WEEK 7 NPTEL: AI

This file, "Lecture 31: Bayesian Network (BN)," discusses Bayesian Networks as a method for representing and reasoning with probabilities, especially in situations with multiple interdependent variables.

Here's a breakdown of the key points:

- **Bayes' Rule:** The lecture begins by explaining Bayes' rule, emphasizing its practical utility in converting causal probabilities (P(effect|cause)) into diagnostic probabilities (P(cause|effect)). An example of medical diagnosis (meningitis and stiff neck) is used to illustrate its application and the importance of prior probabilities.
- **Combining Evidence:** It addresses the challenge of combining multiple pieces of evidence, noting that directly using the full joint distribution or a simple application of Bayes' rule becomes computationally intractable as the number of variables increases.
- **Conditional Independence:** The concept of conditional independence is introduced as a crucial simplification. It explains that two variables (e.g., toothache and a probe "catch") might be dependent, but become independent *given* a common cause (e.g., a cavity). This allows for the decomposition of large probability tables into smaller, more manageable ones.
- **Naive Bayes Model:** The dentistry example leads to the "naive Bayes model," where a single cause directly influences multiple conditionally independent effects. While simplifying assumptions are made, these models often perform surprisingly well in practice.
- **Bayesian Networks (BNs):** BNs are introduced as a data structure to represent these dependencies. They are directed acyclic graphs (DAGs) where nodes are random variables, links show direct influence (causes are parents of effects), and each node has a conditional probability distribution (CPT) given its parents.
- **BN Semantics:** The semantics of BNs are presented in two ways: as a representation of the full joint probability distribution (where the joint probability is a product of individual conditional probabilities from the CPTs) and as an encoding of conditional independence statements.
- **Constructing BNs:** A methodology for constructing BNs is outlined:
    1. **Nodes:** Determine and order variables (causes before effects is preferred for compactness).
    2. **Links:** For each variable, choose a minimal set of parents from preceding variables that directly influence it.
    3. **CPTs:** Define the conditional probability table for each node given its parents.
- **Compactness and Node Ordering:** The lecture highlights that BNs can be significantly more compact than full joint distributions, especially in locally structured systems. The choice of node ordering is crucial; ordering causes before effects generally leads to more compact and intuitive networks with fewer links and easier probability judgments. Poor ordering can result in more complex networks that fail to represent all conditional independence relationships.
- **Example (Burglar Alarm):** A detailed example of a burglar alarm system (with variables for burglary, earthquake, alarm, John calling, and Mary calling) is used throughout to illustrate BN concepts, including network topology, CPTs, and the impact of node ordering on network complexity.

In essence, the file explains how Bayesian Networks provide a powerful and concise way to model probabilistic relationships in complex domains by leveraging conditional independence, making it feasible to handle many variables where full joint distributions would be unmanageable.

Lecture 32

This file, "lecture32.pdf," discusses Dynamic Belief Networks (DBNs) and their application in modeling dynamic and uncertain situations.

Here's a summary of its key points:

- **Motivation for Dynamic Models:** It begins by contrasting static probabilistic reasoning (like car repair diagnosis) with dynamic problems (like treating a diabetic patient or tracking a robot), where states and observations change over time.
- **States and Observations:** The file introduces the concept of viewing the world as a series of "time slices," each with unobservable state variables (X_t) and observable evidence variables (E_t). The "umbrella world" example (raining/not raining vs. umbrella/no umbrella) is used to illustrate this.
- **Markov Assumption:** To handle the unbounded size of past states, the Markov assumption is introduced, stating that the current state depends only on a finite, fixed number of previous states (e.g., first-order Markov process where $P(X_t|X_{0:t-1})=P(X_t|X_{t-1})$). It also defines the sensor Markov assumption: $P(E_t|X_{0:t},E_{0:t-1})=P(E_t|X_t)$.
- **Transition and Sensor Models:** These are crucial for DBNs. The transition model ($P(X_t|X_{t-1})$) describes how the world evolves, and the sensor model ($P(E_t|X_t)$) describes how evidence variables get their values. The file also emphasizes the need for an initial prior distribution $P(X_0)$.
- **Improving Accuracy:** Two methods are discussed for improving the accuracy of Markov process models: increasing the order of the Markov process (e.g., second-order) and increasing the set of state variables (e.g., adding `Season_t`, `Temperature_t`).
- **Dynamic Belief Networks (DBNs):** DBNs are defined as Bayesian networks that represent temporal probability models. They are distinguished from Hidden Markov Models (HMMs) by decomposing complex systems into constituent variables, which allows DBNs to leverage sparseness in the temporal probability model, leading to significantly fewer probabilities and more efficient inference compared to HMMs for large problems.
- **Constructing DBNs:** To construct a DBN, one must specify the prior distribution $P(X_0)$, the transition model $P(X_{t+1}|X_t)$, and the sensor model $P(E_t|X_t)$, along with their topologies. The "robot motion in the X-Y plane" example is used to illustrate a more complex DBN.
- **Sensor Failure Models:** The document delves into the importance of robust sensor models that account for noise and failures. It discusses the "Gaussian error model," "transient failure model" (where sensors occasionally send incorrect values), and "persistent failure model" (where a sensor stays broken once it fails). The "BMBroken" variable is introduced to model the status of a battery meter.
- **Exact Inference in DBNs:** While DBNs are Bayesian networks, naive unrolling for inference can be inefficient, leading to $O(t)$ space and time complexity. A recursive filtering update approach, mimicking variable elimination, can achieve constant time and space per filtering update. However, this "constant" is often exponential in the number of

state variables, making exact inference infeasible for large numbers of variables. This leads to the conclusion that approximate methods are often necessary.

Lecture 33
This file, "Lecture 33: Making decisions," discusses decision theory, focusing on how rational agents make choices under uncertainty.

Here's a summary of the key concepts:

- **Decision Theory and Expected Utility:** The lecture introduces decision theory for non-deterministic, partially observable environments. It defines `RESULT(a)` as a random variable for outcome states and `U(s)` as a utility function expressing the desirability of a state. The core principle is **Maximum Expected Utility (MEU)**, where a rational agent chooses the action that maximizes `EU(a|e) = Σ P(Result(a)=s'|a,e)U(s')`.
- **Constraints on Rational Preferences (Axioms of Utility Theory):** To formalize rational preferences, six constraints (axioms) are introduced:
    - **Orderability:** An agent must prefer one lottery over another or be indifferent.
    - **Transitivity:** If A > B and B > C, then A > C.
    - **Continuity:** If B is between A and C, there's a probability *p* such that the agent is indifferent between B for sure and a lottery yielding A with probability *p* and C with probability (1-*p*).
    - **Substitutability:** If A ~ B, then substituting A for B in a more complex lottery doesn't change preferences.
    - **Monotonicity:** If an agent prefers A to B, they prefer a lottery with a higher probability of A (given the same two outcomes).
    - **Decomposability:** Compound lotteries can be simplified using probability laws.

    These axioms lead to the existence of a utility function where `U(A) > U(B)` if and only if `A > B`, and the utility of a lottery is the sum of the probability of each outcome times its utility.
- **Utility Assessment and Scaling:** The process of determining an agent's utility function is called preference elicitation. Utilities are relative, and a scale can be established by fixing the utilities of a "best possible prize" (`uT`) and a "worst possible catastrophe" (`u⊥`), often normalized to 1 and 0 respectively. The utility of other prizes can then be assessed by finding a probability *p* where the agent is indifferent between the prize and a standard lottery `[p, uT; (1-p), u⊥]`.
- **Utility of Money:** While generally agents prefer more money, the utility of money is not linear. People often exhibit **risk-averse** behavior, preferring a sure thing over a gamble with a higher expected monetary value (e.g., declining a 50/50 chance of $2,500,000 or $0 when offered a sure $1,000,000). The difference between a lottery's Expected Monetary Value (EMV) and its certainty equivalent is called the **insurance premium**, which forms the basis of the insurance industry.
- **Expected Utility and Post-Decision Disappointment (Optimizer's Curse):** When choosing the action with the highest *estimated* expected utility from multiple options, the actual outcome is often worse than estimated, even if the estimates are unbiased. This is because optimistic estimates are favored. This phenomenon is called the **optimizer's curse**.

- **Human Judgment and Irrationality:** The file highlights how humans often deviate from the normative prescriptions of decision theory, leading to "predictably irrational" behavior. Examples include:
    - **Allais paradox:** People prefer a sure gain over a gamble with a higher EMV, but then choose the higher EMV in a different scenario, showing inconsistency. This is partly explained by the **certainty effect**.
    - **Ellsberg paradox:** People show **ambiguity aversion**, preferring known probabilities over unknown ones.
    - **Framing effect:** The wording of a problem (e.g., "90% survival rate" vs. "10% death rate") significantly influences choices.
    - **Anchoring effect:** Relative utility judgments are more comfortable than absolute ones, and an extreme option can skew perceptions of other options.

      Evolutionary psychology suggests that human decision-making mechanisms might not be designed for abstract numerical problems, and presenting problems in "evolutionarily appropriate" forms (e.g., visual scenarios instead of percentages) can lead to more rational behavior.

Lecture 34
This document, "Lecture 34: Decision theory - 1," discusses decision-making in complex situations, particularly those with multiple attributes and uncertainty.

Here's a summary of the key concepts:

- **Multiattribute Utility Function:** Introduces the idea of decision-making problems where outcomes have multiple characteristics (attributes). It assumes higher values of an attribute generally lead to higher utility.
- **Dominance:**
    - **Strict Dominance:** If one option is inferior to another on all attributes, it can be discarded. This is useful for narrowing down choices, especially in deterministic cases.
    - **Stochastic Dominance:** A more common generalization for uncertain outcomes. If the cumulative distribution of an attribute for one action is always "to the right" of another (meaning it's consistently better), then the former stochastically dominates the latter. This allows for discarding options without needing exact utility functions, as long as the utility is monotonically nondecreasing with the attribute.
- **Preference Structure and Multiattribute Utility:** Explores how to simplify the specification of utility functions when preferences exhibit regularity.
- **Preferences Without Uncertainty:**
    - **Preference Independence:** Two attributes are independent of a third if the preference between outcomes for the first two doesn't depend on the value of the third.
    - **Mutual Preferential Independence (MPI):** When each attribute in a set is preferentially independent of the others. MPI implies an **additive value function**, simplifying the assessment of preferences.
- **Preferences With Uncertainty:**

    - **Utility Independence:** Extends preference independence to cover lotteries. A set of attributes is utility independent of another set if preferences between lotteries on the first set are independent of the values of the second set.
    - **Mutual Utility Independence (MUI):** When each subset of attributes is utility-independent of the remaining attributes. MUI implies a **multiplicative utility function**, which can still be simpler to assess than a fully general utility function.
- **Decision Networks:**
    - Combines Bayesian networks with additional node types for actions and utilities to represent decision problems.
    - **Chance Nodes (ovals):** Represent random variables (e.g., construction cost, air traffic).
    - **Decision Nodes (rectangles):** Represent points where the decision-maker chooses an action (e.g., Airport Site).
    - **Utility Nodes (diamonds):** Represent the agent's utility function, with parents being variables that directly affect utility.
    - A simplified form omits outcome state chance nodes, with the utility node representing expected utility associated with each action (action-utility function).
- **Evaluating Decision Networks:** The process involves setting evidence variables, iterating through possible decision node values, calculating posterior probabilities for utility parent nodes, and then calculating the resulting utility to find the action with the highest utility.
- **The Value of Information (VPI):**
    - Focuses on deciding what information to acquire before making a "real" action.
    - VPI is derived from the potential for new information to change the agent's optimal action and the significance of that change.
    - The formula for VPI involves averaging the expected utility of the new best action over all possible values of the information, then subtracting the expected utility of the current best action.
    - Information has value if it's likely to cause a change in plan, and if that new plan is significantly better.

Lecture 35
This file, "Lecture 35: Decision theory - 2," covers various aspects of decision theory, particularly in the context of artificial intelligence.

Here's a summary of its key points:

- **Properties of the Value of Information:**
    - The expected value of information (VPI) is always non-negative; information cannot be deleterious, as one can always choose to ignore it.
    - VPI depends on the current state of information and is generally not additive, but it is order-independent.
- **Information Gathering Agent:**
    - A sensible agent should gather information intelligently by considering relevance, cost, and utility.

- A "myopic" information-gathering agent selects observations with the highest utility gain per unit cost, acting as if only a single evidence variable will be acquired. While effective in some cases (e.g., outperforming expert physicians in diagnostic tests), it can be shortsighted.
- Better approaches involve constructing conditional plans and considering the human respondent's perspective when asking questions.

- **Making Complex Decisions (Sequential Decision Problems):**
  - The lecture shifts from one-shot decisions to sequential decision problems in stochastic environments, where utility depends on a sequence of decisions.
  - It introduces **Markov Decision Processes (MDPs)**, which consist of states, actions, a Markovian transition model (probabilities of moving between states), and a reward function.
  - A **policy** (π) is a solution that specifies what action the agent should take in any given state. An **optimal policy** (π^*) yields the highest expected utility over possible environment histories.
- **Utilities Over Time:**
  - Discusses **finite vs. infinite horizons** for decision-making. Finite horizons lead to nonstationary optimal policies (actions change over time), while infinite horizons (without a fixed deadline) lead to stationary optimal policies (actions depend only on the current state). The lecture primarily focuses on infinite horizons.
  - Explores two coherent ways to assign utilities to state sequences under stationarity: **additive rewards** and **discounted rewards**.
  - **Discounted rewards** (using a discount factor γ between 0 and 1) are generally preferred as they keep the utility of infinite sequences finite and better model animal and human preferences over time. Additive rewards are a special case where $\gamma=1$.
  - Challenges with infinite, undiscounted rewards (potentially infinite utilities) are addressed by discounted rewards, proper policies (guaranteed to reach a terminal state), or average rewards.
- **Optimal Policies and Utility States:**
  - The optimal policy for discounted utilities with infinite horizons is independent of the starting state.
  - The **utility of a state** $U(s)$ is the expected sum of discounted rewards if the agent executes an optimal policy starting from s.
  - The optimal policy $\pi^*(s)$ is chosen by maximizing the expected utility of the subsequent state.
- **Value Iteration:**
  - An algorithm for calculating an optimal policy by iteratively calculating the utility of each state until equilibrium.
  - It's based on the **Bellman equation**, which defines the utility of a state as its immediate reward plus the expected discounted utility of the next state, assuming an optimal action choice.
  - Value iteration is an iterative approach to solve the system of non-linear Bellman equations. The algorithm (VALUE-ITERATION) is shown in Figure 4.
- **Policy Iteration:**
  - An alternative algorithm that alternates between two steps:
    1. **Policy evaluation:** Calculating the utility of each state for a given policy.

2. **Policy improvement:** Calculating a new policy using a one-step look-ahead based on the evaluated utilities.

- The algorithm (POLICY-ITERATION) is shown in Figure 7.
- Policy evaluation involves solving a system of linear equations, which can be done efficiently for small state spaces. For large spaces, approximate methods (modified policy iteration) can be used.
- **Asynchronous policy iteration** allows updating subsets of states, leading to potentially more efficient heuristic algorithms.