrace{\longleftarrow \text{ list all states } \longrightarrow}^{X_{t+1}} \\ X_t \left\{ \begin{array}{c}\text{list}\\ \text{all}\\ \text{states}\end{array}\right. \updownarrow & \left(\begin{array}{c} \texttt{insert} \\ \texttt{probabilities} \\ p_{ij} \end{array} \right) \begin{array}{l} \longleftarrow \text{rows add to 1} \\ \longleftarrow \text{rows add to 1}\end{array} \end{array}$$

The transition matrix is usually given the symbol $P = (p_{ij})$.

In the transition matrix P:

- *the ROWS represent NOW, or FROM (X_t);*
- *the COLUMNS represent NEXT, or TO (X_{t+1});*
- *entry (i, j) is the CONDITIONAL probability that NEXT $= j$, given that NOW $= i$: the probability of going FROM state i TO state j.*

$$\boxed{p_{ij} = \mathbb{P}(X_{t+1} = j \mid X_t = i).}$$

Notes: 1. The transition matrix P must list *all* possible states in the state space S.

2. P is a *square matrix* $(N \times N)$, because X_{t+1} and X_t both take values in the same state space S (of size N).

3. The **rows** of P should each *sum to 1:*

$$\sum_{j=1}^{N} p_{ij} = \sum_{j=1}^{N} \mathbb{P}(X_{t+1} = j \mid X_t = i) = \sum_{j=1}^{N} \mathbb{P}_{\{X_t = i\}}(X_{t+1} = j) = 1.$$

This simply states that X_{t+1} *must* take one of the listed values.

4. The **columns** of P do **not** in general sum to 1.

Definition: Let $\{X_0, X_1, X_2, \ldots\}$ be a Markov chain with state space S, where S has size N (possibly infinite). The **transition probabilities** of the Markov chain are

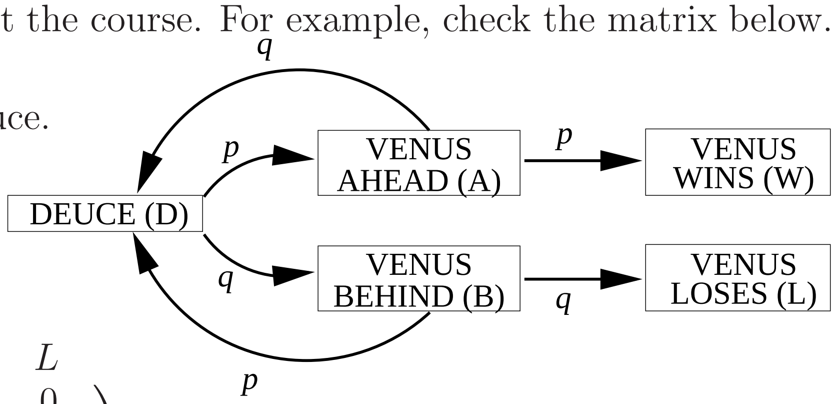
$$p_{ij} = \mathbb{P}(X_{t+1} = j \mid X_t = i) \quad \text{for } i, j \in S, \quad t = 0, 1, 2, \ldots$$

Definition: The **transition matrix** of the Markov chain is $P = (p_{ij})$.

8.4 Example: setting up the transition matrix

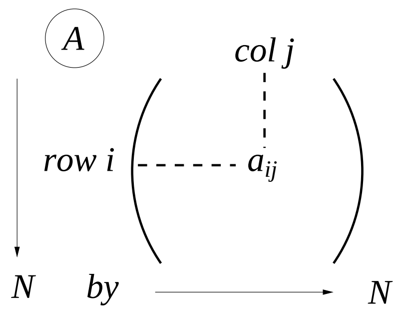
We can create a transition matrix for any of the transition diagrams we have seen in problems throughout the course. For example, check the matrix below.

Example: Tennis game at Deuce.



$$\begin{array}{c} \\ D \\ A \\ B \\ W \\ L \end{array} \begin{array}{c} \begin{array}{ccccc} D & A & B & W & L \end{array} \\ \begin{pmatrix} 0 & p & q & 0 & 0 \\ q & 0 & 0 & p & 0 \\ p & 0 & 0 & 0 & q \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

8.5 Matrix Revision



Notation

Let A be an $N \times N$ matrix.

We write $A = (a_{ij})$,
i.e. A comprises elements a_{ij}.

The (i, j) element of A is written both as a_{ij} and $(A)_{ij}$:
e.g. for matrix A^2 we might write $(A^2)_{ij}$.

Example 4.1 (Forecasting the Weather): Suppose that the chance of rain tomorrow depends on previous weather conditions only through whether or not it is raining today and not on past weather conditions. Suppose also that if it rains today, then it will rain tomorrow with probability α; and if it does not rain today, then it will rain tomorrow with probability β.

If we say that the process is in state 0 when it rains and state 1 when it does not rain, then the above is a two-state Markov chain whose transition probabilities are given by

$$\mathbf{P} = \begin{Vmatrix} \alpha & 1-\alpha \\ \beta & 1-\beta \end{Vmatrix} \quad \blacklozenge$$

Example 4.2 (A Communications System): Consider a communications system which transmits the digits 0 and 1. Each digit transmitted must pass through several stages, at each of which there is a probability p that the digit entered will be unchanged when it leaves. Letting X_n denote the digit entering the nth stage, then $\{X_n, n = 0, 1, \ldots\}$ is a two-state Markov chain having a transition probability matrix

$$\mathbf{P} = \begin{Vmatrix} p & 1-p \\ 1-p & p \end{Vmatrix} \quad \blacklozenge$$

Example 4.3 On any given day Gary is either cheerful (C), so-so (S), or glum (G). If he is cheerful today, then he will be C, S, or G tomorrow with respective probabilities 0.5, 0.4, 0.1. If he is feeling so-so today, then he will be C, S, or G tomorrow with probabilities 0.3, 0.4, 0.3. If he is glum today, then he will be C, S, or G tomorrow with probabilities 0.2, 0.3, 0.5.

Letting X_n denote Gary's mood on the nth day, then $\{X_n, n \geq 0\}$ is a three-state Markov chain (state $0 = C$, state $1 = S$, state $2 = G$) with transition probability matrix

$$\mathbf{P} = \begin{Vmatrix} 0.5 & 0.4 & 0.1 \\ 0.3 & 0.4 & 0.3 \\ 0.2 & 0.3 & 0.5 \end{Vmatrix} \quad \blacklozenge$$

Example 4.4 (Transforming a Process into a Markov Chain): Suppose that whether or not it rains today depends on previous weather conditions through the last two days. Specifically, suppose that if it has rained for the past two days, then it will rain tomorrow with probability 0.7; if it rained today but not yesterday, then it will rain tomorrow with probability 0.5; if it rained yesterday but not today, then it will rain tomorrow with probability 0.4; if it has not rained in the past two days, then it will rain tomorrow with probability 0.2.

If we let the state at time n depend only on whether or not it is raining at time n, then the above model is not a Markov chain (why not?). However, we can transform the above model into a Markov chain by saying that the state at any time is determined by the weather conditions during both that day and the previous day. In other words, we can say that the process is in

state 0 if it rained both today and yesterday,
state 1 if it rained today but not yesterday,
state 2 if it rained yesterday but not today,
state 3 if it did not rain either yesterday or today.

The preceding would then represent a four-state Markov chain having a transition probability matrix

$$\mathbf{P} = \begin{Vmatrix} 0.7 & 0 & 0.3 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0.4 & 0 & 0.6 \\ 0 & 0.2 & 0 & 0.8 \end{Vmatrix}$$

The reader should carefully check the matrix P, and make sure he or she understands how it was obtained. ◆

Example 4.5 (A Random Walk Model): A Markov chain whose state space is given by the integers $i = 0, \pm 1, \pm 2, \ldots$ is said to be a random walk if, for some number $0 < p < 1$,

$$p_{i,i+1} = p = 1 - P_{i,i-1}, \qquad i = 0, \pm 1, \ldots$$

The preceding Markov chain is called a random walk for we may think of it as being a model for an individual walking on a straight line who at each point of time either takes one step to the right with probability p or one step to the left with probability 1 − p. ◆

Example 4.6 (A Gambling Model): Consider a gambler who, at each play of the game, either wins \$1 with probability p or loses \$1 with probability $1 - p$. If we suppose that our gambler quits playing either when he goes broke or he attains a fortune of \$N, then the gambler's fortune is a Markov chain having transition probabilities

$$P_{i,i+1} = p = 1 - P_{i,i-1}, \qquad i = 1, 2, \ldots, N - 1$$

$$P_{00} = P_{NN} = 1$$

States 0 and N are called absorbing states since once entered they are never left. Note that the above is a finite state random walk with absorbing barriers (states 0 and N). ◆

1 Introduction

So far, we have looked at algorithms for reasoning in a static world. However, the real world changes over time. When the world changes, we need to reason about a sequence of events. In this lecture, I will introduce hidden Markov models and describe how we can use hidden Markov models to model a changing world. Hidden Markov models have many real-world applications.

2 A Hidden Markov Model for the Umbrella Story

I will use an Umbrella Story as a running example. Here's the story.

Example: You are a security guard stationed at a secret underground installation. Every day, you want to know whether it's raining or not. Unfortunately, your only access to the outside world is when you see the director brings or does not bring an umbrella each morning.

This story is a bit depressing, but it has some important elements. We are underground, but we want to know whether it's raining or not. The state of the world is whether it's raining or not, and we cannot observe it directly. Instead, we will observe a signal — whether the director comes with an umbrella or not. This signal tells us some information about the state of the world.

Also, the signal is noisy since we are in a world with uncertainty. Even if it's raining, the director might not carry an umbrella because they forget. Or if it's not raining, the director might carry an umbrella because they forgot to check the weather report.

In short, the state is not observable, but we observe a noisy signal, which tells us something about the state.

2.1 Defining Variables

Let's model the umbrella story using a Bayesian network.

We need to reason about events over time. Let's define the time steps. For the umbrella story, a day is a reasonable time step. Every day we observe a new signal, and we may want to update our estimate of the state.

Next, let's define some random variables. We need two types of random variables.

First, we need to model the state — whether it's raining or not. Let's define a binary random variable S to denote the state. S is true when it's raining and false otherwise.

We also need to model our noisy signal or observation. Let's define a binary random variable O to denote the signal. O is true when the director brings an umbrella and false otherwise.

Finally, I will use subscripts to indicate the time step for each variable.

To summarize,

- S_t is true if it rains on day t and false otherwise.
- O_t is true if the director brings an umbrella on day t and false otherwise.

2.2 The Transition Model

Next, let's construct the transition model for the umbrella story.

We need to ask the following question: How does the state change from one day to the next? How does the state today depend on the states in the past?

We're reasoning about events over time. There is one state for every time step. In general, the current state may depend on all the past states. Mathematically, we can express this as a conditional probability distribution $P(S_t|S_0 \wedge \cdots \wedge S_{t-1})$.

Unfortunately, this model has a significant problem. As we advance in time, the size of the conditional distribution increases. If we are modeling an arbitrary time step in the future, the conditional distribution will be un-boundedly large. An unbounded distribution is problematic since we have to store this table somewhere to perform inference.

Let's solve this problem by changing our assumption. Instead of assuming that each state depends on all of the past states, we will assume that each state depends on a fixed number of past states.

2.2.1 K-order Markov chain

Using our new assumption, we can define a K-order Markov chain. Each state depends on the K previous states.

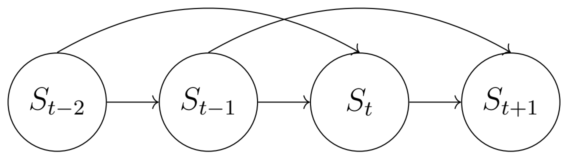
The simplest case is a first-order Markov process. Each state depends on the previous state only. Mathematically, the original transition probability is equal to the conditional probability $P(S_t|S_{t-1})$. Graphically, this model is a single chain.

$$S_{t-2} \rightarrow S_{t-1} \rightarrow S_t \rightarrow S_{t+1}$$

The transition model:

$$P(S_t|S_{t-1} \wedge S_{t-2} \wedge S_{t-3} \wedge \cdots \wedge S_0) = P(S_t|S_{t-1})$$

In some cases, we may not be happy that each state only depends on one previous state. Perhaps, the two previous states both have useful information to determine the current state. We can define a second-order Markov process. Each state depends on the two previous states. For example, S_t depends on S_{t-1} and S_{t-2}.

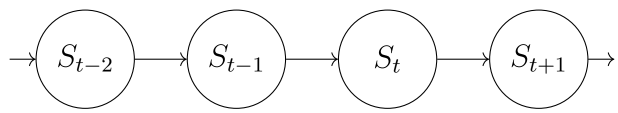


The transition model:

$$P(S_t|S_{t-1} \land S_{t-2} \land S_{t-3} \land \cdots \land S_0) = P(S_t|S_{t-1} \land S_{t-2})$$

We can generalize this to any fixed value of K. In a K-order Markov chain, each state depends on the previous K states.

Let's model our umbrella story as a first-order Markov process. This model makes a key assumption called the Markov assumption.



The transition model:

$$P(S_t|S_{t-1} \land S_{t-2} \land S_{t-3} \land \cdots \land S_0) = P(S_t|S_{t-1})$$

The Markov assumption says that: the current state has sufficient information to determine the next state. We do not have to look at older states in the past. I've always remembered this assumption using this sentence: "the future is independent of the past given the present."

Would you want to live in a world with the Markov assumption? I certainly would. Living in a Markovian world is wonderful because our slate gets wiped clean every day. Every day is a new beginning, and we can start fresh. Forget about the past. We need to seize the moment and do the best we can for today. What happens today determines what will happen tomorrow.

2.2.2 Stationary Process

Given the Markov assumption, how many conditional probability tables do we need to specify the transition model? In general, the transition probabilities at each time step may be different. We potentially need a separate table for each time step.

To simplify our model, we can choose to make it stationary. A stationary process doesn't mean the world does not change over time. The world still changes from one time step to the next. The word "stationary" means that how the world changes remain fixed. In other words, the transition probabilities are the same for every time step.

There are several advantages to using a stationary model. First, it's simple to specify. We can specify one conditional probability table and use it for every time step. In other words, a stationary model allows us to use a finite number of parameters to define an infinite network.

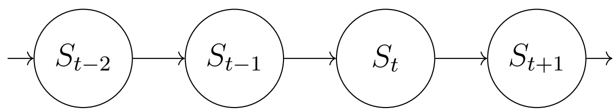
Our umbrella story can run for an unlimited number of time steps, but we can model it using a finite number of probabilities.

Second, a stationary model is a natural choice. When we're modeling real things, the dynamics often do not change. You might be wondering: what if I encounter a situation where the dynamics change? In that case, there is probably another feature causing the dynamics to change. If we model this feature explicitly, the dynamics in the new model become fixed again.

This is a partial model for the umbrella story. The states form a first-order Markov chain. The transition model has a single conditional probability table for every time step. We also have a prior distribution for the state at time 0.

$$\boxed{P(s_0) = 0.5}$$

$$\boxed{\begin{array}{l} P(s_t|s_{t-1}) = 0.7 \\ P(s_t|\neg s_{t-1}) = 0.3 \end{array}}$$



2.3 Sensor Model

By sensors, I mean the noisy signal we observe about the state at each time step. The signal on a day is true if the director carries an umbrella and false otherwise.

The signal at time t corresponds to the evidence variable O_t. The value of O_t could potentially depend on all the states S_0 up to S_t and all the previous signals O_0 up to O_{t-1}. As you can see, we will encounter a similar problem as before: the conditional probability distribution will grow unboundedly as time goes on.

Let's make another Markov assumption for the signals. To distinguish this from the Markov assumption regarding the states, let's call it the sensor Markov assumption. The sensor Markov assumption says that each state has sufficient information to generate its observation. Therefore, O_t needs to condition on S_t only. We can simplify the conditional probability to $P(O_t|S_t)$.

Similar to the transition model, we will assume that the sensor model is stationary. That is, the sensor model for every time step remains the same.

Below is the complete hidden Markov model for the umbrella story. Let me use this example to describe some important components of a hidden Markov model.

(1) The variables: The state at each time step is not observable, but the state generates a noisy signal that is observable.

(2) The state transitions satisfy the Markov assumption. Each state depends on the previous state only. Also, the sensor model satisfies the Markov assumption. Each observation depends on the current state only.

In addition, we simplified our umbrella model by making some further assumptions.

We assumed that the state transitions are stationary. The transition probabilities for each time step are the same.

We also assumed that the sensor model is stationary. The probabilities of generating a signal at each time step are the same.

$$\boxed{P(s_0) = 0.5}$$

$$\boxed{\begin{array}{l} P(s_t|s_{t-1}) = 0.7 \\ P(s_t|\neg s_{t-1}) = 0.3 \end{array}} \quad \boxed{\begin{array}{l} P(o_t|s_t) = 0.9 \\ P(o_t|\neg s_t) = 0.2 \end{array}}$$