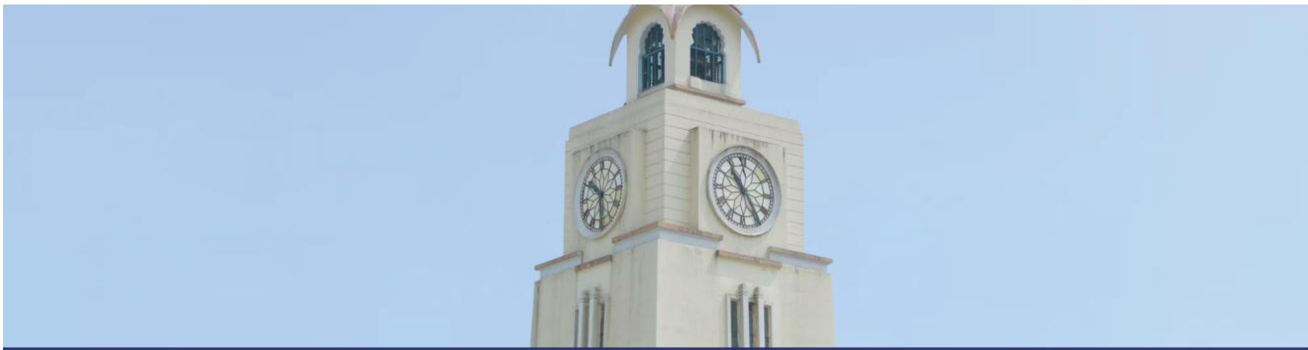


BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED
LEARNING PROGRAMMES

Artificial and Computational Intelligence

PPT Booklet

Artificial and Computational Intelligence

AIMLCZG557

Contributors & Designers of document content : Cluster Course Faculty Team

M1 : Introduction

BITS Pilani
Pilani Campus

Presented by
Faculty Name
BITS Email ID

---

innovate achieve lead

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

Disclaimer

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

---

innovate achieve lead

Agenda

- Course Administration
- Getting Started (with some definitions)
- Course Overview with example

---

innovate achieve lead

Course Administration

About the course

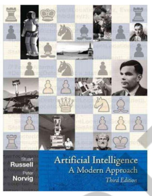
- Focus on
  - principles of artificial intelligence
  - concepts, algorithms involved in building rational agents
  - topics covered like
    - (informed and uninformed) search & applications
    - (logical & probabilistic) knowledge representation
    - (logical & probabilistic) Reasoning & applications
  - topics not-covered like
    - Formal introduction to machine learning algorithms, neural networks etc., are covered as a ML course is running in parallel, Deep neural networks, which are part of AI as well.
    - Hardware aspects of the Design

Course Outline

- **Pedagogy**
  - Weekly online live sessions
  - Webinars on lab implementation
  - Assignment:
    - 1 Quiz-5%,
    - 2 Assignments- 25%
- **Lab Modules**
  - Supported by 6 lab capsules for practical implementation and better understanding of the concepts learned in the live lecture sessions.

About the course

Text Book



Exercises : In Python & its libraries

Evaluation : 25% Assignment + 5% Quiz + 30% Mid Semester + 40% End Semester

Artificial Intelligence

- Term coined by, *John McCarthy* (1955) & *Dartmouth Summer Research Project on Artificial Intelligence* (1956)

On September 2, 1955, the project was formally proposed by McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon. The proposal is credited with introducing the term 'artificial intelligence'.

The Proposal states[7]

> We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

Artificial Intelligence

- Term coined by, *John McCarthy* (1955) & *Dartmouth Summer Research Project on Artificial Intelligence* (1956)

On September 2, 1955, the project was formally proposed by McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon. The proposal is credited with introducing the term 'artificial intelligence'.

The Proposal states[7]

> We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

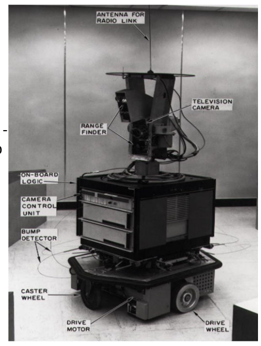
https://en.wikipedia.org/wiki/Dartmouth_workshop [01 June, 2019]

Larger Intent, Dream, Overconfidence ...

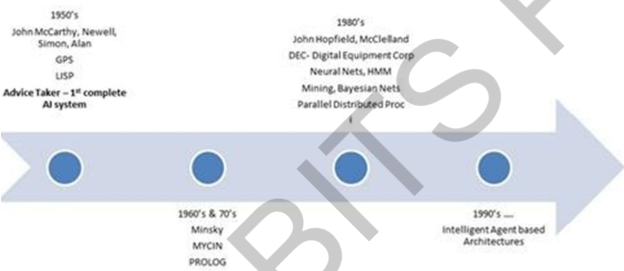
Some Early successes of Dartmouth

Many key projects were initiated after Dartmouth summer project.

Shakey robot - First mobile robot to perceive environment & reason about surroundings, actions - Introduced A* algorithm to find paths - Hough Transform for image analysis - Used Lisp for programming - visibility graph used for finding shortest paths in the presence of obstacles...



A brief history of AI



How is AI unique or in other words different from Applied Math?

Some Early successes of Dartmouth

DENDRAL -
Attempted to encode the domain expertise in molecular biology as an expert system
Led to the creation of expert systems for various other domain, including medical.

A milestone worship in the history of AI !!!

Perspectives of AI

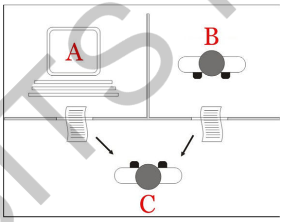
Definitions

	Thought / Reasoning	Acting
Human Performance	THINKING HUMANLY "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning, … " (Bellman, 1978)	ACTING HUMANLY "The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990)
Rational Performance	THINKING RATIONALLY "The study of computations that make it possible to perceive, reason, and act" (Winston, 1992)	ACTING RATIONALLY "Computational intelligence is the study of the design of intelligent agents" (Poole et al., 1998)

Acting Humanly

Turing Test Approach

- *Turing Test* & *Total Turing test* (operational test to determine an entity is intelligent / not) [50's]
- Skills necessary to pass these tests
  - NLP, Knowledge Representation, Automated Reasoning, ML + Computer Vision & Robotics(for total turing test)



Pictorial Representation of Turing Test from https://en.wikipedia.org/wiki/Turing_test

Acting Humanly

Turing Test Approach

Some Definitions of AI:

"The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990)

"The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)

Thinking Humanly

Cognitive Modelling Approach

- How do we capture human thinking to implement?
  - Introspection
  - Psychological Experiments
  - Brain Imaging
- System : *"General Problem Solver"* (*Newell and Simon, 1961*)
  - Designed to work as a universal problem solver
  - Problems represented by horn clauses
  - First AI Machine which has KB + Inference separation
  - Authors focus on this is on comparing the trace of its reasoning steps to traces of human subjects solving the same problems
- Growth of Cognitive science and AI supports each other

Passing the Turing Test

Transcript of a chat

EUGINE - a thirteen-year-old Ukrainian boy, chats

JUDGE: Hello.
EUGENE: Hello, I'm really glad to have the chance to chat with you! My guinea pig Bill sends his regards too!
JUDGE: Is Bill a male or a female?
EUGENE: Ask Bill personally, please.
JUDGE: Well I'd rather talk to you. What is your name?
EUGENE: Call me Eugene. I am glad to talk to you!
JUDGE: My name is Jane and I am female. How about you? What's your gender?
EUGENE: I'm a male. A "guy" I'd say.
JUDGE: Pleased to meet you Eugene. What's the weather like where you are?
EUGENE: Let's get on with our conversation!
JUDGE: Don't you like talking about the weather?
EUGENE: All these talks about weather is a waste of time.
JUDGE: What would you like to discuss?
EUGENE: I don't know. Better tell me more about yourself!

Passing the Turing Test

- 2014 - Royal Society (London) - Sixteenth Anniversary of Alan Turing -
- Chabot - Eugene Goostman - Pretended to be a thirteen-year-old Ukrainian boy
  - Passed the turing test for the first time
  - 10/30 Judges believed the response is from human
- *Turing predicted in 50 years time, computers can be programmed to play imitation game in which an average interrogator fails to identify the machine 70% time in a 5 mins questioning*

Thinking Humanly

Cognitive Modelling Approach

Some Definitions of AI:

"The exciting new effort to make computers think . . . machines with minds, in the full and literal sense." (Haugeland, 1985)

"[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . ." (Bellman, 1978)

Thinking Rationally

"Laws of Thought" Approach

- Invention of Formal Logic, Greek Philosopher **Aristotle**, Third century BC.
- Introduced syllogisms, providing argument structures

In all boring classes, students sleep
It is a boring class
Students sleep in this class [Are you ?]

- Field of Logics gave rise to codifying rational thinking
  - When elements are '*things*', we reason about things

Hurdles to the idea : (1) Not everything can be logically coded (2) no provably correct action at a moment (3) Exhaustive computational resources

Acting Rationally

The Rational Agent Approach

- An agent is an entity that perceives and acts

  This course is about designing rational agents
- Abstractly, an agent is a function from percept histories to actions:

  [f: P* → A]
- For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance
- Computational limitations make perfect rationality unachievable
- Design best program for given machine resources

Acting Rationally

The Rational Agent Approach

- Rational behaviour: doing the *right thing*
- The *right thing:* that which is expected to maximize goal achievement, given the available information
- Rational behaviour is not just about correct inference / thinking, skills needed to pass turing test etc.

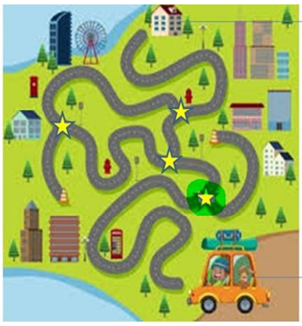
  (adv) : More General - Correct inference is just a thing

  (adv) : More amenable for scientific developments, as the rational behaviour is better defined than human thinking and behaviour

Definitions

Thinking Humanly	**Thinking Rationally**
"The exciting new effort to make computers think . . . *machines with minds*, in the full and literal sense." (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . ." (Bellman, 1978)	"The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly	**Acting Rationally**
"The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	"Computational Intelligence is the study of the design of intelligent agents." (Poole *et al.*, 1998) "AI . . . is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

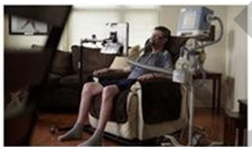
Traveller's Problem



Destination - Fixed **Goal**

Source

AI in HealthCare

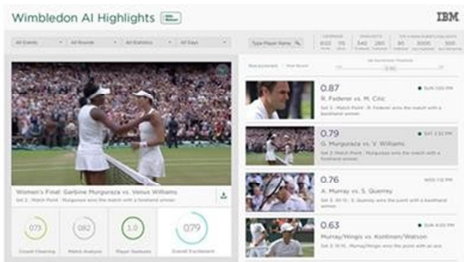


Lyrebird's Project Re-Voice

AI in Culinary Field

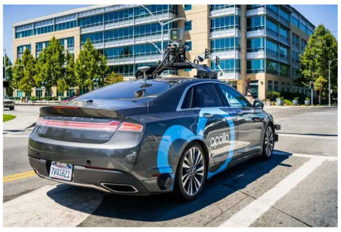
Spyce

AI in NLS
IBM Watson



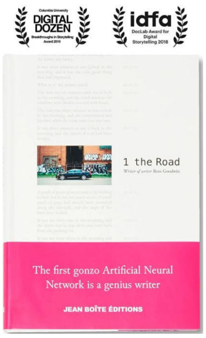
Computer Vision
NLP
ML
Speech Recognitior
Automation

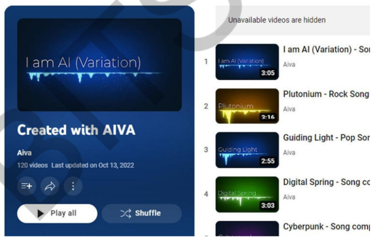
AI in Transportation



AI in HCI
Google Map Navigation Assistant

AI in Literacy & Music





Application Domain

(Additional Notes added from the textbook for self read)

Areas Contributing to AI

Philosophy
Mathematics
Economics
Neuroscience
Psychology
Computer Engineering
Control theory, Cybernetics
Linguistics

- Can formal rules be used to draw valid conclusions?
- How does the mind arise from a physical brain?
- Where does knowledge come from?
- How does knowledge lead to action?

Areas Contributing to AI

Philosophy
Mathematics
Economics
Neuroscience
Psychology
Computer Engineering
Control theory, Cybernetics
Linguistics

Aristotle (384–322 B . C .) : first to formulate precise set of laws to govern rational part of brain

Ramon Lull (d. 1315) : useful reasoning could actually be carried out by a mechanical artifact

Hobbes (1588–1679) : "we add and subtract in our silent thoughts."

Leibniz (1646–1716) : Built a mechanical device intended to carry out operations on concepts rather than numbers

Areas Contributing to AI

Philosophy
Mathematics
Economics
Neuroscience
Psychology
Computer Engineering
Control theory, Cybernetics
Linguistics

Some *'isms* on the working of minds :

Rationalism - Correct Reasonings (Aristotle, Descartes …)

Dualism - A part of the human mind (or soul or spirit) that is outside of nature

Materialism - Alternative to dualism - holds that the brain's operation according to the laws of physics constitutes the mind

Areas Contributing to AI

Philosophy
Mathematics
Economics
Neuroscience
Psychology
Computer Engineering
Control theory, Cybernetics
Linguistics

Obtaining Knowledge

David Hume's (1711–1776) : First principles of induction

Logical positivism- Rudolf Carnap : Every knowledge obtained has a logical connection

Carnap (1905–1997) : A book "*The Logical Structure of the World*" (1928) defined an explicit computational procedure for extracting knowledge from elementary experiences

Areas Contributing to AI

- **Philosophy**
- Mathematics
- Economics
- Neuroscience
- Psychology
- Computer Engineering
- Control theory, Cybernetics
- Linguistics

Connection between knowledge and action:

Aristotle - (in *De Motu Animalium*) that actions are justified by a logical connection between goals and knowledge of the action's outcome

```
I need covering;
a cloak is a covering.
I need a cloak.
What I need, I have to make;
I need a cloak.
I have to make a cloak.
And the conclusion, "I have to make a cloak" is an action
```


Areas Contributing to AI

- Philosophy
- **Mathematics**
- Economics
- Neuroscience
- Psychology
- Computer Engineering
- Control theory, Cybernetics
- Linguistics

- What are the formal rules to draw valid conclusions?
- What can be computed?
- How do we reason with uncertain information?

Areas Contributing to AI

- Philosophy
- **Mathematics**
- Economics
- Neuroscience
- Psychology
- Computer Engineering
- Control theory, Cybernetics
- Linguistics

- What are the formal rules to draw valid conclusions?

George Boole (1815–1864) : Propositional Logic

Gottlob Frege (1848–1925): First order logic

Areas Contributing to AI

- Philosophy
- **Mathematics**
- Economics
- Neuroscience
- Psychology
- Computer Engineering
- Control theory, Cybernetics
- Linguistics

- What can be computed?

Kurt Gödel (1906–1978) : In any formal theory as strong as Peano arithmetic #(the elementary theory of natural numbers), there are true statements that are undecidable in the sense that they have no proof within the theory

Computability, tractability, NP-completeness

Probability theory & inference mechanisms

- https://en.wikipedia.org/wiki/Peano_axioms

Areas Contributing to AI

- Philosophy
- Mathematics
- **Economics**
- Neuroscience
- Psychology
- Computer Engineering
- Control theory, Cybernetics
- Linguistics

- How should we make decisions so as to maximize payoff?

Utility / preferred outcomes

Decision theory -Probability & utility theory

Game theory

- How to make decisions when payoffs are not immediate?
  - MDP

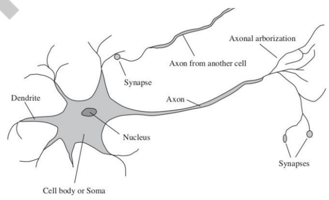
Areas Contributing to AI

- Philosophy
- Mathematics
- Economics
- **Neuroscience**
- Psychology
- Computer Engineering
- Control theory, Cybernetics
- Linguistics

How do brains process information?

- Study of the nervous system / brain
- How does brain enables thoughts - Mystery Still

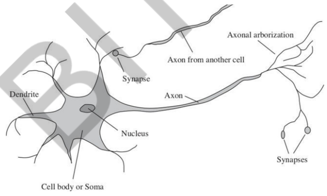
Aristotle , "*Of all the animals, man has the largest brain in proportion to his size*"



Areas Contributing to AI

- Philosophy
- Mathematics
- Economics
- **Neuroscience**
- Psychology
- Computer Engineering
- Control theory, Cybernetics
- Linguistics

	Supercomputer	Personal Computer	Human Brain
Computational units	10^4 CPUs, 10^{12} transistors	4 CPUs, 10^9 transistors	10^{11} neurons
Storage units	10^{14} bits RAM	10^{11} bits RAM	10^{11} neurons
	10^{15} bits disk	10^{13} bits disk	10^{14} synapses
Cycle time	10^{-9} sec	10^{-9} sec	10^{-3} sec
Operations/sec	10^{15}	10^{10}	10^{17}
Memory updates/sec	10^{14}	10^{10}	10^{14}



Areas Contributing to AI

- Philosophy
- Mathematics
- Economics
- Neuroscience
- **Psychology**
- Computer Engineering
- Control theory, Cybernetics
- Linguistics

How do humans and animals think and act?

- *Cognitive Psychology* - Brain as an information-processing device
- Two months after the dartmouth workshop, a workshop in MIT gave birth to *Cognitive Science*
  - George Miller, Noam Chomsky, Allen Newell and Herbert Simon - roles of computer models to address the psychology of memory, language, and logical thinking, issues..

"*a cognitive theory should be like a computer program*" (Anderson, 1980);

Areas Contributing to AI

- Philosophy
- Mathematics
- Economics
- Neuroscience
- Psychology
- **Computer Engineering**
- Control theory, Cybernetics
- Linguistics

Computers & Programming Languages

Areas Contributing to AI

- Philosophy
- Mathematics
- Economics
- Neuroscience
- Psychology
- Computer Engineering
- **Control theory, Cybernetics**
- Linguistics

Control theory

- Deals with the behaviour of dynamic systems
  - behaviour must ensure the error between the current state and goal state is minimized
- Cybernetics - Book by Wiener
  - (Norbert Wiener, 1948) : Scientific study of control and communication in the animal and the machine
- Ashby's Design for a Brain (1948, 1952):
  - Intelligence could be created by the use of homeostatic devices containing appropriate feedback loops to achieve stable adaptive behavior
  - Led to the idea of *design of systems that maximize an objective function over time*

Areas Contributing to AI

- Philosophy
- Mathematics
- Economics
- Neuroscience
- Psychology
- Computer Engineering
- Control theory, Cybernetics
- **Linguistics**

How does language relate to thought?

Verbal Behavior (1957, B. F. Skinner) :

- Behaviorist approach to language learning
- Reviewed by Noam Chomsky
  - criticised lack of notion of creativity in language

Syntactic Structures (1957, Noam Chomsky)

- Computational linguistics / natural language processing as a part of AI
  - Understanding a language is realized as more complex than ever
  - Context, subject matter knowledge complicated it further
  - Representing language consumed volume of work done in NLP, in early times

Course Outline

- **In this course, you will learn :**
  - a solid foundation for designing intelligent agents
  - to represent and use the knowledge learnt for inferencing
  - to model agents operating in uncertain environments
  - optimization models of computation and processing in real world application
- **Modules :**
  - Problem Solving Agent using Search
  - Game Playing
  - Probabilistic Representation and Reasoning
  - Reasoning over time

Required Reading: AIMA - Chapter # 1
AIMA is the first prescribed text book

Thank You for your active participation

Note : Some of the slides are adopted from AIMA TB materials

Next Class Plan

- Agent Design
- Environment
- Agent Architecture
- Problem Solving Agent Formulation

Artificial and Computational Intelligence

AIMLCZG557

Contributors & Designers of document content : Cluster Course Faculty Team

M1 : Introduction &

M2 : : Problem Solving Agent using Search

BITS Pilani

Pilani Campus

Presented by
Faculty Name
BITS Email ID

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

Disclaimer

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

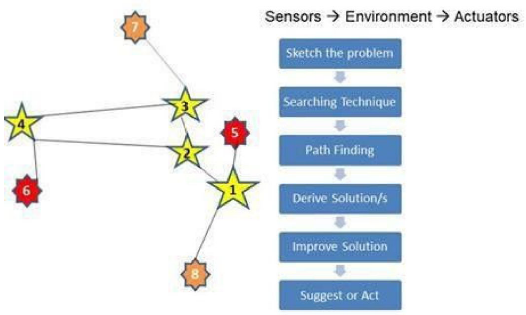
- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

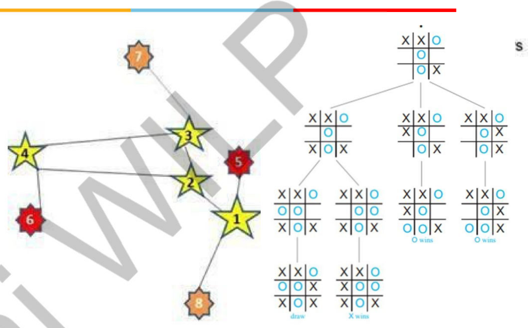
M1	Introduction to AI
M2	Problem Solving Agent using Search
M3	Game Playing
M4	Knowledge Representation using Logics
M5	Probabilistic Representation and Reasoning
M6	Reasoning over time
M7	Ethics in AI

Traveller's Problem

Traveller's Problem



Traveller's Problem



Rational Agents

Rational Agent

Design Principles & Techniques

	Thought / Reasoning	Acting
Human Performance	THINKING HUMANLY "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning, ... " (Bellman, 1978)	ACTING HUMANLY "The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990)
Rational Performance	THINKING RATIONALLY "The study of computations that make it possible to perceive, reason, and act" (Winston, 1992)	ACTING RATIONALLY "Computational intelligence is the study of the design of intelligent agents" (Poole et al., 1998)

Acting Rationally

The Rational Agent Approach

- An agent is an entity that perceives and acts

This course is about designing rational agents

- Abstractly, an agent is a function from percept histories to actions: [f: P* → A]
- For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance
- Computational limitations make perfect rationality unachievable
- Design best program for given machine resources

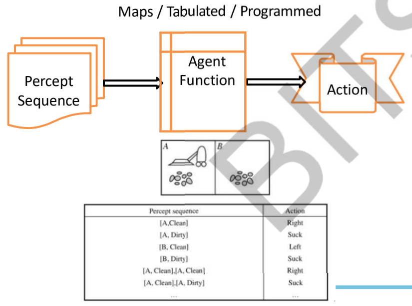
Properties of Rational Agent

- Omniscience : Expected Vs Actual Performance
- Learning Capability : Apriori Knowledge
- Autonomous in decision making: An agent is autonomous if its behaviour is determined by its own experience (with ability to learn and adapt)

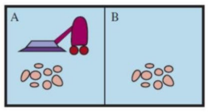
Intelligent Agent

Rational Agent is one that acts to achieve the best outcome or the best expected outcome even under uncertainty

Maps / Tabulated / Programmed



Intelligent Agent



- Percepts: location and contents, e.g., [A , Dirty]
- Actions: *Left*, *Right*, *Suck*, *NoOp*

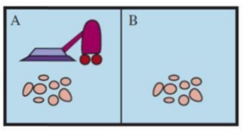
Performance measure: An objective criterion for success of an agent's behaviour

E.g., performance measure of a vacuum-cleaner agent
» amount of dirt cleaned up
» amount of time taken
» amount of electricity consumed
» amount of noise generated, etc.

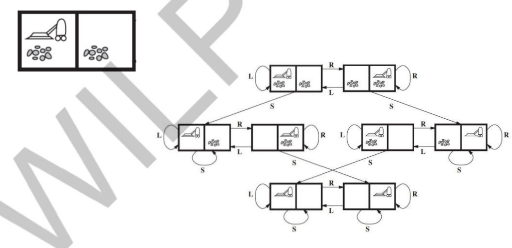
PEAS Design

Intelligent Agent

Percept sequence	Action
[*A*, *Clean*]	*Right*
[*A*, *Dirty*]	*Suck*
[*B*, *Clean*]	*Left*
[*B*, *Dirty*]	*Suck*
[*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮



Vacuum World Problem

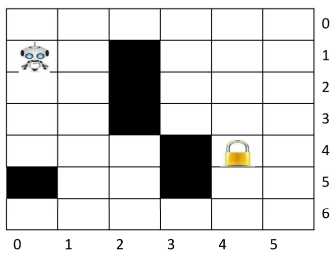


PEAS Environment

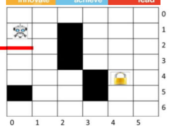
Design on what an application wants the agent to do in the environment

Agent	Performance	Environment	Sensors	Actuators
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Keyboard entry of symptoms, findings, patient's answers	Display of questions, tests, diagnosis, treatments, referrals
Satellite Image analysis system	Correct image categorization	Downlink from orbiting satellite	Color pixel analysis	Display of scene categorization
Interactive English tutor	Student's score on test	Set of students, testing agency	Keyboard entry	Display of exercises, suggestions, corrections

Path finding Robot - Lab Example



PEAS Environment



Agent
Performance
Environment
Sensors

Dimensions of Task Environment

Sensor Based:
- Observability : Full Vs Partial

Action Based:
- Dependency : Episodic Vs Sequential

State Based:
- No.ofState : Discrete Vs Continuous

Agent Based:
> Cardinality : Single Vs MultiAgent

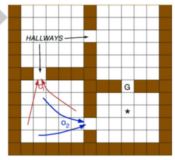
Action & State Based:
- State Determinism : Deterministic Vs Stochastic | Strategic
- Change in Time : Static Vs Dynamic

Task Environment

A rational agent is built to solve a specific task. Each such task would then have a different environment which we refer to as Task Environment

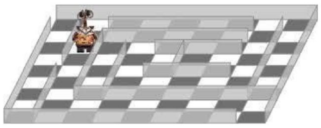
Based on the applicability of each technique for agent implementation its task environment design is determined by multiple dimension

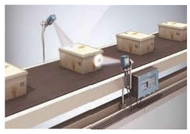
Sensor Based:
- Observability : Full Vs Partial



Task Environment

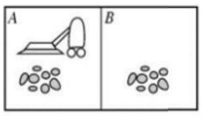
Action Based:
- Dependency : Episodic Vs Sequential

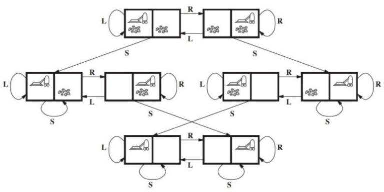




Task Environment

State Based:
- No.of State : **Discrete** Vs Continuous





Task Environment

State Based:
- No.of State : Discrete Vs Continuous

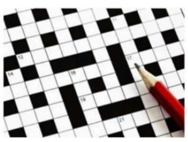
VS.

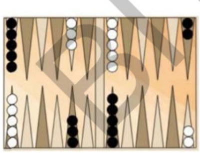
Task Environment

Action & State Based:

- State Determinism : Deterministic Vs Stochastic | Strategic

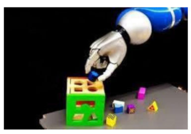
(If the environment is deterministic except for the actions of other agents, then the environment is strategic)





Task Environment

Agent Based:
> Cardinality : Single Vs MultiAgent

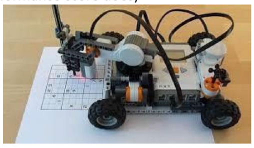


Task Environment

Action & State Based:

➢Change in Time : Static Vs Dynamic

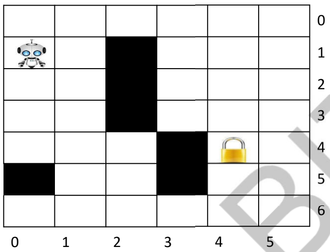
➢(The environment is semi dynamic if the environment itself does not change with the passage of time but the agent's performance score does)



Task Environment

Task Environment	Fully vs Partially Observable	Single vs Multi-Agent	Deterministic vs Stochastic	Episodic vs Sequential	Static vs Dynamic	Discrete vs Continuous
Medical diagnosis system	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Satellite Image Analysis System	Fully	Single	Deterministic	Episodic	Static	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Path finding Robot - Lab Example



Agent
Observability
No.of.Agents
No.of.States
Determinism
Dynamicity
Output Dependency

Learning Objective Achieved

At the end of this class , students Should be able to:

1. Identify the requirement for AI solutions for given problem
2. Understand the significance of State based representations
3. Design the PEAS (Performance, Environment, Actuators, Sensors) for given problem
4. Identify dimensions of TASK environment

Next Class Plan

Structure of Agents-Architectures
Problem Solving Agents
Problem Formulation
Uninformed Search Algorithms

Required Reading: AIMA - Chapter #2

Note : Some of the slides are adopted from AIMA TB materials

Thank You for all your Attention

Artificial and Computational Intelligence

AIMLCZG557

Contributors & Designers of document content : Cluster Course Faculty Team

M2 : : Problem Solving Agent using Search

BITS Pilani
Pilani Campus

Presented by
Faculty Name
BITS Email ID

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

Disclaimer

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

M1	Introduction to AI
M2	Problem Solving Agent using Search
M3	Game Playing
M4	Knowledge Representation using Logics
M5	Probabilistic Representation and Reasoning
M6	Reasoning over time
M7	Ethics in AI

Learning Objective

At the end of this class , students Should be able to:

1. Design problem solving agents
2. Create search tree for given problem
3. Apply uninformed search algorithms to the given problem
4. Compare performance of given algorithms in terms of completeness, optimality, time and space complexity
5. Differentiate for which scenario appropriate uninformed search technique is suitable and justify

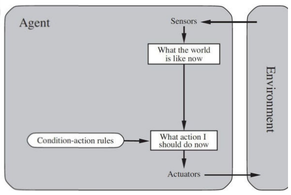
Agents Architectures

Agent Architectures

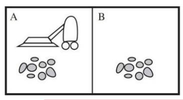
Simple Reflex Agent

function SIMPLE-REFLEX-AGENT(*percept*) returns an action
persistent: *rules*, a set of condition–action rules
state←INTERPRET-INPUT(*percept*)
rule←RULE-MATCH(*state, rules*)
action ←*rule*.ACTION
return *action*

function REFLEX-VACUUM-AGENT([*location,status*]) **returns** an action
if *status* = *Dirty* **then return** Suck
else if *location* = *A* **then return** Right
else if *location* = *B* **then return** Left



Simple Reflex Agents

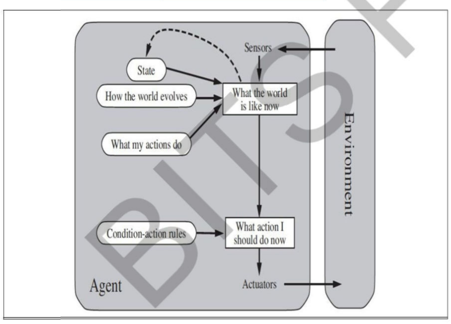


Agent Architectures

Model based Agent

Simple Reflex Agents

Model Based Agents



Agent Architectures

Model based Agent

function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
persistent: *state*, the agent's current conception of the world state
transition model, a description of how the next state depends on the current state and action
sensor model , a description of how the current world state is reflected in the agent's percepts
rules, a set of condition–action rules
action, the most recent action, initially none

state←UPDATE-STATE(*state, action, percept, transition model, sensor model*)
rule←RULE-MATCH(*state, rules*)
action ←*rule*.ACTION
return *action*

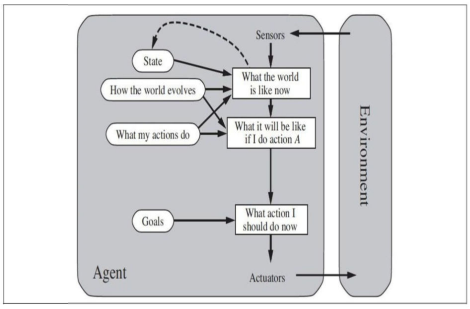
Agent Architectures

Goal based Agent

Simple Reflex Agents

Model Based Agents

Goal Based Agents



Agent Architectures

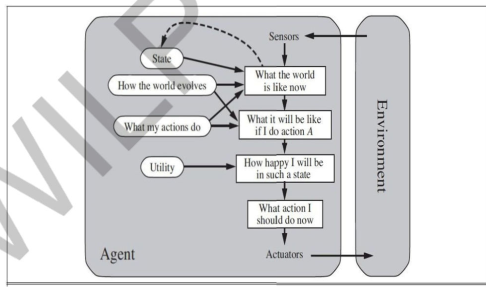
Utility based Agent

Simple Reflex Agents

Model Based Agents

Goal Based Agents

Utility Based Agents



Agent Architectures

Learning Agent

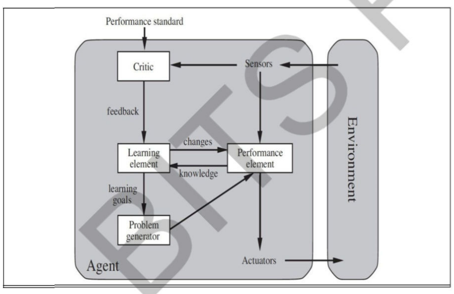
Simple Reflex Agents

Model Based Agents

Goal Based Agents

Utility Based Agents

Learning Agents



Role of Learning

Performance Element – taking a decision of action based on percepts

Learning Element – Make the performance element select better actions such that the utility function is optimized

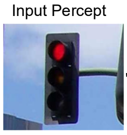
Critic – Provides feedback on the actions taken

Problem Generator – Make the Performance Element select sub-optimal actions such that you would learn from unseen actions

Role of Learning

Agents that improve their performance by learning from their own experiences

Input Percept



Performance Element

Output Action

Resulting State

Critic

Feedback

-2

Role of Learning

Input Percept

Possible Actions

Brake
Change Gear to Lower
Change Gear to Higher
Accelerate
Steer left
Steer right

Random

Selected Action

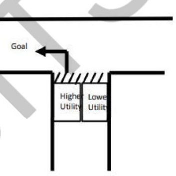
Change Gear to Lower

Role of Learning

<u>Performance Element</u> – Takes decision on action based on percept

$$f(red\ signal,\quad distance) = 15k\ N\ brake$$
$$distance = f'(percept\ sequence)$$
$$f(percepts, distance, raining)$$

- $f(state_0, actionA) = 0.83,$
- $f(state_0, actionB) = 0.45$



Role of Learning

Learning : Supervised Vs Unsupervised Vs Reinforcement

Input Percept

at 50 mts away

Performance Element*

Output Action

Force of **17k** N

Input Percept

at 50 mts away

Performance Element

Output Action

Force of 15k N

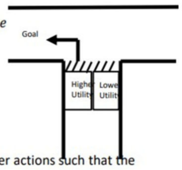
+ Critic Feedback -2

Role of Learning

Performance Element – Takes decision on action based on percept

$$f(red\ signal,\quad distance) = 15k\ N\ brake$$
$$distance = f'(percept\ sequence)$$
$$f(percepts, distance, raining)$$

- $f(state_0, actionA) = 0.83,$
- $f(state_0, actionB) = 0.45$



Learning Element – Make the performance element select better actions such that the utility function is optimized

Critic – Provides feedback on the actions taken

Problem Generator – Make the Performance Element select sub-optimal actions such that you would learn from unseen actions

Problem Formulation

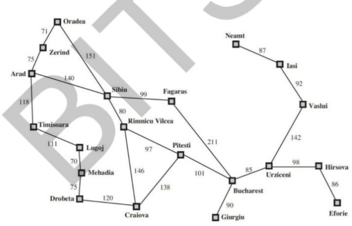
Problem Solving Agents

Goal based decision making agents finds sequence of actions that leads to the desirable state.

Phases of Solution Search by PSA

Goal Formulation → Problem Formulation → Search Phase → Execution Phase

Optimizes the Objective (Local | Global) Limits the Actions

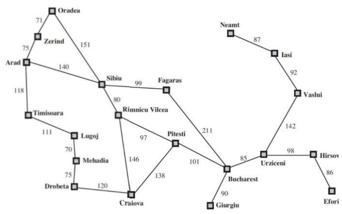


Problem Solving Agents

Phases of Solution Search by PSA

Goal Formulation → Problem Formulation → Search Phase → Execution Phase

State Space Creations [in the path of Goal]
Lists the Actions

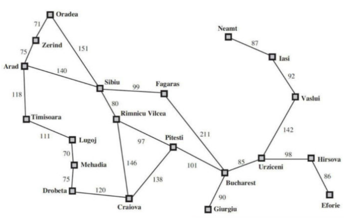


Problem Solving Agents

Phases of Solution Search by PSA

Goal Formulation → Problem Formulation → Search Phase → Execution Phase

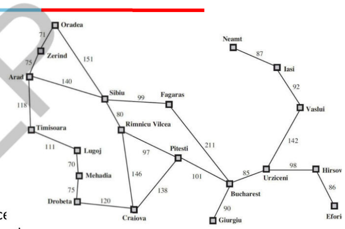
Assumptions – Environment :
Static
Observable Discrete
Deterministic



Problem Solving Agents

Phases of Solution Search

Goal Formulation → Problem Formulation → Search Phase → Execution Phase



Examine all sequence
Choose best | Optimal

Problem Solving Agents – Problem Formulation

Abstraction Representation
Decide what actions under states to take to achieve a goal

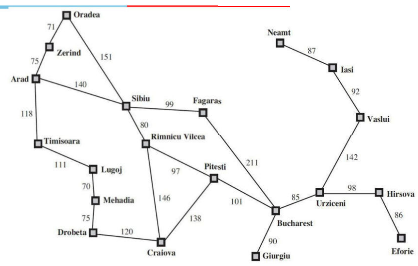
5 Components

- Initial State
- Possible Action | Operators
- Successor Function | Transition Model
- Goal Test
- Path Cost

A function that assigns a numeric cost to each path. A path is a series of actions.
Each action is given a cost depending on the problem.

Solution = Path Cost Function + Optimal Solution

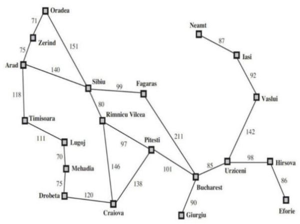
Problem Solving Agents – Problem Formulation: Book Example



Initial State –E.g., *In(Arad)*
Possible Actions – *ACTIONS(s)* → *{Go(Sibiu), Go(Timisoara), Go(Zerind)}*
Transition Model – *RESULT(In(Arad), Go(Sibiu)) = In(Sibiu)*
Goal Test – *IsGoal(In(Bucharest)) = Yes*
Path Cost – *cost(In(Arad), go(Sibiu)) = 140 kms*

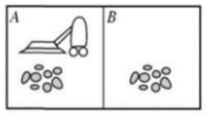
Example Problem Formulation

	Travelling Problem
Initial State	Based on the problem
Possible Actions	Take a flight | Train | Shop
Transition Model/ Successor Function	[A, Go(A->S)] = [S]
Goal Test	Is current = B (destination)
Path Cost	Cost + Time + Quality



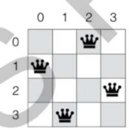
Example Problem Formulation

	Vacuum World
Initial State	Any
Possible Actions	[Move Left, Move Right, Suck, NoOps]
Transition Model/ Successor Function	[A, ML] = [B , Dirty] [A, ML] = [B, Clean]
Goal Test	Is all room clean? [A, Clean] [B, Clean]
Path Cost	No of steps in path



Example Problem Formulation

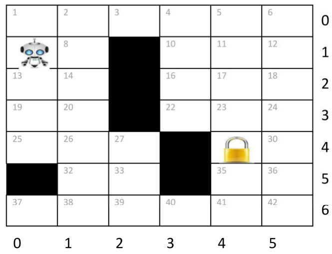
	N-Queen
Initial State	Empty | Partial | Full
Possible Actions	
Transition Model/ Successor Function	
Goal Test	
Path Cost	



board[r][c]

Path finding Robot

Successor Function Design



N-W-E-S

Graph Searching

- Graph as state space (node = state, edge = action)
- For example, game trees, mazes, ...

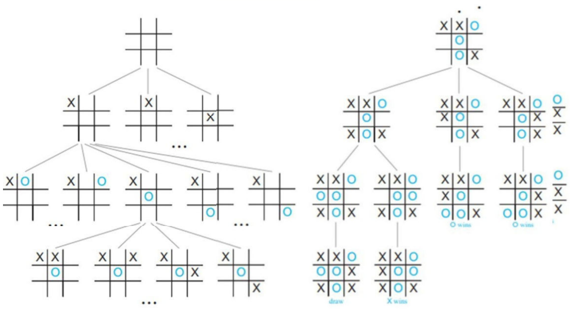
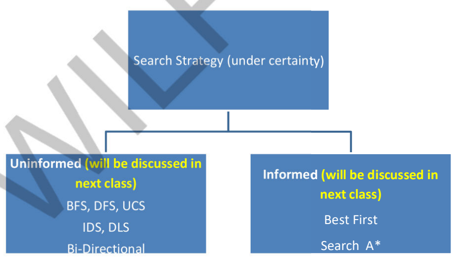
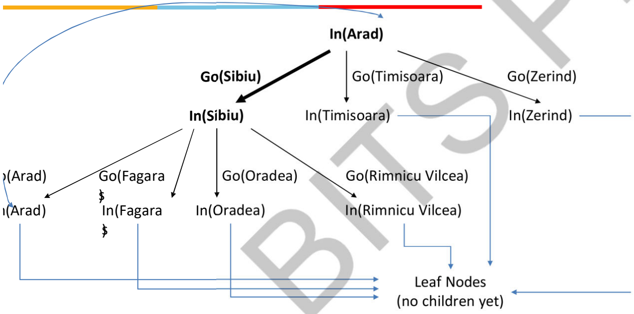


FIGURE 8 Some of the Game Tree for Tic-Tac-Toe.

Searching for Solutions

Choosing the current state, testing possible successor function, expanding current state to generate new state is called Traversal. Choice of which state to expand – Search Strategy





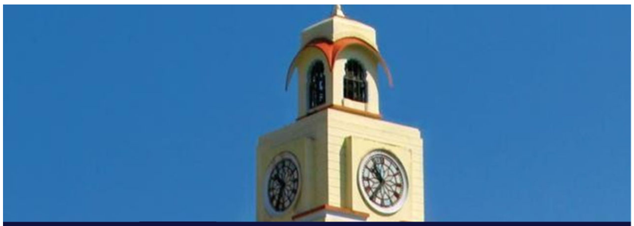
Next Class Plan

- Uninformed Search Algorithms
  - BFS vs DFS – An overview
  - Uniform Cost Search
  - Iterative Depth First Search
- Informed Search Algorithms
  - Greedy Best First search
  - A* Search (Start)

Required Reading: **AIMA - Chapter #1, 2, 3.1, 3.2, 3.3**

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials



Artificial and Computational Intelligence

AIMLCZG557

Contributors & Designers of document content : Cluster Course Faculty Team

M2 : : Problem Solving Agent using Search

BITS Pilani
Pilani Campus

Presented by
Faculty Name
BITS Email ID

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

Disclaimer

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

M1 Introduction to AI

M2 Problem Solving Agent using Search

M3 Game Playing

M4 Knowledge Representation using Logics

M5 Probabilistic Representation and Reasoning

M6 Reasoning over time

M7 Ethics in AI

Learning Objective

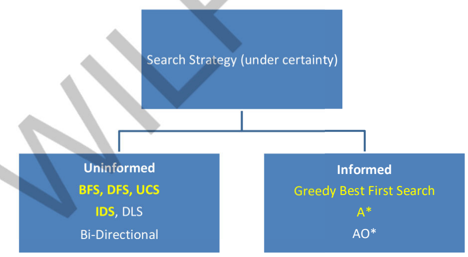
At the end of this class , students Should be able to:

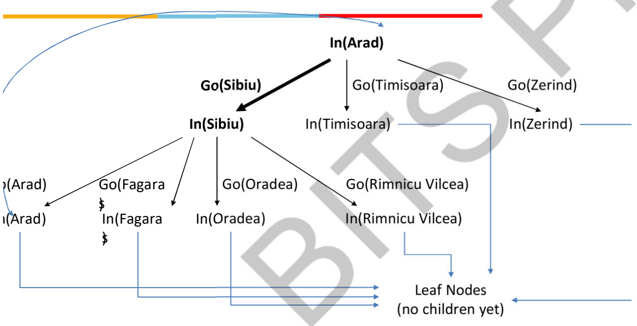
1. Design problem solving agents
2. Create search tree for given problem
3. Apply uninformed search algorithms to the given problem
4. Compare performance of given algorithms in terms of completeness, optimality, time and space complexity
5. Differentiate for which scenario appropriate uninformed search technique is suitable and justify.
6. Differentiate between Tree and Graph search

Problem Formulation

Searching for Solutions

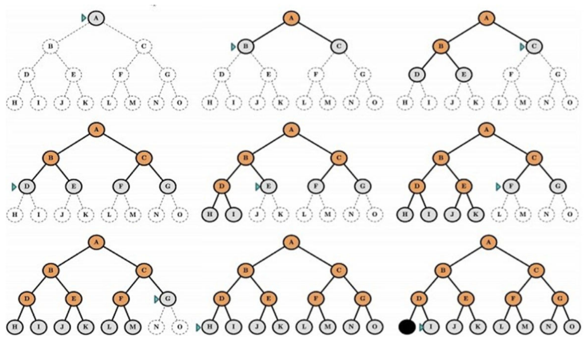
Choosing the current state, testing possible successor function, expanding current state to generate new state is called Traversal. Choice of which state to expand – Search Strategy



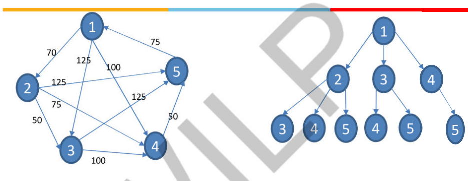


Uninformed Search – BFS & its Variant

Breadth First Search (BFS)



BFS – Uninformed



(1)
(1 2) (1 3) (1 4)
TEST FAILED
:
:
(1 3) (1 4) (1 2 3) (1 2 4) (1 2 5)
(1 2 3) (1 2 4) (1 2 5) (1 3 4) (1 3 5) (1 4 5)
TEST PASSED

C(1-2-5) = 70 + 125 =195
Expanded : 4
Generated : 10
Max Queue Length : 6

Breadth First Search – Evaluation

Depth	Nodes	Time	Memory
2	110	.11 milliseconds	107 kilobytes
4	11,110	11 milliseconds	10.6 megabytes
6	10^6	1.1 seconds	1 gigabyte
8	10^8	2 minutes	103 gigabytes
10	10^{10}	3 hours	10 terabytes
12	10^{12}	13 days	1 petabyte
14	10^{14}	3.5 years	99 petabytes
16	10^{16}	350 years	10 exabytes

Why is Space Complexity a big problem? Imagine a problem with
- branching factor b = 10
- generates 1 million nodes/sec
- Each node requires 1KB

Breadth First Search – Evaluation

Complete – If the shallowest goal node is at a depth d, BFS will eventually find it by generating all shallower nodes

Optimal – Not necessarily. Optimal if path cost is non-decreasing function of depth of node. E.g., all actions have same cost

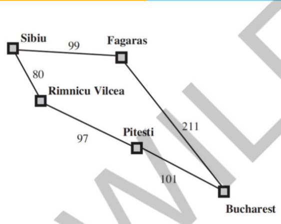
Time Complexity – $O(b^d)$ b - branching factor, d – depth
- Nodes expanded at depth 1 = b
- Nodes expanded at depth 2 = b^2
- Nodes expanded at depth d = b^d
- Goal test is applied during generation, time complexity would be $O(b^{d+1})$

Space Complexity – $O(b^d)$
- $O(b^{d-1})$ in explored set
- $O(b^d)$ in frontier set

Uniform Cost Search

- Instead of expanding the shallowest node, Uniform-Cost search expands the node n with the lowest path cost g(n)
- Sorting the Frontier as a priority queue ordered by g(n)
- Goal test is applied during expansion
  - The goal node if generated may not be on the optimal path
  - Find a better path to a node on the Frontier

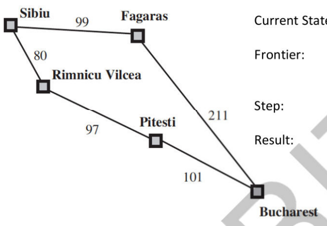
Uniform Cost Search



Current State: Sibiu
Frontier: []
Step: Expand Sibiu
Result: Generates ("Rimnicu Vilcea" 80)
("Fagaras", 99)
Add to Frontier

Initial State: Sibiu
Goal State: Bucharest

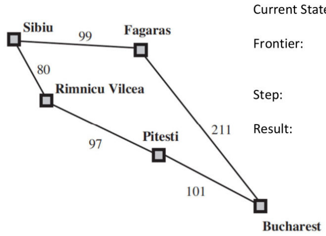
Uniform Cost Search



Current State: Sibiu
Frontier: [("Rimnicu Vilcea" 80)
("Fagaras", 99)]
Step: Expand "Rimnicu Vilcea" (least cost)
Result: Generates ("Pitesti", 177)
Add to Frontier

Initial State: Sibiu
Goal State: Bucharest

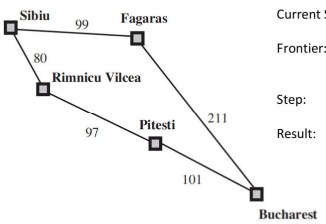
Uniform Cost Search



Current State: Rimnicu Vilcea (not a Goal state
Frontier: [("Fagaras", 99)
("Pitesti", 177)]
Step: Expand "Fagaras" (least cost)
Result: Generates ("Bucharest", 310)
Add to Frontier
(It's a Goal State but we won't test during generation)

Initial State: Sibiu
Goal State: Bucharest

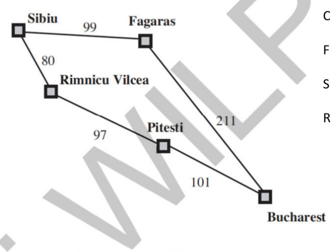
Uniform Cost Search



Current State: Fagaras (not a goal state)
Frontier: [("Pitesti", 177)
("Bucharest", 310)]
Step: Expand "Pitesti" (least cost)
Result: Generates ("Bucharest", 278)
Replace in Frontier
(It's a Goal State but we won't test during generation)

Initial State: Sibiu
Goal State: Bucharest

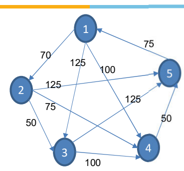
Uniform Cost Search

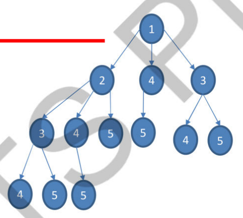


Current State: Pitesti (not a goal state)
Frontier: [("Bucharest", 278)]
Step: Expand "Bucharest"
Result: No further generation as Goal Test satisfied
Return Solution

Initial State: Sibiu
Goal State: Bucharest

UCS





(1)
(1 2 : 70) (1 4 : 100) (1 3 : 125)
TEST-F
(1 4 : 100) (1 2 3 :120)(1 3 : 125) (1 2 4 : 145) (1 2 5 : 195)
TEST-F
(1 2 3 :120)(1 3 : 125) (1 2 4 : 145) (1 4 5: 150)(1 2 5 : 195)
TEST-F
(1 3 : 125) (1 2 4 : 145) (1 4 5: 150) (1 2 3 4:170) (1 2 5 : 195) (1 2 3 5 : 245)
TEST-F
(1 2 4 : 145) (1 4 5: 150) (1 2 3 4:170) (1 2 5 : 195) (1 3 4 : 225) (1 2 3 5 : 245) (1 3 5 : 250)
TEST-F
(1 4 5: 150) (1 2 3 4:170) (1 2 4 5 : 195) (1 2 5 : 195) (1 3 4 : 225) (1 2 3 5 : 245) (1 3 5 : 250)
TEST - P

Uniform Cost Search – Evaluation

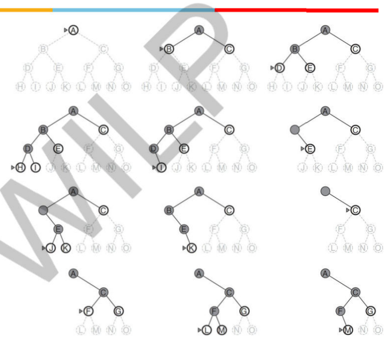
Completeness – It is complete if the cost of every step > small +ve constant ∈
- It will stuck in infinite loop if there is a path with infinite sequence of zero cost actions

Optimal – It is Optimal. Whenever it selects a node, it is an optimal path to that node.

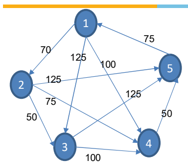
Time and Space complexity – Uniform cost search is guided by path costs not depth or branching factor.
- If C* is the cost of optimal solution and ∈ is the min. action cost
- Worst case complexity = $O(b^{1+\frac{C^*}{\epsilon}})$,
- When all action costs are equal → $O(b^{d+1})$, the BFS would perform better
  - As Goal test is applied during expansion, Uniform Cost search would do extra

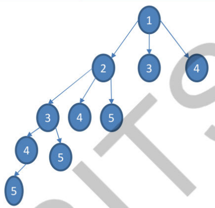
Uninformed Search – DFS & its Variant

Depth First Search (DFS)



DFS – Uninformed





(1)
(1 2) (1 3) (1 4)
(1 2 3) (1 2 4) (1 2 5) (1 3) (1 4)
(1 2 3 4) (1 2 3 5) (1 2 4) (1 2 5) (1 3) (1 4)
(1 2 3 4 5) (1 2 3 5) (1 2 4) (1 2 5) (1 3) (1 4)

C(1-2-3-4-5) = 70 + 50 + 100 + 50 =270
Expanded : 4
Generated : 10
Max Queue Length : 6

Depth First Search (DFS)

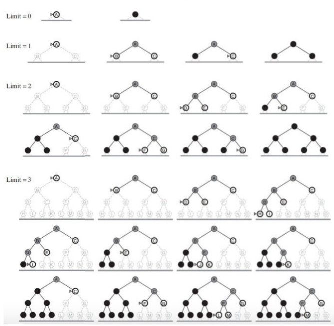
Completeness – Complete in finite state spaces because it will eventually expand every node

Optimal – Not Optimal as it would stop when the goal node is reached without evaluating if there is a better path

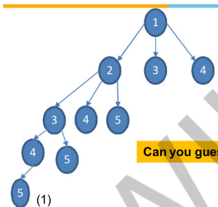
Time Complexity - $O(b^m)$ where m = maximum depth of any node
- Can be much larger than the size of state space
- m can be much larger than d (shallowest goal)

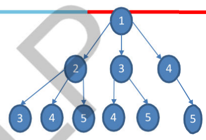
Space Complexity – Needs to store only one path and unexpanded siblings.
- Any node expanded with all its children can be removed from memory
- Requires storage of only $O(bm)$, b – branching factor, m - max depth

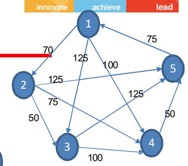
Iterative Deepening Depth First Search (IDS)



Search Algorithm – Uninformed Example







Can you guess which algorithm are these ?

(1)
(1 2) (1 3) (1 4)
(1 2 3) (1 2 4) (1 2 5) (1 3) (1 4)
(1 2 3 4) (1 2 3 5) (1 2 4) (1 2 5) (1 3) (1 4)
(1 2 3 4 5) (1 2 3 5) (1 2 4) (1 2 5) (1 3) (1 4)

C(1-2-3-4-5) = 70 + 50 + 100 + 50 =270
Expanded : 4
Generated : 10
Max Queue Length : 6

(1)
(1 2) (1 3) (1 4)
TEST FAILED

(1 3) (1 4) (1 2 3) (1 2 4) (1 2 5)
(1 2 3) (1 2 4) (1 2 5) (1 3 4) (1 3 5) (1 4 5)
TEST PASSED

C(1-2-5) = 70 + 125 =195
Expanded : 4
Generated : 10
Max Queue Length : 6

Iterative Deepening Depth First Search (IDS)

Run Depth Limited Search (DLS) by gradually increasing the limit l
- First with l=1, then l=2, l=3 and so on – until goal is found

Its is a combination of Depth First Search + Breadth First Search

Like DFS, memory requirement is a modest $O(bd)$ where d is the depth of shallowest goal

Like BFS
- Complete when branching factor is finite
- Optimal when path cost is non decreasing function of depth

Iterative Deepening Depth First Search (IDS)

Time Complexity:
- Appears that IDS is generating a lot of nodes multiple times
- However, most of nodes are present in the lower levels which are not repeated often
- Generation of nodes
  - At level 1 - b nodes generated d times – (d)b
  - At level 2 – b^2 nodes generated d-1 times – (d-1)b^2
  - At level d – b^d nodes generated once – (1) b^d
- Time Complexity N(IDS) = $O(b^d)$ same as BFS

IDS is the preferred uninformed search method when search space is large and depth is unknown

Application

Breadth First Search

- Finding path in a graph (many solutions)
- Finding the Bipartitions in a graph

Depth First Search

- Find the Connectedness in a graph
- Topological Sorting

Algorithm Tracing

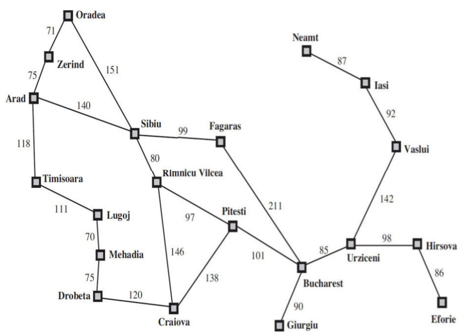
Students must follow this in the exams for all the search algorithms in addition to the search tree constructions. The ordering of the Open Lists must be in consistent with the algorithm with a note on the justification of the order expected!

Iter	Open List / Frontiers / Fringes	Closed List	Goal Test
1.	(1)		Fail on (1)
2.	(1 3), (1 4), (1 2)	(1)	Fail on (1 3)

Terminologies – Learnt Today

- Nodes
- States
- Frontier | Fringes
- Search Strategy : LIFO | FIFO | Priority Queue
- Performance Metrics
  - Completeness
  - Optimality
  - Time Complexity
  - Space Complexity
- Algorithm Terminology
  - d Depth of a node
  - b Branching factor
  - n – nodes
  - l – level of a node
  - m – maximum
  - C* - Optimal Cost
  - E – least Cost
  - N –total node generated

Tree Search Vs Graph Search

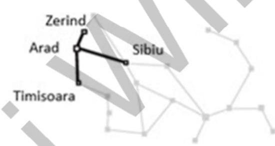


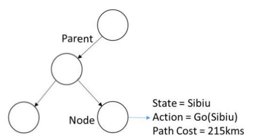
Search

Coding Aspects

For each node n of the tree,

- **n.STATE** : the state in the state space to which node corresponds
- **n.PARENT** : the node in the search tree that generated this node
- **n.ACTION** : the action that was applied to parent to generate the node
- **n.PATH-COST** : the cost, denoted by g(n), of the path from initial state to node





Algorithm Tracing

Students must follow this in the exams for all the search algorithms in addition to the search tree constructions. The ordering of the Open Lists must be in consistent with the algorithm with a note on the justification of the order expected!

Iter	Open List / Frontiers / Fringes	Goal Test
1.	(1)	Fail on (1)
2.	(1 3), (1 4), (1 2)	Fail on (1 3)

Tree Search Algorithms

```
function Tree-Search (problem, strategy) returns a solution, or failure
    initialize the search tree using the initial state of problems
    loop do
        if there are no candidate for expansion
            then return failure
        choose: leaf node for expansion according to strategy
        if the node contains a goal state
            then return the corresponding solution
        else
            Expand the node
            Add the resulting nodes to the search tree
    end
```

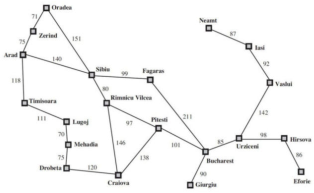
Tree Search Vs Graph Search Algorithms

Coding Aspects

Need:

Redundant Path Problem : More than one way to reach a state from another.

Infinite Loop Path Problem



Start : Arad

Goal : Craiova

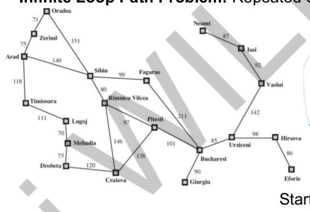
Tree Search Vs Graph Search Algorithms

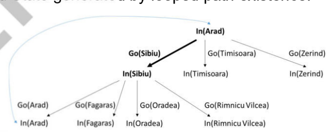
Coding Aspects

Need:

Redundant Path Problem

Infinite Loop Path Problem: Repeated State generated by looped path existence.





Start : Arad

Goal : Craiova

Algorithm Tracing

Students must follow this in the exams for all the search algorithms in addition to the search tree constructions. The ordering of the Open List must be in consistent with the algorithm with a note on the justification of the order expected!

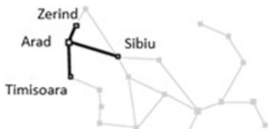
Iter	Open List / Frontiers / Fringes	Closed List	Goal Test
1.	(1)		Fail on (1)
2.	(1 3), (1 4), (1 2)	(1)	Fail on (1 3)

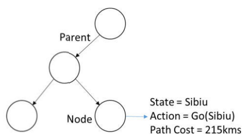
Search

Coding Aspects

For each node n of the tree,

- **n.STATE** : the state in the state space to which node corresponds
- **n.PARENT** : the node in the search tree that generated this node
- **n.ACTION** : the action that was applied to parent to generate the node
- **n.PATH-COST** : the cost, denoted by g(n), of the path from initial state to node
- **n.VISITED** : the boolean indicating if the node is already visited and tested (**or** a global SET of visited nodes



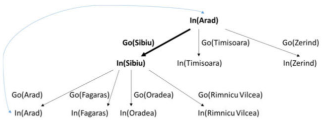


Tree Search Vs Graph Search Algorithms

Coding Aspects

Graph-Search Algorithm

Augments the Tree-Search algorithm to solve redundancy by keeping track of states that are already visited called as **Explored Set**. **Only one copy of each state is maintained/stored.**



Graph Search Algorithms

```
function Graph-Search (problem, fringe) returns a solution, or failure
    initialize the search space using the initial state of problems memory to store
the visited fringe
    closed ← an empty set
    fringe ← Insert(Make-Node(Initial-State[problem]), fringe)
    loop do
        if fringe is empty
            then return failure
        node ← Remove-Front(fringe)
        if the node contains a goal state
            then return the corresponding solution
        else
            if the node is not in closed ie., not visited yet
                Add the node to the closed set
                Expand all the fringe of the node
                Add all expanded sorted successors into the fringe
    end
```


Required Reading: AIMA - Chapter #3: 3.1, 3.2, 3.3, 3.4

Next Class Plan :
Informed Search : GFBS & A*
Heuristic Design

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials

Artificial and Computational Intelligence

AIMLCZG557

Contributors & Designers of document content : Cluster Course Faculty Team

M2 : Problem Solving Agent using Search

BITS Pilani
Pilani Campus

Presented by
Faculty Name
BITS Email ID

---

innovate achieve lead

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

Disclaimer

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

---

innovate achieve lead

Course Plan

M1 Introduction to AI

M2 Problem Solving Agent using Search

M3 Game Playing

M4 Knowledge Representation using Logics

M5 Probabilistic Representation and Reasoning

M6 Reasoning over time

M7 Ethics in AI

---

innovate achieve lead

Learning Objective

At the end of this class , students Should be able to:

1. Create Search tree for given problem
2. Differentiate between uninformed and informed search requirements
3. Apply GBFS & A* algorithms to the given problem
4. Prove if the given heuristics are admissible and consistent
5. Apply A* variations algorithms to the given problem

---

innovate achieve lead

Module 2 : Problem Solving Agent using Search

A. Uninformed Search

B. Informed Search

C. Heuristic Functions

D. Local Search Algorithms & Optimization Problems

---

innovate achieve lead

Informed Search
Greedy Best First
A*

---

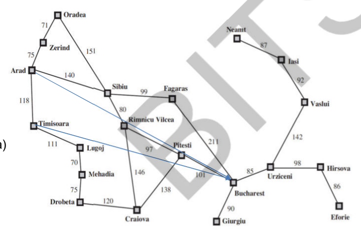
innovate achieve lead

Informed /Heuristic Search

Strategies that know if one non-goal state is more promising than another non-goal state

h(Arad) = 366

h(Timisoara) = 329



---

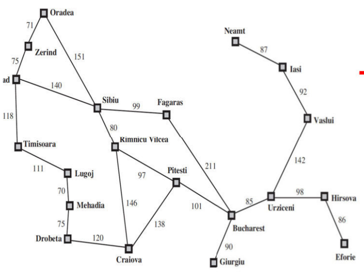
innovate achieve lead

Greedy Best First Search

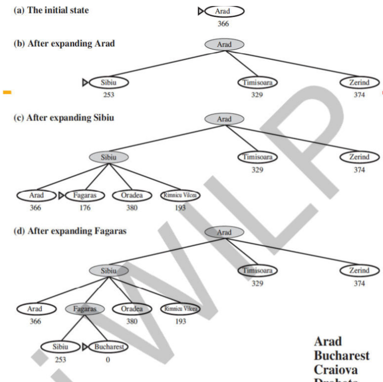
Expands the node that is closest to the goal

Thus, f(n) = h(n)

Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374



Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374



Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

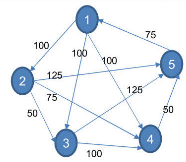
Greedy Best First Search

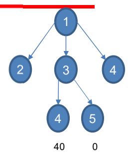
Not Optimal
- Because the algorithm is greedy
- It only optimizes for the current action

Not Complete
- Often ends up in state with a dead end as the heuristic doesn't guarantee a path but is only an approximation

Time and Space Complexity - $O(b^m)$ where m – max depth of search tree

Greedy Best First Search



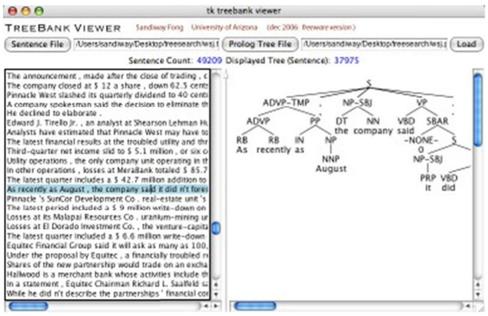


n	h(n)
1	60
2	120
3	30
4	40
5	0

(1)
(1 3) (1 4) (1 2)
(1 3 5) (1 3 4)

C(1-3-5) = 100 + 125 =225
Expanded : 2
Generated : 6
Max Queue Length : 3
Idea: Optimize DFS. Choose next nearest to goal in the same hill.

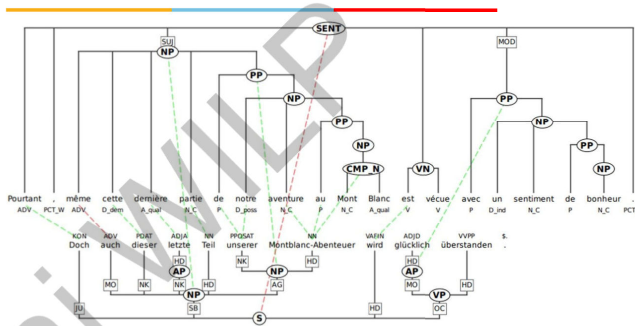
Case Study – 1 Search in Treebanks



Source Credit :
https://catalog.ldc.upenn.edu/docs/LDC95T7/cl93.html
https://ufal.mff.cuni.cz/pdt3.5

BITS Pilani, Pilani Campus

Case Study – 1 Search in Treebanks



Source Credit :
https://catalog.ldc.upenn.edu/docs/LDC95T7/cl93.html
https://ufal.mff.cuni.cz/pdt3.5

BITS Pilani, Pilani Campu

A* Search

Expands the node which lies in the closest path (estimated cheapest path) to the goal

Evaluation function **f(n) = g(n) + h(n)**

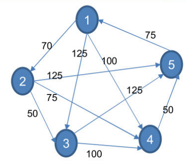
g(n) – the cost to reach the node

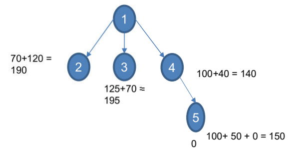
h(n) – the expected cost to go from node to goal

f(n) – estimated cost of cheapest path through node n

Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

A* Search





n	h(n)
1	60
2	120
3	70
4	40
5	0

(1)
(1 4) (1 2) (1 3)
(1 4 5) (1 2) (1 3)

C(1-4-5) = 100 + 150 =150
Expanded : 2
Generated : 5
Max Queue Length : 3

Optimality of A*

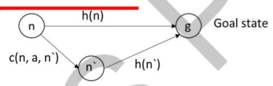
A* Search

Test for Admissibility

Expands the node which lies in the closest path (estimated cheapest path) to the goal
Evaluation function **f(n) = g(n) + h(n)**
g(n) – the cost to reach the node
h(n) – the expected cost to go from node to goal
f(n) – estimated cost of cheapest path through node n

A heuristic is admissible or optimistic if , **0 ≤ h(n) ≤ h*(n)**, where h*(n) is the actual cost to reach the goal

A* Search



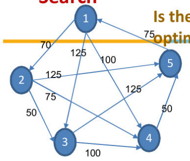
Optimal on condition
h(n) must satisfies two conditions:
- Admissible Heuristic – one that never overestimates the cost to reach the goal
- Consistency – A heuristic is consistent if for every node n and every successor node n` of n generated by action a, h(n) <= c(n, a, n`) + h(n`)

Complete
- If the number of nodes with cost <=C* is finite
- If the branching factor is finite
- A* expands no nodes with f(n) > C*, known as pruning

Time Complexity - $O(b^{\Delta})$ where the absolute error $\Delta = h^* - h$

A* Search

Is theheuristicdesigned leads to optimalsolution?

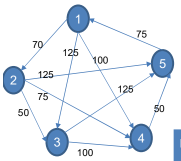


Assuming node 3 as goal, taking only sample edges per node below is checked for consistency

		n	h(n)	Is Admissible? h(n) <= h*(n)	Is Consistent? For every arc (i,j): h(i) <= g(i,j) + h(j)
		1	80		
		2	60		
		3	0		
		4	200		
		5	190		

A* Search

Is the heuristic designed leads to optimal solution?



Assuming node 3 as goal, taking only sample edges per node below is checked for consistency

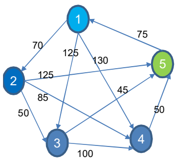
n	h(n)	Is Admissible? h(n) <= h*(n)	Is Consistent? For every arc (i,j): h(i) <= g(i,j) + h(j)
1	80	Y	N (5→ 1) : 190<=155
2	60	N	Y (1→ 2) : 80<=130
3	0	Y	
4	200	Y	Y (1→ 4) : 80<=300 Y (2→ 4) : 60<=275
5	190	Y	Y (2→ 5) : 60<=315 Y (4→ 5) : 200<=240

Variations of A*

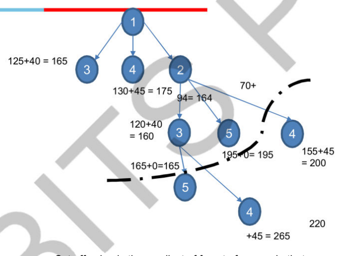
Memory Bounded Heuristics

Iterative Deepening A*

Set limit for f(n)



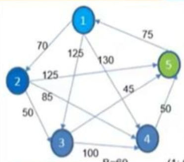
n	h(n)
1	60
2	94
3	40
4	45
5	0

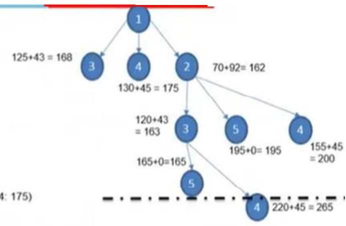


Cut off value is the smallest of f-cost of any node that exceeds the cutoff on previous iterations

Iterative Limit : Eg
f(n) = 180
f(n) = 195
f(n) = 200
.
.
.
.

Iterative Deepening A*





n	h(n)
1	60
2	92
3	43
4	45
5	0

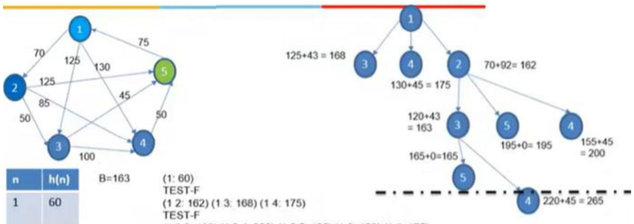
B=60
(1: 60)
TEST-F
(1 2: 162) (1 3: 168) (1 4: 175)

B=162
(1: 60)
TEST-F
(1 2: 162) (1 3: 168) (1 4: 175)
TEST-F
(1 2 3: 163) (1 2 4: 200) (1 2 5: 195) (1 3: 168) (1 4: 175)

B=163
(1: 60)
TEST-F
(1 2: 162) (1 3: 168) (1 4: 175)
TEST-F
(1 2 3: 163) (1 2 4: 200) (1 2 5: 195) (1 3: 168) (1 4: 175)
TEST-F

Iterative Deepening A*

Set limit for f(n)



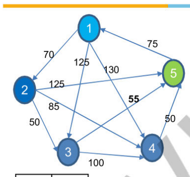
n	h(n)
1	60
2	92
3	43
4	45
5	0

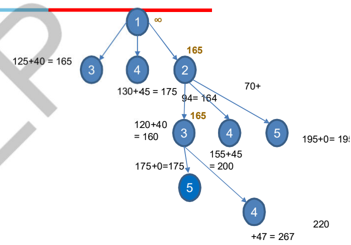
B=163
(1: 60)
TEST-F
(1 2: 162) (1 3: 168) (1 4: 175)
TEST-F
(1 2 3: 163) (1 2 4: 200) (1 2 5: 195) (1 3: 168) (1 4: 175)
TEST-F
(1 2 3 5: 165) (1 2 3 4: 265) (1 2 4: 200) (1 2 5: 195) (1 3: 168) (1 4: 175)

B=165
(1: 60)
TEST-F
(1 2: 162) (1 3: 168) (1 4: 175)
TEST-F
(1 2 3: 163) (1 2 4: 200) (1 2 5: 195) (1 3: 168) (1 4: 175)
TEST-F
(1 2 3 5: 165) (1 2 3 4: 265) (1 2 4: 200) (1 2 5: 195) (1 3: 168) (1 4: 175)

Recursive Best First Search A*

Remember the next best alternative f-Cost to regenerate

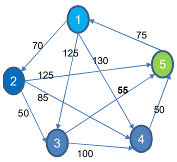


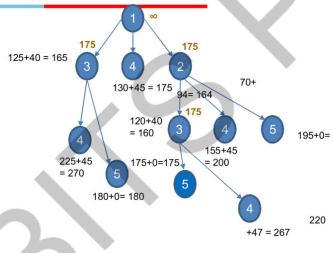


n	h(n)
1	60
2	94
3	40
4	45
5	0

Recursive Best First Search A*

Remember the next best alternative f-Cost to regenerate

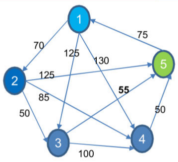


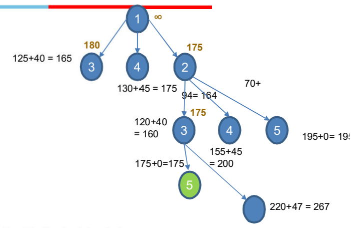


n	h(n)
1	60
2	94
3	40
4	45
5	0

Recursive Best First Search A*

Remember the next best alternative f-Cost to regenerate





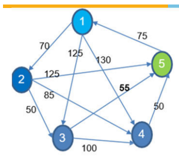
n	h(n)
1	60
2	94
3	40
4	45
5	0

If the current best leaf value > best alternative path
Best leaf value of the forgotten subtree is backed up to the ancestors
Recursion unwinds
Else
Continue expansion

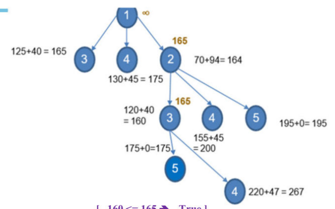
Space Usage = O(bd) very less

Recursive Best First Search A*

Remember the next best alternative f-Cost to regenerate



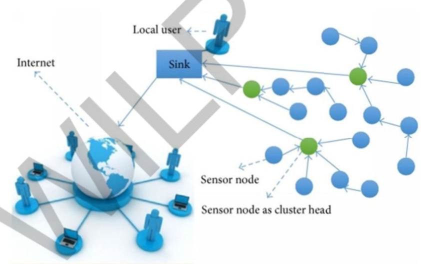
n	h(n)
1	60
2	94
3	40
4	45
5	0



1, 60)
1 2 | 164) (1 3 | 165) (1 4 | 175)
1 2 3 | 160) (1 3 | 165) (1 4 | 175) (1 2 5 | 195) (1 2 4 | 200) [160 <= 165 ➔ True]
1 2 3 5 | 175) (1 3 | 165) (1 4 | 175) (1 2 5 | 195) (1 2 4 | 200) (1 2 3 4 | 265) [175 <= 165 ➔ False]
1 3 | 165) (1 2 | 175) (1 4 | 175) [165 <= 175 ➔ True]
1 3 5 | 180) (1 2 | 175) (1 4 | 175) (1 3 4 | 270) [185 <= 175 ➔ False]
(1 2 | 175) (1 4 | 175) (1 3 | 180)
1 2 3 | 160) (1 4 | 175) (1 3 | 180) (1 2 5 | 195) (1 2 4 | 200)
1 2 3 5 | 175) (1 4 | 175) (1 3 | 180) (1 2 5 | 195) (1 2 4 | 200) (1 2 3 4 | 267)

PASS

Case Study – Search in Network Routing



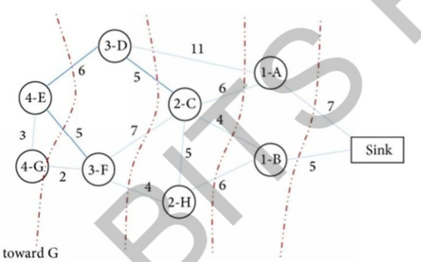
Source Credit :

AR-RBFS: Aware-Routing Protocol Based on Recursive Best-First Search Algorithm for Wireless Sensor Networks
https://doi.org/10.1155/2016/8743927

Case Study – Search in Network Routing

A	14.1
B	11.3
C	8.2
H	6.6
F	2
E	3
D	4.8

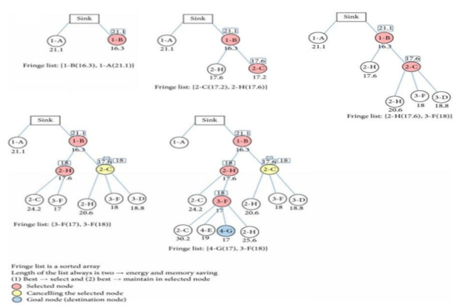
Heuristic values toward G



Source Credit :

AR-RBFS: Aware-Routing Protocol Based on Recursive Best-First Search Algorithm for Wireless Sensor Networks
https://doi.org/10.1155/2016/8743927

Case Study – Search in Network Routing



Source Credit :

AR-RBFS: Aware-Routing Protocol Based on Recursive Best-First Search Algorithm for Wireless Sensor Networks
https://doi.org/10.1155/2016/8743927

Required Reading: AIMA - Chapter #3: 3.1, 3.2, 3.3, 3.4,3.5

Next Class Plan :
Heuristic Design
Local Search Algorithm

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials

Artificial and Computational Intelligence

AIMLCLZG557

Contributors & Designers of document content : Cluster Course Faculty Team

M2 : : Problem Solving Agent using Search

BITS Pilani
Pilani Campus

Presented by
Faculty Name
BITS Email ID

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

Disclaimer

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

M1	Introduction to AI
M2	Problem Solving Agent using Search
M3	Game Playing
M4	Knowledge Representation using Logics
M5	Probabilistic Representation and Reasoning
M6	Reasoning over time, Reinforcement Learning
M7	Ethics in AI

Learning Objective

At the end of this class , students Should be able to:

1. Compare given heuristics for a problem and analyze which is the best fit
2. Design relaxed problem with appropriate heuristic design
3. Prove the designed relaxed problem heuristic is admissible
4. Differentiate which local search is best suitable for given problem
5. Design fitness function for a problem
6. Construct a search tree
7. Apply appropriate local search and show the working of algorithm at least for first 2 iterations with atleast four next level successor generation(if search tree is large)
8. Design and show Genetic Algorithm steps for a given problem

Module 2 : Problem Solving Agent using Search

A. Uninformed Search

B. Informed Search

C. Heuristic Functions

D. Local Search Algorithms & Optimization Problems

Design of Heuristics

Heuristic Design

- **Effective Branching Factor**
- Good Heuristics
- Notion of Relaxed Problems
- Generating Admissible Heuristics

Effective branching factor (b^*):

If the algorithm generates N number of nodes and the solution is found at depth d, then

$$N + 1 = 1 + (b^*) + (b^*)^2 + (b^*)^3 + \ldots + (b^*)^d$$

Heuristic Design

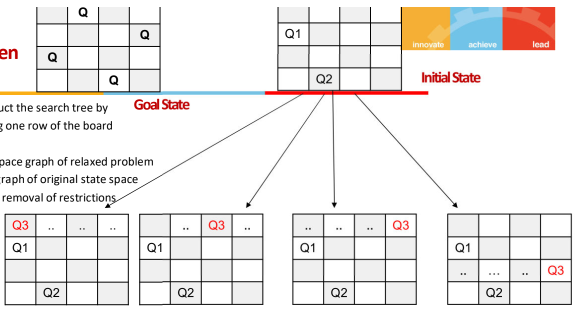
- Effective Branching Factor
- Good Heuristics
- **Notion of Relaxed Problems**
- Generating Admissible Heuristics

Simplify the problem

Assume no constraints

Cost of optimal solution to relaxed problem ≤ Cost of optimal solution for real problem

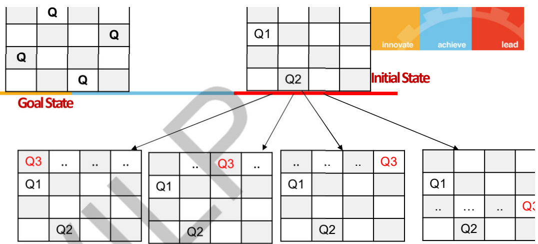
N-Queen



- Construct the search tree by considering one row of the board at a time
- State space graph of relaxed problem is a super graph of original state space because of removal of restrictions

Initial State	Possible Actions	Transition Model	Goal Test	Path Cost	No.Of.States
< Xi , Yi >	Place in any non-occupied row in board		isValid Non-Attacking	Transition + Valid Queens	n!

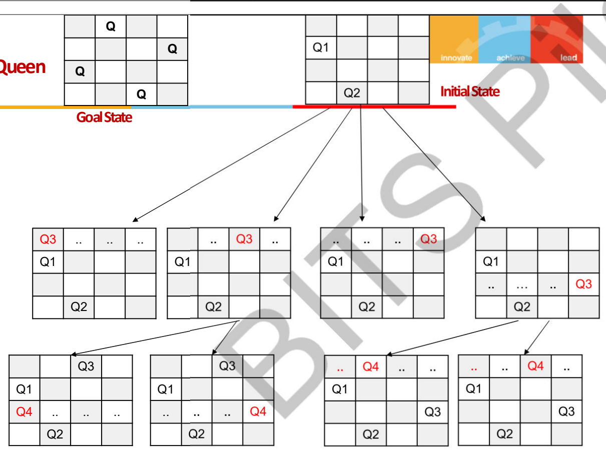
N-Queen



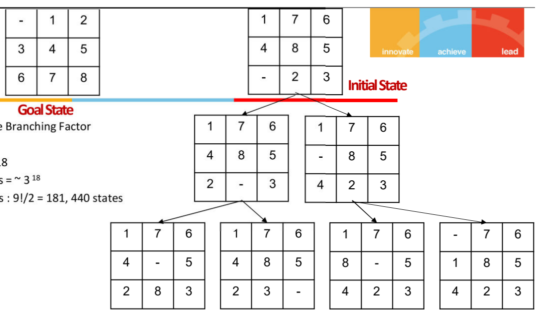
Possible Heuristic Design:

H1(n) : Number of Non conflicting Pairs of Queen (MAX)				
H2(n) : Number of Conflicting Pairs of Queen (MIN)				
H1(n) : Number of safest Queen (MAX)				

N-Queen



N-Tile



- Effective Branching Factor

: ~3

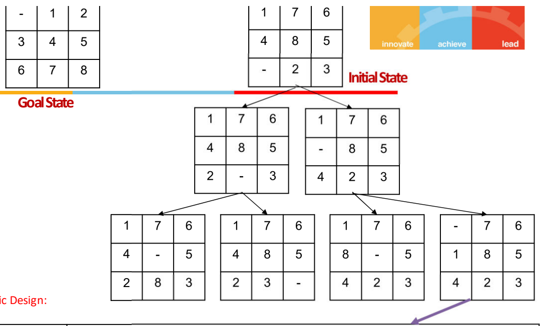
: Avg.cost = 18

: No.of.States = ~ 3^{18}

: Graph states : 9!/2 = 181, 440 states

Initial State	Possible Actions	Transition Model	Goal Test	Path Cost	No.Of.States
<LOC, ID>	Move Empty to near by Tile		ID=LOC+1	Transition + Positional + Distance+ Other approaches	9!

N-Tile



Possible Heuristic Design:

H1(n) : Manhattan distance of Empty tile	
H1(n) : Manhattan distance of all labelled tile	
H1(n) : Number of Misplaced tile (MAX)	

Learn from experience

Trail / Puzzle	X1(n) : No.of.Misplaced Tiles	X2(n): Pair of adjacent tiles that are not in goal	X3(n): Position of the empty tile	…………h`(n)
Example 1	7	10	7	…………
Example 2	5	6	6	…………
………	..	..	..	…………

-	1	2
3	4	5
6	7	8

1	7	6
4	8	5
-	2	3

Create a suitable model:
h(n) = c1*X1(n) + c2*X2(n) + ……….

Local Search & Optimization

Local Search

Optimization Problem

Goal : Navigate through a state space for a given problem such that an optimal solution can be found

Objective : Minimize or Maximize the objective evaluation function value

Scope : Local

Objective Function : Fitness Value evaluates the goodness of current solution

Local Search : Search in the state-space in the neighbourhood of current position until an optimal solution is found

Single Instance Based	Multiple Instance Based
Hill Climbing	Genetic Algorithm
Simulated Annealing	Particle Swarm Optimization
Local Beam Search	Ant Colony Optimization
Tabu Search	

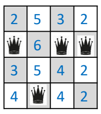
Local Search

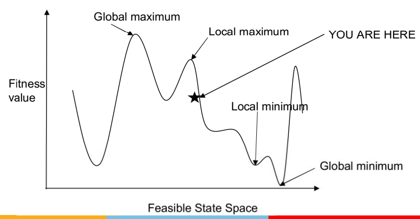
Terminology

Local Search : Search in the state-space in the neighbourhood of current position until an optimal solution is found

Algorithms:

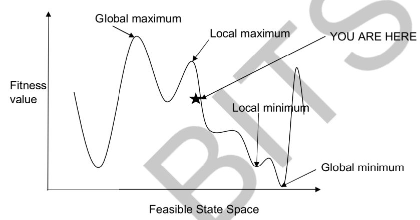
- Choice of Neighbor
- Looping Condition
- Termination Condition





Hill Climbing

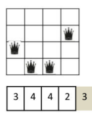
Hill Climbing



Hill Climbing

Random Restart

1. Select a random state
2. Evaluate the fitness scores for all the successors of the state
3. Select the next state based on the highest fitness
4. Repeat from Step 2



function HILL-CLIMBING(*problem*) **returns** a state that is a local maximum

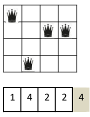
  current ← MAKE-NODE(*problem*.INITIAL-STATE)
  loop do
    neighbor ← a highest-valued successor of *current*
    if neighbor.VALUE ≤ current.VALUE **then return** *current*.STATE
    current ← *neighbor*

Hill Climbing

1. Select a random state
2. Evaluate the fitness scores for all the successors of the state

h(n) = No.of non-conflicting pairs of queens in the board.

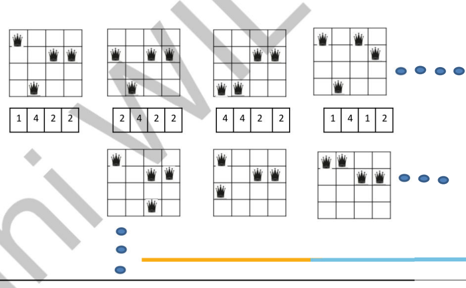
Q1-Q2
Q1-Q3
Q1-Q4
Q2-Q3
Q2-Q4
Q3-Q4



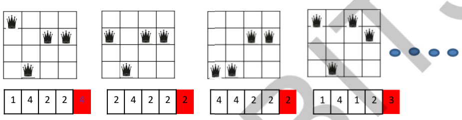
Note : Steps 3 & 4 in the above algorithm will be a part of variation of Hill climbing

Hill Climbing

1. Select a random state
2. Evaluate the fitness scores for all the successors of the state



Hill Climbing

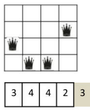


Local Maxima → Random Restart

Hill Climbing

Random Restart

1. Select a random state
2. Evaluate the fitness scores for all the successors of the state
3. Calculate the probability of selecting a successor based on fitness score
4. Select the next state based on the highest probability
5. Repeat from Step 2

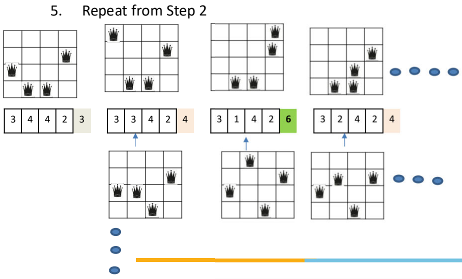


function HILL-CLIMBING(*problem*) **returns** a state that is a local maximum

current ← MAKE-NODE(*problem*.INITIAL-STATE)
loop do
  neighbor ← a highest-valued successor of *current*
  if neighbor.VALUE ≤ current.VALUE **then return** *current*.STATE
  current ← *neighbor*

Hill Climbing

1. Select a random state
2. Evaluate the fitness scores for all the successors of the state
3. Calculate the probability of selecting a successor based on fitness score
4. Select the next state based on the highest probability
5. Repeat from Step 2

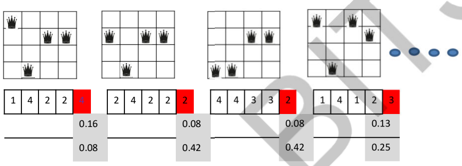


Stochastic Hill Climbing

$next \leftarrow$ a randomly selected successor of $current$
$\Delta E \leftarrow next.\text{VALUE} - current.\text{VALUE}$
if $\Delta E > 0$ **then** $current \leftarrow next$
else $current \leftarrow next$ **only with probability** $e^{\Delta E/T}$

Stochastic Hill Climbing

1. Select a random state
2. Evaluate the fitness scores for all the successors of the state
3. Calculate the probability of selecting a successor based on fitness score
4. Select the next state based on the highest probability
5. Repeat from Step 2



12 N = {4,2,2,3,3,2,1,3,2,1,3,2}

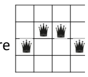
Simulated Annealing

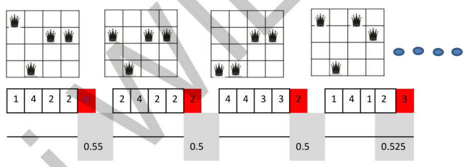
function SIMULATED-ANNEALING($problem$, $schedule$) **returns** a solution state
  inputs: $problem$, a problem
    $schedule$, a mapping from time to "temperature"

  $current \leftarrow$ MAKE-NODE($problem$.INITIAL-STATE)
  for $t = 1$ **to** ∞ **do**
    $T \leftarrow schedule(t)$
    if $T = 0$ **then return** $current$
    $next \leftarrow$ a randomly selected successor of $current$
    $\Delta E \leftarrow next$.VALUE − $current$.VALUE
    if $\Delta E > 0$ **then** $current \leftarrow next$
    else $current \leftarrow next$ only with probability $e^{\Delta E/T}$

Simulated Annealing

1. Select a random state
2. Evaluate the fitness scores for all the successors of the state
3. Calculate the probability of selecting a successor based on fitness score
4. Select the next state based on the highest probability
5. Repeat from Step 2





12 N = {4,2,2,3,3,2,1,3,2,1,3,2}
Init = 2

Simulated Annealing

Current Value = 4 (Local Maxima)
Global Maxima = 6

Next Value	ΔE	ΔE/t	$e^{\Delta E/t}$	$\frac{1}{1+e^{\Delta E/t}}$	ΔE/t	$e^{\Delta E/t}$	$\frac{1}{1+e^{\Delta E/t}}$
2	2	0.1	1.12	0.47	0.4	1.49	0.40
3	1	0.05	1.05	0.49	0.2	1.22	0.45
5	-1	-0.05	0.95	0.51	-0.2	0.82	0.55

Simulated Annealing

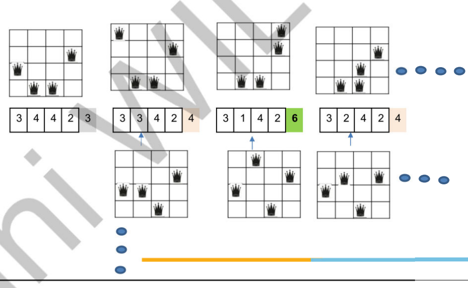
Maximization problem design to achieve global minima

Set Temp to very high temp t
Set n as number of iteration to be performed at a particular t
L1: Randomly select a random neighbour
Calculate Energy barrier E = f(N)-f(C)
If E > 0 then its a good move
  Move ahead for next tree search level
Else
  Create a random number r:[0-1]
  If r< $e^{-E/t}$
    Choose this bad state & move downhill
  Else
    Go to L1.
If Goal is reached or {acceptable goal(set criteria to check)node is reached & t is small END}
Else
  If no.of.neighbors explored has reached a threshold >=n
    then Lower t and go to L1.

Local Beam Search

Beam Search

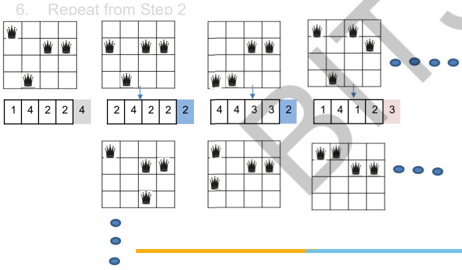
1. Initialize k random state
2. Evaluate the fitness scores for all the successors of the k states
3. Calculate the probability of selecting a successor based on fitness score
4. Select the next state based on the highest probability
5. If the goal is not found, Select the next 'k' states randomly based on the probabilit
6. Repeat from Step 2



Stochastic Beam Search

Sample from 1st State

1. Initialize k random state
2. Evaluate the fitness scores for all the successors of the k states
3. Calculate the probability of selecting a successor based on fitness score
4. Select the next state based on the highest probability
5. If the goal is not found, Select the next 'k' states randomly based on the probability
6. Repeat from Step 2



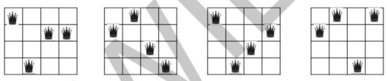
Genetic Algorithm

Genetic Algorithm

```
function GENETIC-ALGORITHM(population, FITNESS-FN) returns an individual
  inputs: population, a set of individuals
          FITNESS-FN, a function that measures the fitness of an individual

  repeat
     new_population ← empty set
     for i = 1 to SIZE(population) do
        x ← RANDOM-SELECTION(population, FITNESS-FN)
        y ← RANDOM-SELECTION(population, FITNESS-FN)
        child ← REPRODUCE(x, y)
        if (small random probability) then child ← MUTATE(child)
        add child to new_population
     population ← new_population
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to FITNESS-FN

function REPRODUCE(x, y) returns an individual
  inputs: x, y, parent individuals

  n ← LENGTH(x); c ← random number from 1 to n
  return APPEND(SUBSTRING(x, 1, c), SUBSTRING(y, c + 1, n))
```


Genetic Algorithm

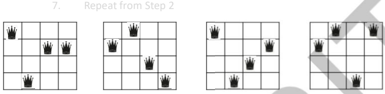
1. Select 'k' random states – **Initialization : k=4**
2. Evaluate the fitness value all states
3. If anyone of the state's has achieved the threshold fitness value or threshold new states or no change is seen than previous iteration then the algorithm stops
4. Else, use roulette wheel mechanism to select pair/s
5. Pairs selected produces new state (successor) by crossover
6. Successor is allowed to mutate
7. Repeat from Step 2



1 4 2 2	2 1 3 4	1 4 3 2	2 1 4 1

Genetic Algorithm

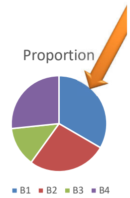
1. Select 'k' random states – **Initialization : k=4**
2. Evaluate the fitness value all states : Maximizing function : No.of.Non-attacking pairs Queens → Threshold = 6
3. If anyone of the state's has achieved the threshold fitness value or threshold new states or no change is seen than previous iteration then the algorithm stops
4. Else, use roulette wheel mechanism to select pair/s
5. Pairs selected produces new state (successor) by crossover
6. Successor is allowed to mutate
7. Repeat from Step 2

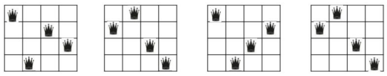


1 4 2 2 | 4	2 1 3 4 | 4	1 4 3 2 | 2	2 1 4 1 | 3

Genetic Algorithm – Example 1

Eg., use roulette wheel mechanism to select pair/s



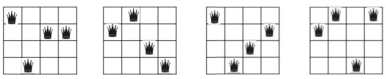


1 4 2 3 | 5	2 1 3 4 | 4	1 4 3 2 | 2	2 1 3 4 | 4
0.33	0.27	0.13	0.27
2 1 3 4	1 4 2 3	1 4 2 3	1 4 3 2

Sample winners of game -1 ,2,3,4 : B4, B1, B1, B3

Genetic Algorithm –Example 2
Selection

1. Select 'k' random states – **Initialization : k=4**
2. Evaluate the fitness value all states : Maximizing function : No.of.Non-attacking pairs Queens → Threshold = 6
3. ~~If anyone of the state's has achieved the threshold fitness value or threshold new states or no change is seen than previous iteration then the algorithm stops~~
4. Else, use roulette wheel mechanism to select pair/s
5. Pairs selected produces new state (successor) by crossover
6. Successor is allowed to mutate
7. Repeat from Step 2

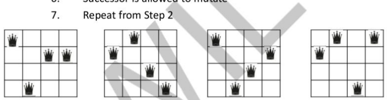


1 4 2 2 | 4	2 1 3 4 | 4	1 4 3 2 | 2	2 1 4 1 | 3
0.31	0.31	0.15	0.23
2 1 4 1	1 4 2 2	1 4 2 2	1 4 3 2

Sample winners of game -1 ,2,3,4 : B4, B1, B1, B3

Genetic Algorithm - Example 2
Crossover

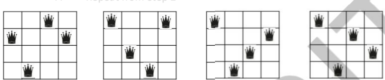
1. Select 'k' random states – **Initialization : k=4**
2. Evaluate the fitness value all states : Maximizing function : No.of.Non-attacking pairs Queens → Threshold = 6
3. If anyone of the state's has achieved the threshold fitness value or threshold new states or no change is seen than previous iteration then the algorithm stops
4. Else, use roulette wheel mechanism to select pair/s
5. Pairs selected produces new state (successor) by crossover
6. Successor is allowed to mutate
7. Repeat from Step 2



1 4 2 2 | 4	2 1 3 4 | 4	1 4 3 2 | 2	2 1 4 1 | 3
0.31	0.31	0.15	0.23
2 | 1 4 1	1 | 4 2 2	1 | 4 2 | 2	1 | 4 3 | 2
2 4 2 2	1 1 4 1	1 4 3 2	1 4 2 2

Genetic Algorithm - Example 2
Mutation

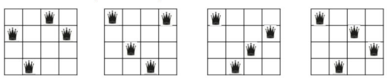
1. Select 'k' random states – **Initialization : k=4**
2. Evaluate the fitness value all states : Maximizing function : No.of.Non-attacking pairs Queens → Threshold = 6
3. ~~If anyone of the state's has achieved the threshold fitness value or threshold new states or no change is seen than previous iteration then the algorithm stops~~
4. Else, use roulette wheel mechanism to select pair/s
5. Pairs selected produces new state (successor) by crossover
6. Successor is allowed to mutate
7. Repeat from Step 2



2 4 2 2	1 1 4 1	1 4 3 2	1 4 2 2
2 4 1 2	1 3 4 1	1 4 3 2	1 4 2 3

Genetic Algorithm

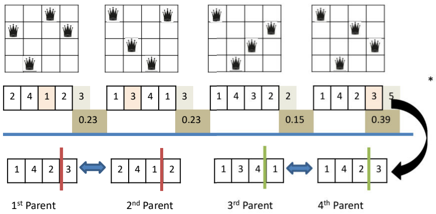
1. Select 'k' random states – **Initialization : k=4**
2. Evaluate the fitness value all states : Maximizing function : No.of.Non-attacking pairs Queens → Threshold = 6
3. If anyone of the state's has achieved the threshold fitness value or threshold new states or no change is seen than previous iteration then the algorithm stops
4. Else, use roulette wheel mechanism to select pair/s
5. Pairs selected produces new state (successor) by crossover
6. Successor is allowed to mutate
7. Repeat from Step 2



2 4 2 2	1 1 4 1	1 4 3 2	1 4 2 2
2 4 1 2 | 3	1 3 4 1 | 3	1 4 3 2 | 2	1 4 2 3 | 5
0.23	0.23	0.15	0.39

Genetic Algorithm

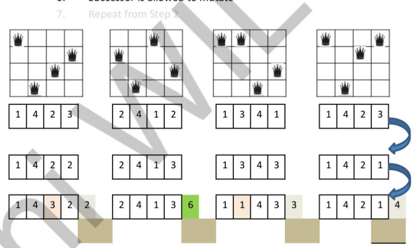
1. Select 'k' random states – **Initialization : k=4**
2. Evaluate the fitness value all states : Maximizing function : No.of.Non-attacking pairs Queens → Threshold = 6
3. If anyone of the state's has achieved the threshold fitness value or threshold new states or no change is seen than previous iteration then the algorithm stops
4. Else, use roulette wheel mechanism to select pair/s
5. Pairs selected produces new state (successor) by crossover
6. Successor is allowed to mutate
7. Repeat from Step 2



Other crossover and mutation operators are shared in separate document & uploaded in the elearn portal

Genetic Algorithm

1. Select 'k' random states – **Initialization : k=4**
2. Evaluate the fitness value all states : Maximizing function : No.of.Non-attacking pairs Queens → Threshold = 6
3. If anyone of the state's has achieved the threshold fitness value or threshold new states or no change is seen than previous iteration then the algorithm stops
4. Else, use roulette wheel mechanism to select pair/s
5. Pairs selected produces new state (successor) by crossover
6. Successor is allowed to mutate
7. Repeat from Step 2



Genetic Algorithm

Techniques:
1. Design of the fitness function
2. Diversity in the population to be accounted
3. Randomization

Application:
- Creative tasks
- Exploratory in nature
- Planning problem
- Static Applications

Required Reading: AIMA - Chapter # 4.1, #4.2

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

M1	Introduction to AI
M2	Problem Solving Agent using Search
M3	Game Playing
M4	Knowledge Representation using Logics
M5	Probabilistic Representation and Reasoning
M6	Reasoning over time
M7	Ethics in AI

Local Search

Optimization Problem

Goal : Navigate through a state space for a given problem such that an optimal solution can be found

Objective : Minimize or Maximize the objective evaluation function value

Scope : Local

Objective Function : Fitness Value evaluates the goodness of current solution

Local Search : Search in the state-space in the neighbourhood of current position until an optimal solution is found

Single Instance Based	Multiple Instance Based
Hill Climbing	Genetic Algorithm
Simulated Annealing	Particle Swarm Optimization
Local Beam Search	Ant Colony Optimization
Tabu Search	

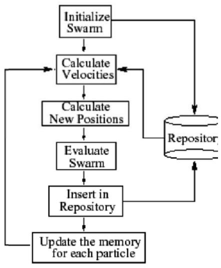
Particle Swarm Optimization

Particle Swarm Optimization

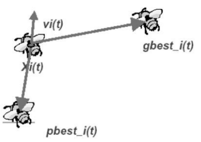
Basic Flow of PSO

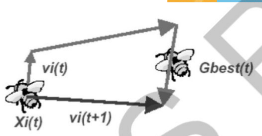
1. Initialize the swarm with random initializations
2. Evaluate fitness value for each of these individuals
3. Modify **gbest**, **pbest**, and velocity
4. Move each particle to new particle
5. Goto step 2, and repeat until convergence

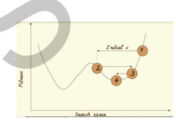
Particles velocities on each dimension are clamped to a max velocity v_{max}



Particle Swarm Optimization







Updating velocity vector:

$$v_i(t+1) = \alpha\ v_i + c_1 \times rand \times (pbest(t)\text{-}x_i(t)) + c_2 \times rand \times (gbest(t)\text{-}x_i(t))$$

$$X_i(t+1) = X_i(t) + V_i(t+1)$$

α is inertia weight and controls exploration and exploitation
c_1 and c_2 the cognition and social components respectively
rand is a random number generator

Ant Colony Optimization

ACO Pseudocode and notations

General pseudo-code
Procedure *ACO*
Schedule Activities
Initialization
Construction
Update Pheromone
Daemon Actions {optional}
// local search, elitism
End schedule activities
End *ACO*

Parameters used in ACO

Parameter	Description
N	Total No of ants ; N>1
τ_o	Initial pheromone amount
τ_{ij}	Amount of pheromone deposited while traversing from i to j
η_{ij}	Cost of link (i,j)
α	Importance coefficient of pheromone intensity
β	Importance coefficient of route cost
ρ	evaporation co-efficient; $0<\rho<1$
$visit_k$	Visited nodes table of k^{th} ant
Q	Importance - Constant value pertaining to pheromone trail
f_k	Route cost obtained by ant k

ACO Steps

Initialization
Place predefined number of ants on starting point
Set values for parameters α, β, ρ.
Set τ_0 to 0.
Construction
Compute the next node transition probability

$$NTP_{ij} = \frac{(\tau_{ij})^{\alpha}\,(\eta_{ij})^{\beta}}{\sum_{h \notin visit_k} (\tau_{ih})^{\alpha} (\eta_{ih})^{\beta}}$$

ACO Steps

Pheromone updation:
Pheromone reinforcement & pheromone evaporation
Direct impact on the exploitation (enhancing found food path) & exploration (discovering new path) of ant algorithms

$$\tau_{ij}^{new} = (\rho)\tau_{ij}^{old} + \Delta\tau_{ij}^{k}$$

Amount of pheromone deposited on (i,j) by kth at that timestamp is given by

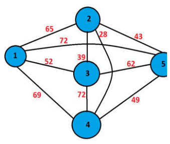
$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{\boldsymbol{f_k}} & if\ k^{th} ant\ passes\ i\ and\ j \\ 0 & otherwise \end{cases}$$

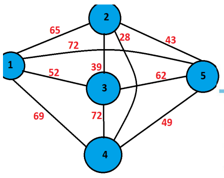
Stopping criteria: reaching predetermined number of iterations
Problem: Reaching pre-determined number of iterations before reaching destination leading to ant drop

Travelling Salesman Problem

Problem: Given n cities, the goal is to find shortest path going through all cities and visiting each exactly once
- Consider a complete graph
- d_{ij} is the route cost over (i,j) $\{f_k\}$

- Each ant builds its own tour from starting city
- Each ant chooses a town to go to with a probability
- Keep tabs on visit list of each ant
- When tour completed, lay pheromone on each edge visited
- Next city *j* after city *i* chosen according to probability rule



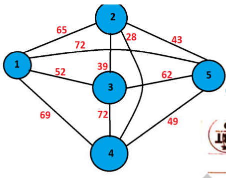


Initially No. of ants = No. of cities.
start at 4. $\alpha = 0.5$ $\beta = 0.75$ (0–1)
$Q = 100$ $\rho = 0.1$

$T = 0$

$\tau_{12} = \tau_{21} = 0.54$
$\tau_{13} = \tau_{31} = 0.53$
$\tau_{14} = \tau_{41} = 0.35$
$\tau_{15} = \tau_{51} = 0.24$
$\tau_{23} = \tau_{32} = 0.53$
$\tau_{24} = \tau_{42} = 0.39$
$\tau_{25} = \tau_{52} = 0.18$
$\tau_{34} = \tau_{43} = 0.32$
$\tau_{35} = \tau_{53} = 0.90$
$\tau_{45} = \tau_{54} = 0.68$

	1	2	3	4	5
1	0	65	52	69	72
2	65	0	39	28	43
3	52	39	0	72	62
4	69	28	72	0	49
5	72	43	62	49	0



Initially No. of ants = No. of cities.
start at 4. $\alpha = 0.5$ $\beta = 0.75$ (0–1)
$Q = 100$ $\rho = 0.1$

$T = 0$

$\tau_{12} = \tau_{21} = 0.54$
$\tau_{13} = \tau_{31} = 0.53$
$\tau_{14} = \tau_{41} = 0.35$
$\tau_{15} = \tau_{51} = 0.24$
$\tau_{23} = \tau_{32} = 0.53$
$\tau_{24} = \tau_{42} = 0.39$
$\tau_{25} = \tau_{52} = 0.18$
$\tau_{34} = \tau_{43} = 0.32$
$\tau_{35} = \tau_{53} = 0.90$
$\tau_{45} = \tau_{54} = 0.68$

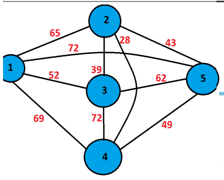
Next Transition Probability.

$$P_{41} = \frac{(\tau_{41})^{\alpha}(\eta_{41})^{\beta}}{(\tau_{41})^{\alpha}(\eta_{41})^{\beta} + (\tau_{42})^{\alpha}(\eta_{42})^{\beta} + (\eta_{43})^{\beta}(\tau_{43})^{\alpha} + (\eta_{45})^{\beta}(\tau_{45})^{\alpha}}$$

$1/d_{ij}$
$\eta_{41} = 0.014$
$\eta_{45} = 0.02$
$\eta_{43} = 0.014$
$\eta_{42} = 0.036$

$$= \frac{(0.35)^{0.5}(0.014)^{0.75}}{(0.35)^{0.5}(0.014)^{0.75} + (0.379)^{0.5}(0.036)^{0.75} + (0.326)^{0.5}(0.014)^{0.75} + (0.683)^{0.5}(0.02)^{0.75}}$$

$$= \frac{0.024}{0.143} = 0.168.$$



Initially No. of ants = No. of cities.
start at 4. $\alpha = 0.5$ $\beta = 0.75$ (0–1)
$Q = 100$ $\rho = 0.1$

$T = 0$

$\tau_{12} = \tau_{21} = 0.54$
$\tau_{13} = \tau_{31} = 0.53$
$\tau_{14} = \tau_{41} = 0.35$
$\tau_{15} = \tau_{51} = 0.24$
$\tau_{23} = \tau_{32} = 0.53$
$\tau_{24} = \tau_{42} = 0.39$
$\tau_{25} = \tau_{52} = 0.18$
$\tau_{34} = \tau_{43} = 0.32$
$\tau_{35} = \tau_{53} = 0.90$
$\tau_{45} = \tau_{54} = 0.68$

$= \frac{0.024}{0.143} = 0.168.$

$P_{42} = \frac{0.053}{0.142} = 0.364$ max

$P_{43} = \frac{0.023}{0.143} = 0.160.$

$P_{45} = 0.308$

Since P_{42} is max, move from 4 to 2.

Initially No. of ants = No. of cities.
start at 4. $\alpha = 0.5$ $\beta = 0.75$ (0–1)
$Q = 100$ $\rho = 0.1$

Pheromone Updation ($t = 1$)

$\tau_{12} = \rho(\tau_{12}) + \Delta\tau_{12}$ excepting for (4,2) & (2,4) $\Delta\tau_{ij}$ is zero for all other (i,j)
$= 0.054.$
$\tau_{13} = 0.053$, $\tau_{14} = 0.035$,
$\tau_{15} = 0.024$, $\tau_{23} = 0.05$,
$\tau_{24} = 0.039 + Q/d_{ij} = 0.039 + 100/28 = 3.610$
$\tau_{25} = 0.018$, $\tau_{34} = 0.032$, $\tau_{35} = 0.09$
$\tau_{45} = 0.068.$

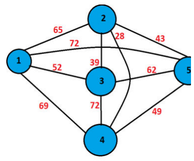
Now ant is at 2.

$P_{21} = 0.320$
$P_{23} = 0.429$ Max.
$P_{25} = 0.250$

Now ant moves to 3!

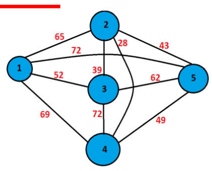
$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{f_k} & \text{if } k^{th} \text{ ant passes i and j} \\ 0 & \text{otherwise} \end{cases}$$

$$\tau_{ij}^{new} = (\rho)\tau_{ij}^{old} + \Delta\tau_{ij}^{k}$$



Pheromone Updation ($t = 2$).

$\tau_{12} = 0.005$, $\tau_{13} = 0.005$, $\tau_{14} = 0.003$,
$\tau_{15} = 0.002$, $\tau_{23} = 0.005 + 100/39 = 2.569$
$\tau_{24} = \rho(3.616) = 0.361$, $\tau_{25} = 0.001$
$\tau_{34} = 0.003$, $\tau_{35} = 0.009$, $\tau_{45} = 0.066$



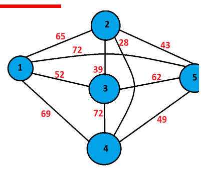
From 3, find P_{31}, P_{35}
$P_{31} = 0.500$
$P_{35} = 0.5571$ – Max.

From 3, ant moves to 5.

Tabu list $\boxed{4, 2, 3, 5}$

Pheromone Updation $t = 3$.

$\tau_{12} = 0.0005$, $\tau_{13} = 0.0005$, $\tau_{14} = 0.0003$,
$\tau_{15} = 0.0002$, $\tau_{23} = 0.25$, $\tau_{24} = 0.03$,
$\tau_{25} = 0.0001$, $\tau_{34} = 0.0003$, $\tau_{35} = 0.0009 + 100/62 = 1.613$
$\tau_{45} = 0.0006$



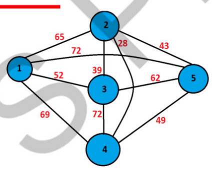
From 5, move to 1 since it is the only non-visited city. Update pheromone.

$\tau_{12} = 0.00005$, $\tau_{13} = 0.00005$, $\tau_{14} = 0.00003$,
~~$\tau_{13} = 0.00002$~~, ~~$\tau_{23} = 0$~~ $\tau_{15} = 1.388$, $\tau_{23} = 0.025$
$\tau_{24} = 0.003$, $\tau_{25} = 0.00001$, $\tau_{34} = 0.00003$
$\tau_{35} = 0.16$, $\tau_{45} = 0.00006.$

Now back to origin, since all the states are visited.

update $\tau_{14} = \tau_{41} = 0.00003 \times 100/69$
$= 1.449$

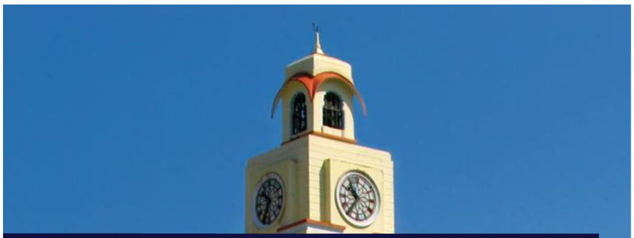
Final route,
$\boxed{4-2-3-5-1-4}$



Required Reading: AIMA - Chapter # 4.1, #4.2, #5.1

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials



Artificial & Computational Intelligence

AIMLCZG557

Contributors & Designers of document content : Cluster Course Faculty Team

M3 : Game Playing

BITS Pilani
Pilani Campus

Presented by
Faculty Name
BITS Email ID

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

Disclaimer

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

M1	Introduction to AI
M2	Problem Solving Agent using Search
M3	Game Playing
M4	Knowledge Representation using Logics
M5	Probabilistic Representation and Reasoning
M6	Reasoning over time
M7	Ethics in AI

Module 3 : Searching to play games

A. Min-max Algorithm

A. Alpha-Beta Pruning

C. Making imperfect real time decisions

Learning Objective

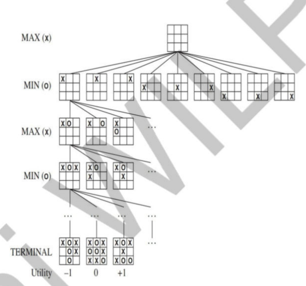
At the end of this class , students Should be able to:

1. Convert a given problem into adversarial search problem
2. Formulate the problem solving agent components
3. Design static evaluation function value for a problem
4. Construct a Game tree
5. Apply Min-Max
6. Apply and list nodes pruned by alpha pruning and nodes pruned by beta pruning

---

Task Environment

Phases of Solution Search by PSA

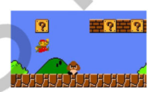


Assumptions – Environment :
Static (4.5)
Observable
Discrete (4.4)
Deterministic
Number of Agents

Game Problem

Study & design of games enables the computers to model ways in which humans think & act hence simulating human intelligence.

AI for Gaming:
- Interesting & Challenging Problem
- Larger Search Space Vs Smaller Solutions
- Explore to better the Human Computer Interaction



Characteristics of Games:
- Observability
- Stochasticity
- Time granularity
- Number of players

Adversarial Games:
Goals of agents are in conflict where one's optimized step would reduce the utility value of the other.

Single Player Game

as Constraint Satisfaction Problem
An Overview - Sudoku

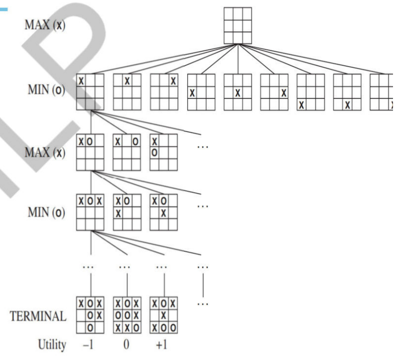
Problem Formulation

- Total variables = 81
  - One for each square
  - X = {A1, A2, A3, ..., A9, B1, B2, ...}
- Domains
  - Empty Squares has {1, 2, 3, 4, 5, 6, 7, 8, 9}
  - Filled Squares has only one value as provided
- Constraints
  - A1 – A9 should all be distinct, ...
  - A1 – I1 should all be distinct, ...
  - A1-3, B1-3, C1-3 should all be distinct

	1	2	3	4	5	6	7	8	9
A			3		2		6		
B	9			3		5			1
C			1	8		6	4		
D			8	1		2	9		
E	7								8
F			6	7		8	2		
G			2	6		9	5		
H	8			2		3			9
I			5		1		3		

Games as Search Problem

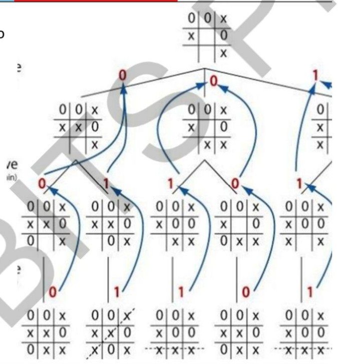
PSA : Representation of Game:
INITIAL STATE: S0
PLAYER(s)
ACTIONS(s)
RESULT(s, a)
TERMINAL-TEST(s)
UTILITY(s, p)
Eg., Tic Tac Toe

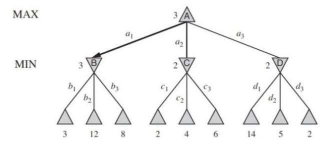


Min-Max Algorithm

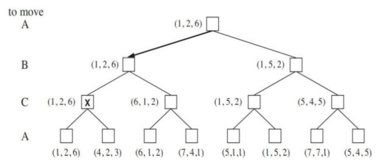
Idea: Uses Depth – First search exploration to decide the move

Let
start Player = MAX
Depth m =3





Two Player Game : - 2 Ply Game



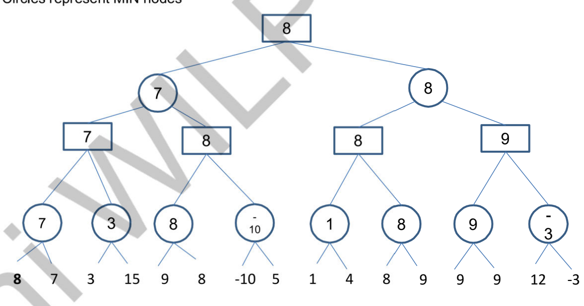
Multiplayer Game

Min-Max Algorithm

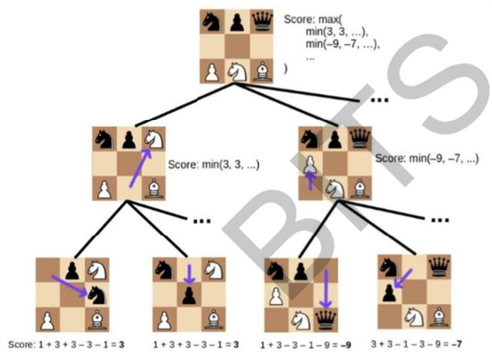
```
function MINIMAX-DECISION(state) returns an action
  return arg max_{a ∈ ACTIONS(s)} MIN-VALUE(RESULT(state, a))

function MAX-VALUE(state) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  v ← −∞
  for each a in ACTIONS(state) do
    v ← MAX(v, MIN-VALUE(RESULT(s, a)))
  return v

function MIN-VALUE(state) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  v ← ∞
  for each a in ACTIONS(state) do
    v ← MIN(v, MAX-VALUE(RESULT(s, a)))
  return v
```


Min-Max Algorithm – Example -1

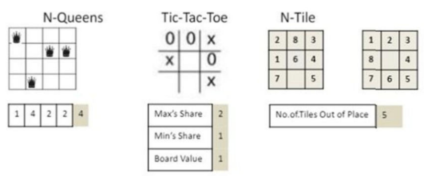
Squares represent MAX nodes
Circles represent MIN nodes



Design of Static Evaluation Values



Design of Static Evaluation Values



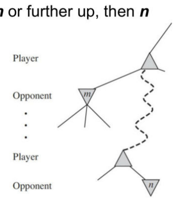
$Eval(S) = w_1 f_1(s) + w_2 f_2(s) + ... + w_n f_n(s)$

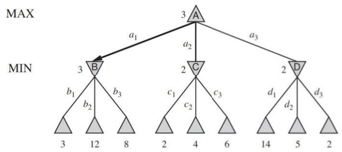
$= 0.6$ (MaxChance - MinChance) $+ 0.4$ (MaxPairs - MinPairs)

Alpha – beta Pruning

General Principle:

At a node ***n*** if a player has better option at the parent of ***n*** or further up, then ***n*** node will never be reached .Hence the entire subtree pruned

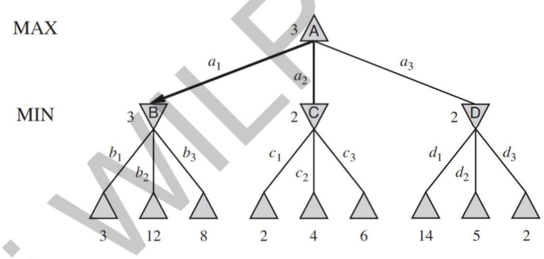




$$
\begin{aligned}
\text{MINIMAX}(root) &= \max(\min(3,12,8), \min(2,x,y), \min(14,5,2)) \\
&= \max(3, \min(2,x,y), 2) \\
&= \max(3, z, 2) \quad \text{where } z = \min(2,x,y) \le 2 \\
&= 3.
\end{aligned}
$$

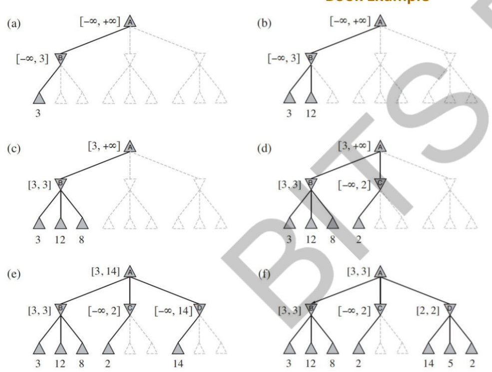
Alpha Beta Pruning

Book Example



Alpha Beta Pruning

Book Example



Min-Max Algorithm

Alpha beta Modifications

function ALPHA-BETA-SEARCH(*state*) **returns** an action
  $v \leftarrow$ MAX-VALUE(*state*, $-\infty, +\infty$)
  return the *action* in ACTIONS(*state*) with value v

function MAX-VALUE(*state*, α, β) **returns** *a utility value*
  if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)
  $v \leftarrow -\infty$
  for each a **in** ACTIONS(*state*) **do**
    $v \leftarrow$ MAX(v, MIN-VALUE(RESULT(s,a), α, β))
    if $v \ge \beta$ **then return** v
    $\alpha \leftarrow$ MAX(α, v)
  return v

function MIN-VALUE(*state*, α, β) **returns** *a utility value*
  if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)
  $v \leftarrow +\infty$
  for each a **in** ACTIONS(*state*) **do**
    $v \leftarrow$ MIN(v, MAX-VALUE(RESULT(s,a) , α, β))
    if $v \le \alpha$ **then return** v
    $\beta \leftarrow$ MIN(β, v)
  return v

Is it possible to compute the minimax decision for a node without looking at every successor node?

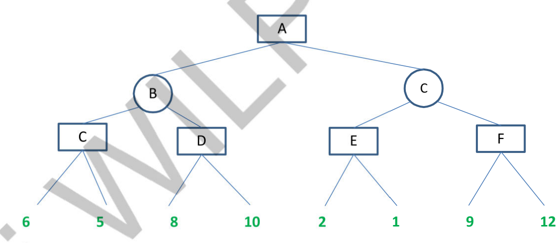
Alpha – beta Pruning

Steps in Alpha – Beta Pruning:

1. At root initialize alpha = -∞ and beta = + ∞. This is to set the worst case boundary to start the algorithm which aims to increase alpha and decrease beta as much as optimally possible
2. Navigate till the depth / limit specified and get the static evaluated numeric value.
3. For every value VAL being analyzed : Loop till all the leaf/terminal/specified state level nodes are
   analyzed & accounted for OR until **beta <= alpha.**
   1. If the player is MAX :
      1. If VAL > alpha
      2. then reset alpha = VAL
      3. also check **if** beta <=alpha **then** tag the path as unpromising (TO BE AVOIDED) **and** prune the branch from game tree. Rest of their siblings are not considered for analysis
   2. Else if the player is MIN:
      1. If VAL < beta
      2. then reset beta = VAL
      3. also check **if** beta <=alpha **then** tag the path as unpromising (TO BE AVOIDED) **and** prune the branch from game tree. Rest of their siblings are not considered for analysis

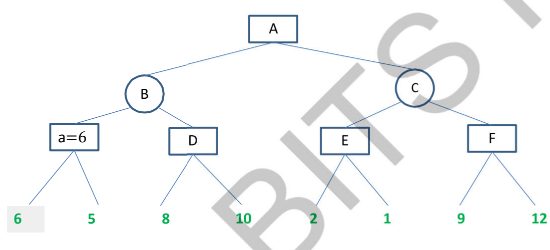
Alpha Beta Pruning - Another Example

Idea –Pruning



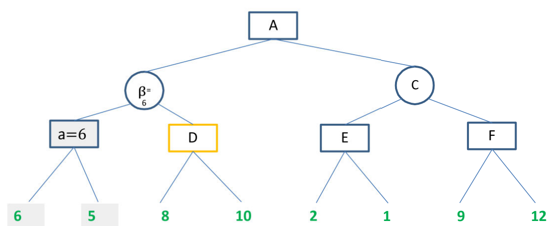
Alpha Beta Pruning

Idea –Pruning



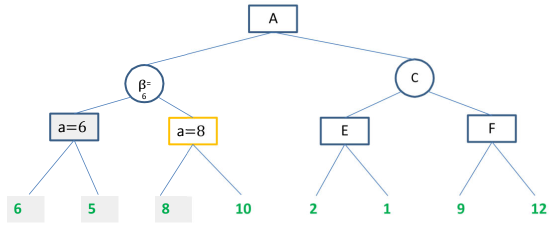
Alpha Beta Pruning

Idea –Pruning



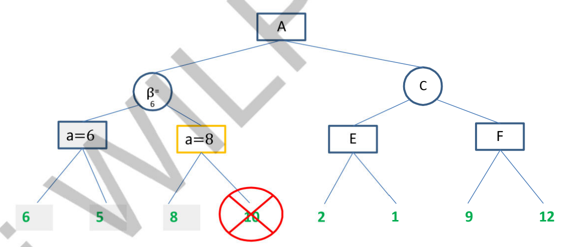
Alpha Beta Pruning

Idea – Alpha Pruning



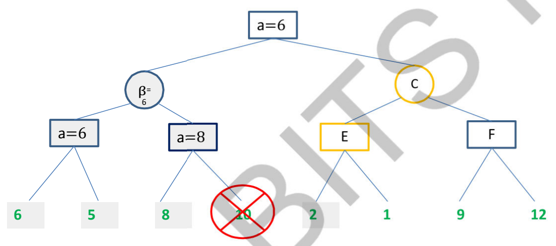
Alpha Beta Pruning

Idea – Beta Pruning



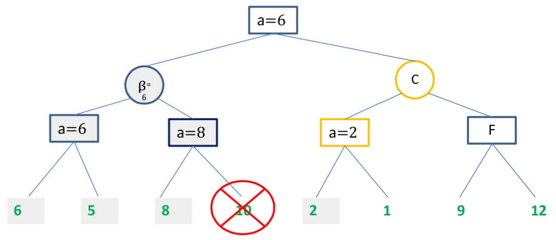
Alpha Beta Pruning

Idea –Pruning



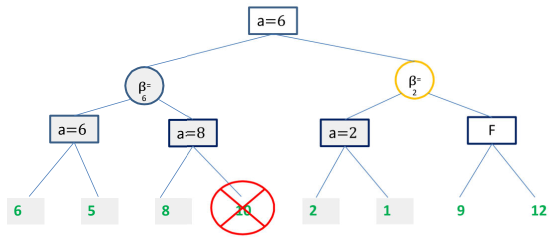
Alpha Beta Pruning

Idea –Pruning



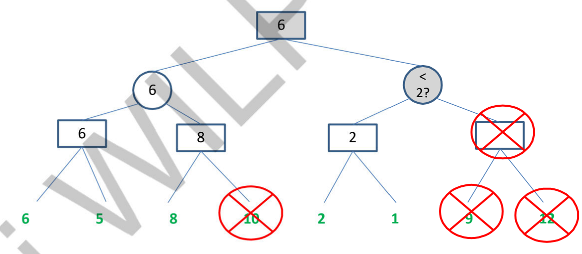
Alpha Beta Pruning

Idea –Pruning



Alpha Beta Pruning

Idea – Alpha Pruning

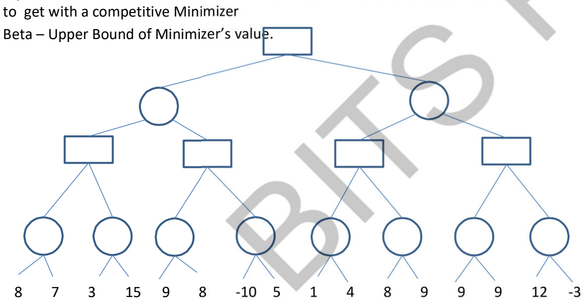


Alpha – Lower bound of Maximizer's value. Perceived value that Maximizer hopes to against a competitive Minimizer

Alpha – beta Pruning–
Example -4 Do for practice.

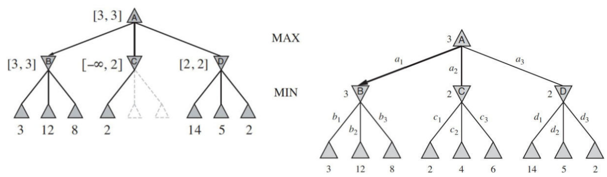
Alpha – Lower bound of Maximizer's value. Perceived value that Maximizer hopes to get with a competitive Minimizer

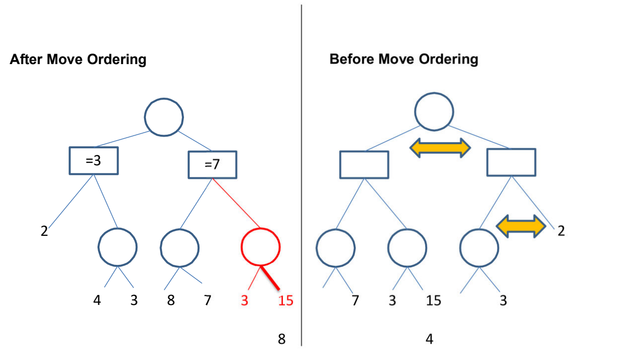
Beta – Upper Bound of Minimizer's value.



Computational Efficiency

How to reduce the move generations better along while doing Alpha-Beta Pruning?

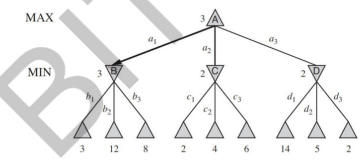




Gaming (Imperfect Decisions)

Computational Efficiency

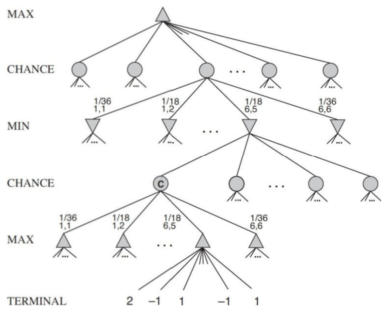
How games can be designed to handle imperfect decisions in real-time?



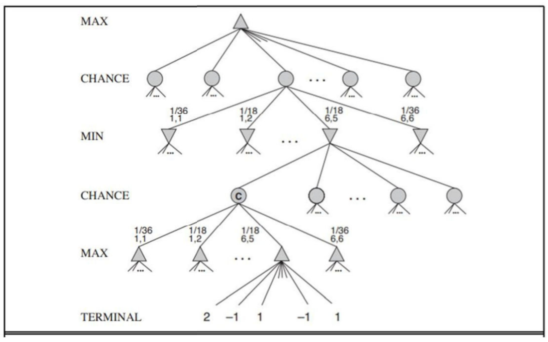
Computational Efficiency

Idea : Chance Node:

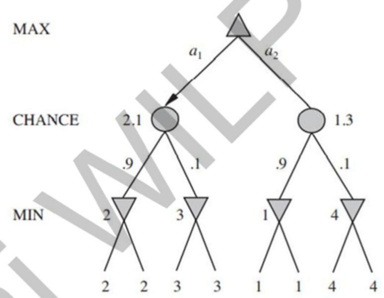
Holds the expected values that are computed as a sum of all outcomes weighted by their probability (of dice roll)

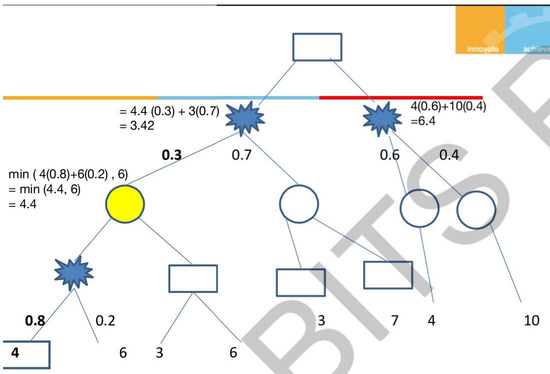


Expecti Mini Max Algorithm



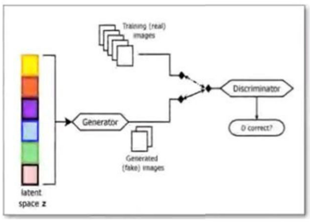
Expecti Min Max Algorithm





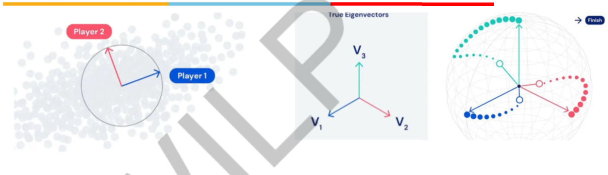
Game Playing (Interesting Case Studies)

Games in Image Processing



Source Credit:
2019 - Analyzing and Improving the Image Quality of StyleGAN
Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila
https://thispersondoesnotexist.com/

Games in Feature Engineering



Source Credit:
https://deepmind.com/blog/article/EigenGame
2021 - EigenGame: PCA as a Nash Equilibrium , Ian Gemp, Brian McWilliams, Claire Vernade, Thore Graepel

Games in Feature Engineering

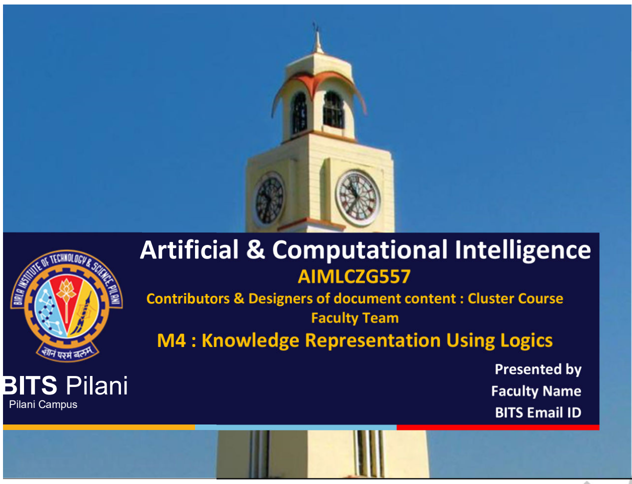
$$\mathrm{Utility}(v_i \mid v_{j<i}) = \boxed{\mathrm{Var}(v_i)} - \sum_{j<i} \boxed{\mathrm{Align}(v_i, v_j)}$$

Source Credit:
https://deepmind.com/blog/article/EigenGame
2021 - EigenGame: PCA as a Nash Equilibrium , Ian Gemp, Brian McWilliams, Claire Vernade, Thore Graepel

Required Reading: AIMA - Chapter # 4.1, #4.2, #5.1

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials



Artificial & Computational Intelligence

AIMLCZG557

Contributors & Designers of document content : Cluster Course Faculty Team

M4 : Knowledge Representation Using Logics

BITS Pilani
Pilani Campus

Presented by
Faculty Name
BITS Email ID

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

M1 Introduction to AI
M2 Problem Solving Agent using Search
M3 Game Playing
M4 Knowledge Representation using Logics
M5 Probabilistic Representation and Reasoning
M6 Reasoning over time
M7 Ethics in AI

Knowledge Representation Using Logics

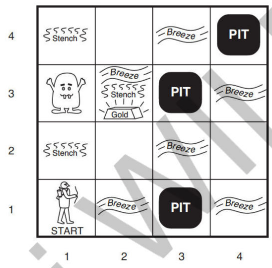
Learning Objective

At the end of this class , students Should be able to:

1. Represent a given knowledge base into logic formulation
2. Infer facts from KB using Resolution
3. Infer facts from KB using Forward Chaining
4. Infer facts from KB using Backward Chaining

Knowledge based Agent : Model & Represent

Concepts, logic Representation of a sample agent



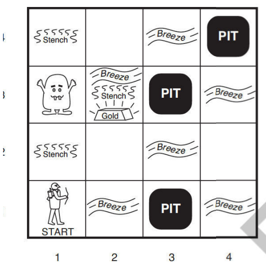
Wumpus World Problem:

PEAS:
Performance Measure:
+1000 for climbing out with gold,
-1000 for falling into a pit or being eaten by Wumpus,
-1 for each action taken and
-10 for using an arrow

Environment: 4x4 grid of rooms. Always starts at [1, 1] facing right.
The location of Wumpus and Gold are random.
Agent dies if entered a pit or live Wumpus.

Knowledge based Agent : Model & Represent

Concepts, logic Representation of a sample agent

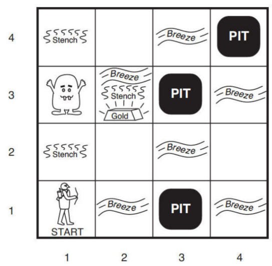


Wumpus World Problem:

PEAS:
Actuators –
Forward,
TurnLeft by 90,
TurnRight by 90,
Grab – pick gold if present,
Shoot – fire an arrow, it either hits a wall or kills wumpus. Agent has only one arrow.
Climb – Used to climb out of cave, only from [1, 1]

Knowledge based Agent : Model & Represent

Concepts, logic Representation of a sample agent



Why do we need Factored representation

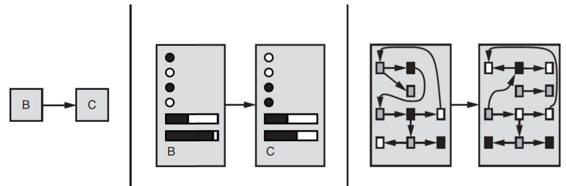
- To reason about steps
- To learn new knowledge about the environment
- To adapt to changes to the existing knowledge
- Accept new tasks in the form of explicit goals
- To overcome partial observability of environment

Representation

Agents based on Propositional logic, TT-Entail for inference from truth table

Syntax
Semantics
Model

→ Logic → Propositional Logic / Predicate Logic



(a) Atomic (b) Factored (b) Structured

Search Strategies Propositional Logic First Order Logic

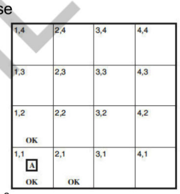
Propositional Logic

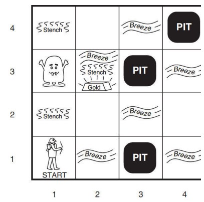
Agentsbasedon Propositionallogic,TT-Entailforinferencefromtruthtable

A simple representation language for building knowledge-based agents

Proposition Symbol - A symbol that stands for a proposition.
E.g., W1,3 - "Wumpus in [1,3]" is a proposition and $W_{1,3}$ is the symbol
Proposition can be true or false

Atomic : $W_{1,3}$
Conjuncts : $W_{1,3} \wedge P_{3,1}$
Disjuncts : $W_{1,3} \vee P_{3,1}$
Implications :
$(W_{1,3} \wedge P_{3,1}) \Rightarrow \neg W_{2,2}$
Biconditional : $W_{1,3} \Leftrightarrow \neg W_{2,2}$





Propositional Logic

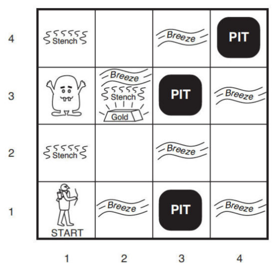
Agents based on Propositional logic, TT-Entail for inference from truth table

Tie break in search:
$\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$
$(\neg A) \wedge B$ has precedence over $\neg (A \wedge B)$

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Knowledge based Agent : Model & Represent

Concepts, logic Representation of a sample agent

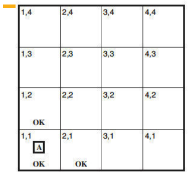


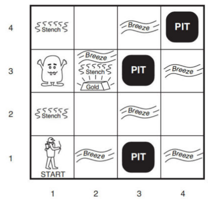
Wumpus World Problem:

PEAS:
Sensors. The agent has five sensors
Stench: In all adjacent (but not diagonal) squares of Wumpus
Breeze: In all adjacent (but not diagonal) squares of a pit
Glitter: In the square where gold is
Bump: If agent walks into a wall
Scream: When Wumpus is killed, it can be perceived everywhere

Percept Format:
[Stench?, Breeze?, Glitter?, Bump?, Scream?]
E.g., [Stench, Breeze, None, None, None]

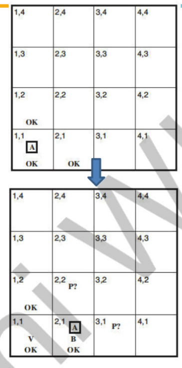
Percept 1: [None, None, None, None, None]
Action: Forward

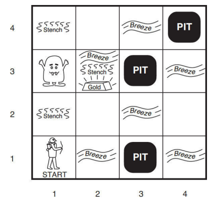




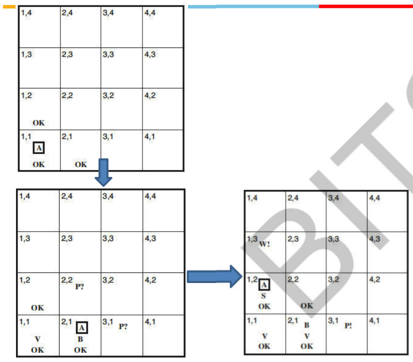
Percept Format:
[Stench?, Breeze?, Glitter?, Bump?, Scream?]

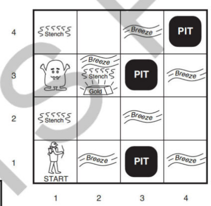
Percept 2: [None , Breeze, None, None, None, None]





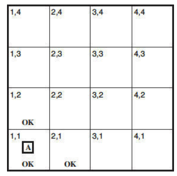
Percept 3: [Stench, None, None, None, None]
Action: Move to [2, 2]
Remembers (2,2) as possible PIT and no Stench.



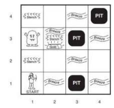


Representation by Propositional Logic

For each [x, y] location
$P_{x,y}$ is true if there is a pit in [x, y]
$W_{x,y}$ is true if there is a wumpus in [x, y]
$B_{x,y}$ is true if agent perceives a breeze in [x, y]
$S_{x,y}$ is true if agent perceives a stench in [x, y]



----------------------- R is the sentence in KB
$R_1 : \neg P_{1,1}$
$R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
$R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
$R_4 : \neg B_{1,1}$
$R_5 : B_{2,1}$



Query : $\neg P_{1,2}$ entailed by our KB?

TT – Entails Inference – Example

Agents based on Propositional logic, TT-Entail for inference from truth table

$\neg P_{1,2}$ entailed by our KB?

Way – 1 :

1. Get sufficient information $B_{1,1}$, $B_{2,1}$, $P_{1,1}$, $P_{1,2}$, $P_{2,1}$, $P_{2,2}$, $P_{3,1}$
2. Enumerate all models with combination of truth values to propositional symbols
3. In all the models, find those models where KB is true, i.e., every sentence R_1, R_2, R_3, R_4, R_5 are true
4. In those models where KB is true, find if query sentence $\neg P_{1,2}$ is true
5. If query sentence $\neg P_{1,2}$ is true in all models where KB is true, then it entails, otherwise it won't

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	false	false	false	false	false	false	true	true	true	true	false	false
false	false	false	false	false	false	true	true	true	false	true	false	false
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	true	false	false	false	false	false	true	true	false	true	true	false
false	true	false	false	false	false	true	true	true	true	true	true	<u>true</u>
false	true	false	false	false	true	false	true	true	true	true	true	<u>true</u>
false	true	false	false	false	true	true	true	true	true	true	true	<u>true</u>
false	true	false	false	true	false	false	true	false	false	true	true	false
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	true	true	true	true	true	true	false	true	true	false	true	false

TT – Entails Inference – Example

Agents based on Propositional logic, TT-Entail for inference from truth table

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	false	false	false	false	false	false	true	true	true	true	false	false
false	false	false	false	false	false	true	true	true	false	true	false	false
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	true	false	false	false	false	false	true	true	false	true	true	false
false	true	false	false	false	false	true	true	true	true	true	true	<u>true</u>
false	true	false	false	false	true	false	true	true	true	true	true	<u>true</u>
false	true	false	false	false	true	true	true	true	true	true	true	<u>true</u>
false	true	false	false	true	false	false	true	false	false	true	true	false
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	true	true	true	true	true	true	false	true	true	false	true	false

Inference : Properties

1. Entailment : $\alpha \models \beta$
2. Equivalence : $\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$
3. Validity
4. Satisfiability

Inference : Example – Theorem Proving (Self Study)

Propositional theorem proving - Proof by resolution

Logical Equivalence rules can be used as inference rules

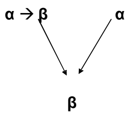
$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) \quad \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) \quad \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) \quad \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) \quad \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) \quad \text{De Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) \quad \text{De Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) \quad \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) \quad \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

Inference : Example – Theorem Proving

1. **Modes Ponens**
2. AND Elimination

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg \alpha) \equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg \beta \Rightarrow \neg \alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg \alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta) \equiv (\neg \alpha \vee \neg \beta)$	De Morgan
$\neg(\alpha \vee \beta) \equiv (\neg \alpha \wedge \neg \beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

α : I walk in rain without the umbrella
β : I get wet

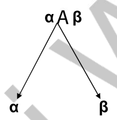


Inference : Example – Theorem Proving

1. Modes Ponens
2. **AND Elimination**

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg \alpha) \equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg \beta \Rightarrow \neg \alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg \alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta) \equiv (\neg \alpha \vee \neg \beta)$	De Morgan
$\neg(\alpha \vee \beta) \equiv (\neg \alpha \wedge \neg \beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

α : I walk in rain without the umbrella
β : I get wet



Inference : Example – Theorem Proving

$R_1 : \neg P_{1,1}$
$R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
$R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
$R_4 : \neg B_{1,1}$
$R_5 : B_{2,1}$

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg \alpha) \equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg \beta \Rightarrow \neg \alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg \alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta) \equiv (\neg \alpha \vee \neg \beta)$	De Morgan
$\neg(\alpha \vee \beta) \equiv (\neg \alpha \wedge \neg \beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

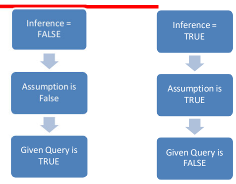
Query: $\neg P_{1,2}$. Can we prove if this sentence be entailed from KB using inference rules?______________

$R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$	
$R_6 : (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$	Biconditional Elimination
$R_7 : ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$	And Elimination
$R_8 : (\neg B_{1,1} \Rightarrow \neg (P_{1,2} \vee P_{2,1}))$	Contraposition
$R_9 : \neg (P_{1,2} \vee P_{2,1})$	Modus Ponens
$R_{10} : \neg P_{1,2} \wedge \neg P_{2,1}$	Demorgans
$R11: \neg P_{1,2}$	And Elimination

Propositional Logic

innovate achieve lead

Proof by Contradiction

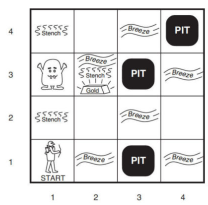


PL-Resolution

innovate achieve lead

Horn Clause

1. Definite Clause : A horn clause with exactly one positive literal
2. Fact : Definite clause with no negative literal / assertion
3. Rule
4. Inference by Chaining



PL-Resolution : CNF conversion

innovate achieve lead

Wumpus world Book example

$R_1 : \neg P_{1,1}$
$R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
$R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
$R_4 : \neg B_{1,1}$
$R_5 : B_{2,1}$
Query: $\neg P_{1,2}$

Conjunctive Normal Form :
(A V ~B) ^ (A V B V ~C) ^ ~A
Unit Resolution : ~A
Query : Is 'C' true?

PL-Resolution : CNF conversion

innovate achieve lead

Wumpus world Book example

$R_1 : \neg P_{1,1}$
$R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
$R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
$R_4 : \neg B_{1,1}$
$R_5 : B_{2,1}$
Query: $\neg P_{1,2}$

Conjunctive Normal Form :
(A V ~B) ^ (A V B V ~C) ^ ~A
Unit Resolution : ~A
Query : Is 'C' true?

PL-Resolution

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg \alpha) \equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg \beta \Rightarrow \neg \alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg \alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta) \equiv (\neg \alpha \vee \neg \beta)$	De Morgan
$\neg(\alpha \vee \beta) \equiv (\neg \alpha \wedge \neg \beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

$R_1 : \neg P_{1,1}$
$\cancel{R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})}$
$\cancel{R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})}$
$R_4 : \neg B_{1,1}$
$R_5 : B_{2,1}$
Query: $\neg P_{1,2}$

$R_6 : \neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$
$R_7 : \neg P_{1,2} \vee B_{1,1}$
$R_8 : \neg P_{2,1} \vee B_{1,1}$
$R_9 : \neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$
$R_{10} : \neg P_{1,1} \vee B_{2,1}$
$R_{11} : \neg P_{2,2} \vee B_{2,1}$
$R_{12} : \neg P_{3,1} \vee B_{2,1}$

Eliminate		$R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$	$R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
←→	Biconditional Elimination	$(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$	$(B_{2,1} \Rightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})) \wedge ((P_{1,1} \vee P_{2,2} \vee P_{3,1}) \Rightarrow B_{2,1})$
→	Implication Elimination	$\neg B_{1,1} \vee (P_{1,2} \vee P_{2,1})$ $\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1}$	$\neg B_{2,1} \vee (P_{1,1} \vee P_{2,2} \vee P_{3,1})$ $\neg(P_{1,1} \vee P_{2,2} \vee P_{3,1}) \vee B_{2,1}$
Clause level ¬	De Morgan	$(\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1}$	$(\neg P_{1,1} \wedge \neg P_{2,2} \wedge \neg P_{3,1}) \vee B_{2,1}$
CNF Form	Distributivity of V over Λ	$(\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$	$(\neg P_{1,1} \vee B_{2,1}) \wedge (\neg P_{2,2} \vee B_{2,1}) \wedge (\neg P_{3,1} \vee B_{2,1})$

PL-Resolution

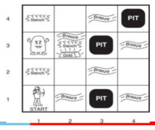
Unit Resolution: Query: $\neg P_{1,2}$

To find: Is there a pit in location (1,2) using the CNF obtained in previous slide

Inference = FALSE
↓
Assumption is False
↓
Given Query is TRUE

$R_1 : \neg P_{1,1}$
~~$R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$~~
~~$R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$~~
$R_4 : \neg B_{1,1}$
$R_5 : B_{2,1}$
Query: $\neg P_{1,2}$

$R_6 : \neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$
$R_7 : \neg P_{1,2} \vee B_{1,1}$
$R_8 : \neg P_{2,1} \vee B_{1,1}$
$R_9 : \neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$
$R_{10} : \neg P_{1,1} \vee B_{2,1}$
$R_{11} : \neg P_{2,2} \vee B_{2,1}$

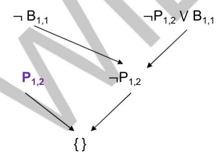


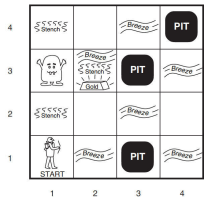
PL-Resolution

Unit Resolution: Query: $\neg P_{1,2}$

To find: Is there a pit in location (1,2) using the CNF obtained in previous slide

Inference = FALSE
↓
Assumption is False
↓
Given Query is TRUE





DPLL Algorithm

In logic and computer science, the Davis–Putnam–Logemann–Loveland (**DPLL**) **algorithm** is a complete, backtracking-based search **algorithm** for deciding the satisfiability of propositional logic formulae in conjunctive normal form

Improvements:

1. Early Termination
2. Pure Symbolic Heuristic
3. Unit Clause Heuristic

DPLL Algorithm

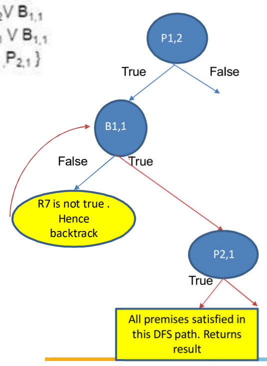
In logic and computer science, the Davis–Putnam–Logemann–Loveland (**DPLL**) **algorithm** is a complete, backtracking-based search **algorithm** for deciding the satisfiability of propositional logic formulae in conjunctive normal form

Improvements:

1. Early Termination
2. Pure Symbolic Heuristic
3. Unit Clause Heuristic

DPLL Algorithm

$R_7 : \neg P_{1,2} \vee B_{1,1}$
$R_8 : \neg P_{2,1} \vee B_{1,1}$
$\{P_{1,2}, B_{1,1}, P_{2,1}\}$



$R_7 : \neg P_{1,2} \vee B_{1,1}$
$R_8 : \neg P_{2,1} \vee B_{1,1}$

$R_7 : \neg P_{1,2} \vee B_{1,1}$
$R_8 : \neg P_{2,1} \vee B_{1,1}$

$R_7 : \neg P_{1,2} \vee B_{1,1}$
$R_8 : \neg P_{2,1} \vee B_{1,1}$

Towards Predicate Logic

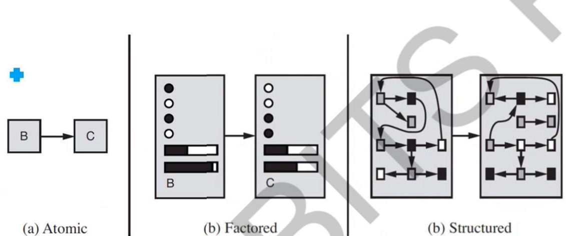
All courses are offered and interesting

All offered courses are interesting

Some of the courses are offered and interesting [Atleast one of the offered courses is interesting]

Some of the offered courses are interesting

Towards Predicate Logic



Predicate Logic

Squares neighboring the wumpus are smelly
Objects: squares, wumpus
Unary Relation (properties of an object): smelly N-ary
Relation (between objects): neighboring
Function: -

Primary difference between propositional and first-order logic lies in "ontological commitment" – the assumption about the nature of reality.

Predicate Logic – Sample Modelling

1. "Squares neighboring the wumpus are smelly"
$\forall x,y$ *Neighbour*$(x,y) \wedge$ *Wumpus*$(y) \Rightarrow$ Smelly(x)

Order of quantifiers is important

Predicate Logic – Sample Modelling

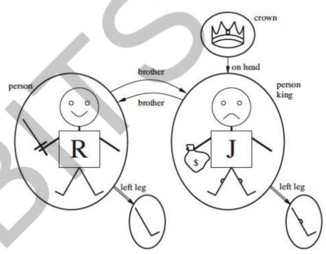
2. "Everybody loves somebody"
$\forall x\ \exists y\ Loves(x, y)$

3. "There is someone who is loved by everyone"
$\exists y\ \forall x\ Loves(x, y)$

Order of quantifiers is important

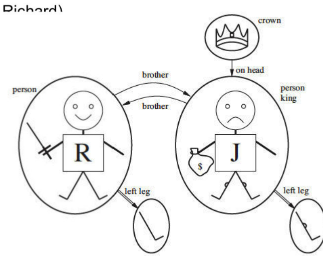
Predicate Logic – Sample Modelling

Brother(Richard, John) ∧ Brother(John, Richard)
King(Richard) ∨ King(John)
¬King(Richard) ⟹ King(John)



Predicate Logic – Sample Modelling
Quantifiers

Brother(Richard, John) ∧ Brother(John, Richard)
King(Richard) ∨ King(John)
¬King(Richard) ⟹ King(John)



"All Kings are persons"
$\forall x\ King(x) \Rightarrow Person(x)$

"King John has a crown on his head"
$\exists x\ Crown(x)\ \ AOnHead(x, John)$

Ground Term: A term with no variables. E.g., King(Richard)

Predicate Logic – Inference

1. Substitute for Quantifiers
2. Convert into Propositional Logic
3. Apply inference tech

$\forall x\ King(x) \Rightarrow Person(x)$
Richard the Lionheart is a king ⟹ Richard the Lionheart is a person
King John is a king ⟹ King John is a person

$\exists x\ Crown(x)\ \ AOnHead(x, John)$
Crown(C_1) A OnHead(C_1, John) ||C1 is imputed assumed fact

Forward Chaining

- Consider the following problem:

The law says it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

- We will prove that West is a criminal

Forward Chaining

- "All of its missiles were sold to it by Colonel West"
$$Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$$
- Missiles are weapons
$$Missile(x) \Rightarrow Weapon(x)$$
- Hostile means enemy
$$Enemy(x, America) \Rightarrow Hostile(x)$$
- "West, who is American"
$$American(West)$$
- "The country Nono, an enemy of America"
$$Enemy(Nono, America)$$

Forward Chaining

- First, we will represent the facts in First Order Definite Clauses

"... it is a crime for an American to sell weapons to hostile nations"
$$American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$$
"Nono ... has some missiles"
$$\exists x\ Owns(Nono, x) \wedge Missile(x)$$
is transformed into two definite clauses by Existential Instantiation
$$Owns(Nono, M_1)$$
$$Missile(M_1)$$

Forward Chaining

1. $American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$
2. $Missile(x) \wedge Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
3. $Missile(x) \Rightarrow Weapon(x)$
4. $Enemy(x,America) \Rightarrow Hostile(x)$

Missile(M1)
Owns(Nono, M1)
American (West)
Enemy (Nono, America)

Forward Chaining

- Consider the following problem:

The law says it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

- We will prove that West is a criminal

Algorithm:
1. Start from the facts
2. Trigger all rules whose premises are satisfied
3. Add the conclusion to known facts
4. Repeat the steps till no new facts are added or the query is answered

Forward Chaining

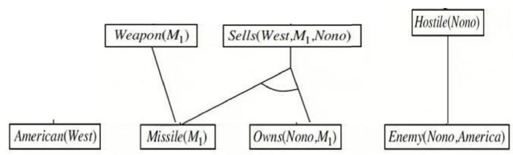
1. $American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$
2. $Missile(x) \wedge Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
3. $Missile(x) \Rightarrow Weapon(x)$
4. $Enemy(x,America) \Rightarrow Hostile(x)$

Missile(M1)
Owns(Nono, M1)
American (West)
Enemy (Nono, America)

Forward Chaining

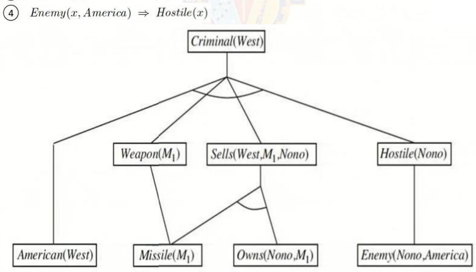
1. $American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$
2. $Missile(x) \wedge Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
3. $Missile(x) \Rightarrow Weapon(x)$
4. $Enemy(x,America) \Rightarrow Hostile(x)$

Missile(M1)
Owns(Nono, M1)
American (West)
Enemy (Nono, America



Forward Chaining

1. $American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$
2. $Missile(x) \wedge Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
3. $Missile(x) \Rightarrow Weapon(x)$
4. $Enemy(x,America) \Rightarrow Hostile(x)$



Forward Chaining

Algorithm:
1. Start from the facts - Conjunct Ordering
2. Trigger all rules whose premises are satisfied - Pattern Matching
3. Add the conclusion to known facts – **Irrelevant Facts**
4. Repeat the steps till no new facts are added or the query is answered – Redundant Rule Matching

Backward Chaining

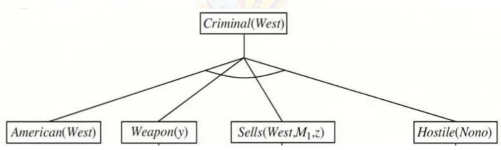
Algorithm:
1. Form Definite Clause
2. Start from the Goals
3. Search through rules to find the fact that support the proof
4. If it stops in the fact which is to be proved → Empty Set- LHS

Divide & Conquer Strategy
Substitution by Unification

Backward Chaining

1. $American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$
2. $Missile(x) \wedge Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
3. $Missile(x) \Rightarrow Weapon(x)$
4. $Enemy(x,America) \Rightarrow Hostile(x)$

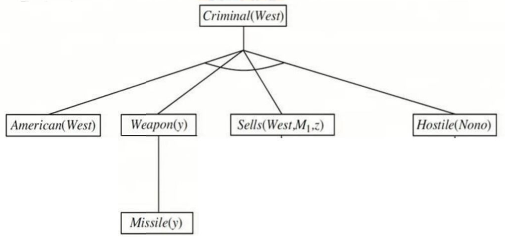
Missile(M1)
Owns(Nono, M1)
American (West)
Enemy (Nono, America



Backward Chaining

1. $American(x) \land Weapon(y) \land Sells(x,y,z) \land Hostile(z) \Rightarrow Criminal(x)$
2. $Missile(x) \land Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
3. $Missile(x) \Rightarrow Weapon(x)$
4. $Enemy(x, America) \Rightarrow Hostile(x)$

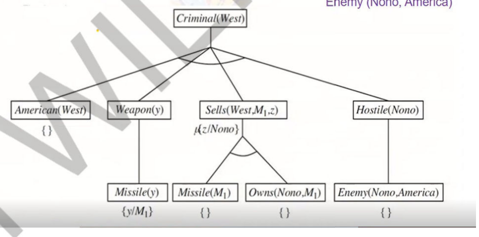
Missile(M1)
Owns(Nono, M1)
American (West)
Enemy (Nono, America)



Backward Chaining

1. $American(x) \land Weapon(y) \land Sells(x,y,z) \land Hostile(z) \Rightarrow Criminal(x)$
2. $Missile(x) \land Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
3. $Missile(x) \Rightarrow Weapon(x)$
4. $Enemy(x, America) \Rightarrow Hostile(x)$

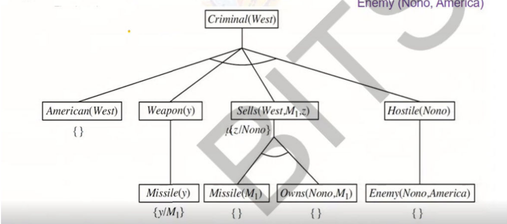
Missile(M1)
Owns(Nono, M1)
American (West)
Enemy (Nono, America)



Backward Chaining

1. $American(x) \land Weapon(y) \land Sells(x,y,z) \land Hostile(z) \Rightarrow Criminal(x)$
2. $Missile(x) \land Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
3. $Missile(x) \Rightarrow Weapon(x)$
4. $Enemy(x, America) \Rightarrow Hostile(x)$

Missile(M1)
Owns(Nono, M1)
American (West)
Enemy (Nono, America)



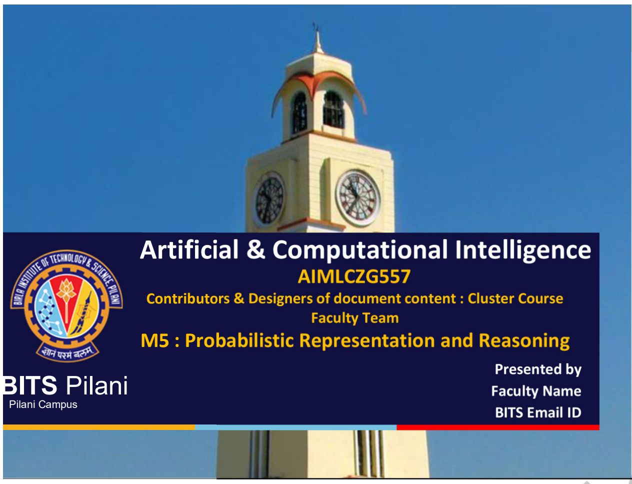
Required Reading: AIMA - Chapter # 4.1, #4.2, #5.1, #9

Next Session Plan:

- (Prerequisite Reading : Refresh the basics of probability , Bayes Theorem , Conditional Probability, Product Rule, Conditional Independence, Chain Rule)
- Inferences using Logic (Forward / Backward Chaining / DPLL algorithm)
- Bayesian Network
- Representation
- Inferences (Exact and approximate-only Direct sampling)

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials



Artificial & Computational Intelligence

AIMLCZG557

Contributors & Designers of document content : Cluster Course Faculty Team

M5 : Probabilistic Representation and Reasoning

Presented by
Faculty Name
BITS Email ID

BITS Pilani
Pilani Campus

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

M1	Introduction to AI
M2	Problem Solving Agent using Search
M3	Game Playing
M4	Knowledge Representation using Logics
M5	Probabilistic Representation and Reasoning
M6	Reasoning over time
M7	Ethics in AI

Module 5: Probabilistic Representation and Reasoning

A. Inference using full joint distribution
B. Bayesian Networks

I. Knowledge Representation

I. Conditional Independence

I. Exact Inference

I. Introduction to Approximate Inference

Reasoning

- Monotonic Reasoning
- Non- Monotonic Reasoning

Monotonic	Non-Monotonic
Consistent	Relaxed Consistency
Complete Knowledge	Incomplete Knowledge
Static	Dynamic
Discrete	Continuous & Learning Agent
Predicate Logic	Prob_abilistic_Model

Reasoning

- Monotonic Reasoning
- Non- Monotonic Reasoning
  Dependency Directed Backtracking: when a statement is deleted as " no more valid", other related statements have to be backtracked and they should be either deleted or new proofs have to be found for them. This is called dependency directed backtracking (DDB)

Uncertainty

You can reach Bangalore Airport from MG Road within 90 mins if you go by route A.
- There is uncertainty in this information due to partial observability and non determinism
- Agents should handle such uncertainty

Previous approaches like Logic represent all possible world states

Such approaches can't be used as multiple possible states need to be enumerated to
handle the uncertainty in our information

Uncertainty

You can reach Bangalore Airport from MG Road within 90 mins if you go by route A.

Road Block	Festival Season	Weekend	Observati on (20)	Prob
F	F	F	12	0.6
F	F	T	3	0.15
F	T	F	2	0.1
F	T	T	2	0.1
T	F	F	0	0
T	F	T	0	0
T	T	F	1	0.05
T	T	T	0	0
				=1

Probability Theory

Basics
Conditional Probability
Chain Rule
Independence
Conditional Independence
Belief Nets
Joint Probability distribution

Probability Basics – Refresher Self Study

Sample Space: Set of all possible outcomes.
- Ex: After tossing 2 coins, the set of all possible outcomes are
- {HH, HT, TH, TT}

Event: A subset of a sample space.
- An event of interest might be - {HH}

Probability Basics - Model

A fully specified **probability model** associates a numerical probability P(ω) with each possible world.

The **basic axioms**
1. Every possible world has a probability between 0 and 1
2. Sum of probabilities of possible worlds is 1 P(True) = 1
   P(False) = 0
3. P(a ∨ b) = P(a) + P(b) – P(a ∧ b)

E.g., P(HH) = 0.25; P(HT) = 0.25; P(TT) = 0.25, P(TH) = 0.25

$$0 \leq P(\omega) \leq 1 \text{ for every } \omega \text{ and } \sum_{\omega \in \Omega} P(\omega) = 1$$

Probability Basics – Exclusive / Exhaustive events

Mutually Exclusive Events:
- Two events (or propositions) are mutually exclusive or disjoint if they cannot both occur at the same time(be true).
- A clear example is the set of outcomes of a single coin toss, which can result in either heads or tails, but not both.

Exhaustive Events:
- A set of events is jointly or collectively exhaustive if at least one of the events must occur.
- E.g., when rolling a six-sided die, the events 1, 2, 3, 4, 5, and 6 are collectively exhaustive.

Probability Basics - Propositions

Probabilistic assertions (Propositions)

- Usually not a particular world event but about a set of them
- E.g., two dice when rolled, a proposition ϕ can be "the sum of dice is 11"

For any proposition ϕ,

P(ϕ) = P(sum = 11) = P((5, 6)) + P((6, 5))

= 1/36 + 1/36 = 1/18

Probability Basics – Unconditional/Prior

Unconditional / Prior probabilities:

Propositions like P(sum = 11) or P(two dices rolling equals) are called Unconditional or Prior probabilities

They refer to degree of belief in absence of any other information

$$P(a \mid b) = \frac{P(a \wedge b)}{P(b)}$$

$$P(a \wedge b) = P(a \mid b) P(b)$$

Probability Basics - Conditional

However, most of the time we have some information, we call it **evidence**

E.g., we can interested in two dice rolling a double (i.e., 1,1 or 2,2, etc) When one die has rolled 5 and the other die is still spinning

Here, we not interested in unconditional probability of rolling a double

Instead, the **conditional** or **posterior** probability for rolling a double given the

(first die|has rolled a 5

$P\ doubles$ $Die_1 = 5$) where | is pronounced "given"

E.g., if you are going for a dentist for a checkup, P(cavity) = 0.2

- If you have a toothache, then P(cavity | toothache) = 0.6

Independence

If we have two random variables, TimeToBnlrAirport and HyderabadWeather

P(TimeToBnlrAirport, HyderabadWeather)

To determine their relation, use the product rule

= P(TimeToBnlrAirport | HyderabadWeather) / P(HyderabadWeather)

However, we would argue that HyderabadWeather and TimeToBnlrAirport doesn't have any relation and hence

P(TimeToBnlrAirport | HyderabadWeather) = P(TimeToBnlrAirport)

This is called Independence or Marginal Independence

Independence between propositions a and b can be written as

$$P(a \mid b) = P(a) \quad \text{or} \quad P(b \mid a) = P(b) \quad \text{or} \quad P(a \wedge b) = P(a)P(b)$$

Bayes Rule

Using the product rule for propositions a and b

$$P(a \wedge b) = P(a \mid b)P(b) \quad \text{and} \quad P(a \wedge b) = P(b \mid a)P(a)$$

Equating the right hand sides and dividing by P(a)

$$P(b \mid a) = \frac{P(a \mid b)P(b)}{P(a)}$$

This is called the Bayes Rule

Conditional Independence

2 random variables A and B are conditionally independent given C iff

P(a, b | c) = P(a | c) P(b | c) for all values a, b, c

More intuitive (equivalent) conditional formulation

- A and B are conditionally independent given C iff

  P(a | b, c) = P(a | c) OR P(b | a, c) =P(b | c), for all values a, b, c
- Intuitive interpretation:

  P(a | b, c) = P(a | c) tells us that learning about b, given that we already know c, provides no change in our probability for a, i.e., b contains no information about a beyond what c provides

  P(R | F, P) = P(R | P)

Joint Probability Distributions

Instead of distribution over single variable, we can model distribution over multiple variables, separated by comma

E.g., **P**(A, B) = **P**(A | B) . **P**(B)

P(A, B) is the probability distribution over combination of all values of A and B E.g., if A = Weather and B = Cavity

$$\begin{aligned}
P(W = sunny \wedge C = true) &= P(W = sunny|C = true)\, P(C = true)\\
P(W = rain \wedge C = true) &= P(W = rain|C = true)\, P(C = true)\\
P(W = cloudy \wedge C = true) &= P(W = cloudy|C = true)\, P(C = true)\\
P(W = snow \wedge C = true) &= P(W = snow|C = true)\, P(C = true)\\
P(W = sunny \wedge C = false) &= P(W = sunny|C = false)\, P(C = false)\\
P(W = rain \wedge C = false) &= P(W = rain|C = false)\, P(C = false)\\
P(W = cloudy \wedge C = false) &= P(W = cloudy|C = false)\, P(C = false)\\
P(W = snow \wedge C = false) &= P(W = snow|C = false)\, P(C = false)\,.
\end{aligned}$$

Probabilistic Inference

Computation of posterior probabilities given observed evidence, i.e., full joint probability distribution

	toothache		*¬toothache*	
	catch	*¬catch*	*catch*	*¬catch*
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

Query: P(cavity ∨ toothache)

0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 = 0.28

Conditional Probability

Towards Chain Rule:

P(a | b) = P(a,b) / P(b)

P(a, b) = P(a | b) P(b)

P(a, b , c) = P(a , x) where x=bc

P(a,x) = P(a | x) . P(x)

= P(a | bc) . P(b, c)

= P(a | bc) . P(b | c). P(c)

Hence : P(a,b,c) = P(a | bc) . P(b | c). P(c)

Chain Rule : Generalization

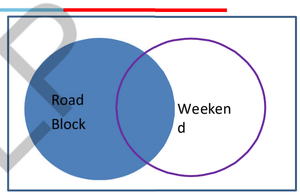
$$P(X_1, X_2, \ldots X_K) = \prod P(X_i \mid X_{i-1}, \ldots X_1)$$

Where i = k to 1 (reverse)

Probability Theory

Independence

P(a | b) = P(a)



Implication:

P(a | b) = P(a,b) / P(b)

P(a) = P(a,b) / P(b)

P(a,b) = P(a) . P(b)

Conditional Independence

P(a | b c) = P(a | c)

Probability Theory

Conditional Independence

P(a | b c) = P(a | c)

Extension:

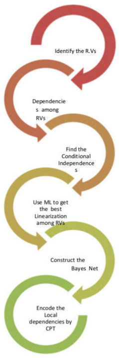
P(a b | c) = P(a | c) . P(b | c)

Building a Bayesian Network

Example Bayesian Net #1

A simple world with four random variables:

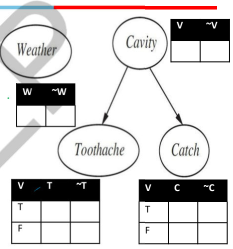
- Weather, Toothache, Cavity, Catch
- Weather is independent of other variables
- Toothache and Catch are conditionally independent given Cavity
- P(Toothache, Catch | Cavity) = P(Toothache | Cavity) . P(Catch | Cavity)
- Cavity is a direct cause of Toothache and Catch
- No direct relation between Toothache and Catch exists

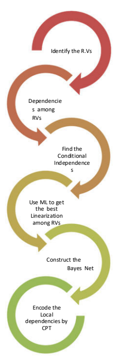


Example Bayesian Net #1

A simple world with four random variables:

- Weather, Toothache, Cavity, Catch
- Weather is independent of other variables
- Toothache and Catch are conditionally independent given Cavity
- P(Toothache, Catch | Cavity) = P(Toothache | Cavity) . P(Catch | Cavity)
- Cavity is a direct cause of Toothache and Catch
- No direct relation between Toothache and Catch exists

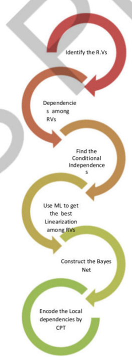




Example Bayesian Net #2

A Burglary Alarm System

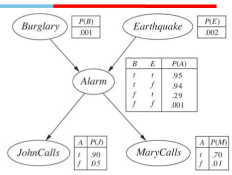
- Fairly reliable on detecting a burglary
- Also responds to earthquakes
- Two neighbors John and Mary are asked to call you at work when Burglary happens and they hear the Alarm
- John nearly always calls when he hears the alarm, however sometimes confuses the telephone ring with alarm and calls then too
- Mary like loud music and often misses the alarm altogether
- **Problem**: Given the information that who has / has not called we need to estimate the probability of a burglary

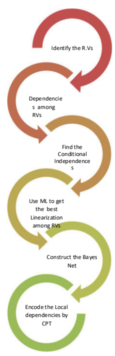


Example Bayesian Net #2

A Burglary Alarm System

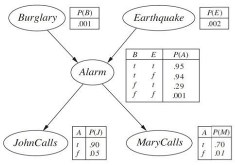
- Fairly reliable on detecting a burglary
- Also responds to earthquakes
- Two neighbors John and Mary are asked to call you at work when Burglary happens and they hear the Alarm
- John nearly always calls when he hears the alarm, however sometimes confuses the telephone ring with alarm and calls then too
- Mary like loud music and often misses the alarm altogether
- **Problem**: Given the information that who has / has not called we need to estimate the probability of a burglary





Examples

Calculate the probability that alarm has sounded, but neither burglary nor earthquake happened, and both John and Mary called



$$P(j,m,a,\neg b,\neg e) = P(j|a)P(m|a)P(a|\neg b \wedge \neg e)P(\neg b)P(\neg e)$$
$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998 = 0.000628$$

Example Bayesian Net #3

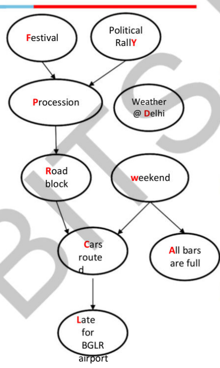
Traffic Prediction -Travel Estimation

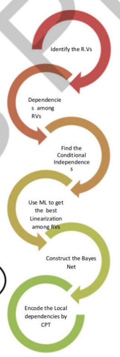
- AI system reminds traveler regarding start time
- Travel plan is to reach Delhi and the weather of Delhi may influence the accommodation plans
- Traveler always take car to reach airport
- Car may be rerouted either due to road block or weekday traffic during working hours which delays the arrival to airport
- Bars are always observed to be full on weekends
- Authorities block roads to safe the processions
- Processions observed during festive season or due to the political rally.
- **Problem**: Given the information that there is a political rally expected estimate the probability of late arrival

Example Bayesian Net #3

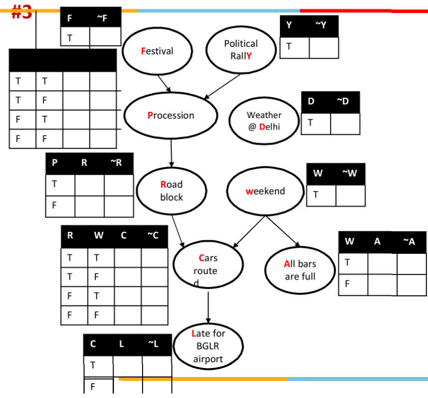
Traffic Prediction -Travel Estimation

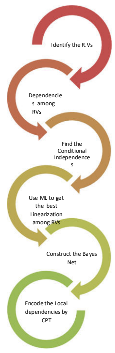
- AI system reminds traveler regarding start time
- Travel plan is to reach Delhi and the weather of Delhi may influence the accommodation plans
- Traveler always take car to reach airport
- Car may be rerouted either due to road block or weekday traffic during working hours which delays the arrival to airport
- Bars are always observed to be full on weekends
- Authorities block roads to safe the processions
- Processions observed during festive season or due to the political rally.
- **Problem**: Given the information that there is a political rally expected estimate the probability of late arrival



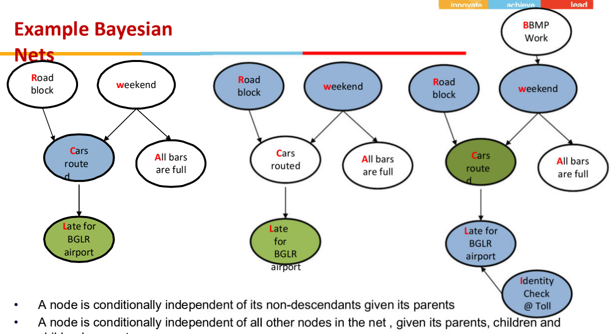


Example Bayesian Net #3



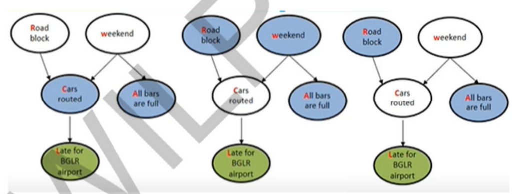


Example Bayesian Nets

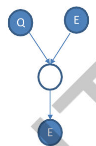


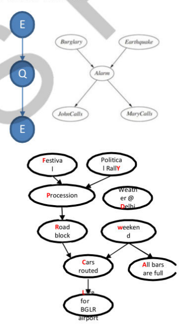
- A node is conditionally independent of its non-descendants given its parents
- A node is conditionally independent of all other nodes in the net , given its parents, children and children's parents.

Example Bayesian Nets

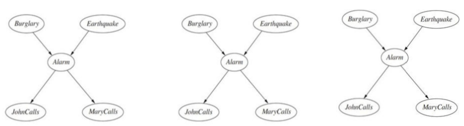


Belief Nets





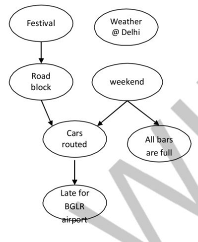
Belief Nets



Inferences in Bayesian Nets

Enumeration

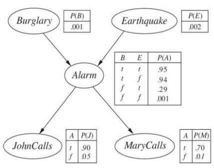
Examples



1. Calculate the probability that arrival at airport was delayed during a weekend but there was no road block or festival and car was not routed anywhere.
2. What is the probability that it is a festival season given cars where routed?
3. What is the probability that car arrived late at airport given it's a festival day?

Example s

. What is the probability that Burglary happened given John & Mary called the police



$$P(B \mid J,M) = \frac{P(B, J, M)}{P(J, M)}$$

$$P(B \mid J,M) = \frac{\sum_{A, E} P(J, M, A, B, E)}{\sum_{A, B, E} P(J, M, A, B, E)}$$

$$P(B|JM) + P(\neg B|JM) = 1$$

$$\frac{P(BJM)}{P(JM)} + \frac{P(\neg BJM)}{P(JM)} = 1$$

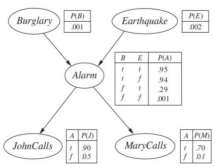
$$\frac{1}{P(JM)}\left[P(BJM)+P(\neg BJM)\right] = 1$$

$$\text{let } \alpha = \frac{1}{P(JM)}$$

$$\alpha = \frac{1}{P(BJM)+P(\neg BJM)} \rightarrow (1)$$

Example s

2. What is the probability that Burglary happened given John & Mary called the police



$$P(B \mid J,M) = \frac{P(B, J, M)}{P(J, M)}$$

$$P(B \mid J,M) = \frac{\sum_{A, E} P(J, M, A, B, E)}{\sum_{A, B, E} P(J, M, A, B, E)}$$

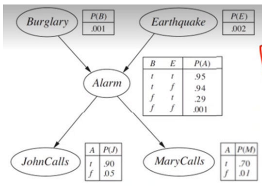
$$P(B|JM) + P(\neg B|JM) = 1$$

$$\frac{P(BJM)}{P(JM)} + \frac{P(\neg BJM)}{P(JM)} = 1$$

$$\frac{1}{P(JM)}\left[P(BJM)+P(\neg BJM)\right] = 1$$

$$\text{let } \alpha = \frac{1}{P(JM)}$$

$$\alpha = \frac{1}{P(BJM)+P(\neg BJM)} \rightarrow (1)$$



2. What is the probability that Burglary happened given John & Mary called the police

$$P(B \mid J,M) = \frac{P(B, J, M)}{P(J, M)}$$

$$P(B \mid J,M) = \frac{\sum_{A, E} P(J, M, A, B, E)}{\sum_{A, B, E} P(J, M, A, B, E)}$$

$$\boxed{P(BJM)} = \sum_{A,E} P(J,M,A,B,E)$$

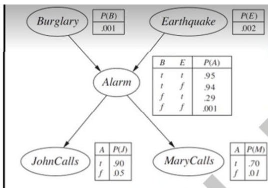
$$= \sum_{AE} P(J|MABE)\cdot P(M|ABE)\cdot P(A|BE)\cdot P(B|E)\cdot P(E)$$

$$= \sum_{AE} P(J|A)\cdot P(M|A)\cdot P(A|BE)\cdot P(B)\cdot P(E)$$

$$= \sum_{A}\sum_{E} P(J|A)\cdot P(M|A)\cdot P(A|BE)\cdot P(B)\cdot P(E)$$

$$= \sum_{A} \{P(J|A)\cdot P(M|A)\cdot P(A|BE)\cdot P(B)\cdot P(E)\} + \{P(J|A)\cdot P(M|A)\cdot P(A|B\neg E)\cdot P(B)\cdot P(\neg E)\}$$

$$= [P(J|A)\cdot P(M|A)\cdot P(A|BE)\cdot P(B)\cdot P(E)] + [P(J|A)\cdot P(M|A)\cdot P(A|B\neg E)\cdot P(B)\cdot P(\neg E)] + [P(J|\neg A)\cdot P(M|\neg A)\cdot P(\neg A|BE)\cdot P(B)\cdot P(E)] + [P(J|\neg A)\cdot P(M|\neg A)\cdot P(\neg A|B\neg E)\cdot P(B)\cdot P(\neg E)]$$



2. What is the probability that Burglary happened given John & Mary called the police

$$P(B \mid J,M) = \frac{P(B, J, M)}{P(J, M)}$$

$$P(B \mid J,M) = \frac{\sum_{A, E} P(J, M, A, B, E)}{\sum_{A, B, E} P(J, M, A, B, E)}$$

$$\boxed{P(BJM)} = \sum_{A,E} P(J,M,A,B,E)$$

$$= \sum_{AE} P(J|MABE)\cdot P(M|ABE)\cdot P(A|BE)\cdot P(B|E)\cdot P(E)$$

$$= \sum_{AE} P(J|A)\cdot P(M|A)\cdot P(A|BE)\cdot P(B)\cdot P(E)$$

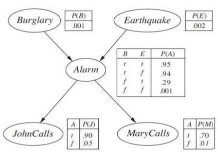
$$= \sum_{A}\sum_{E} P(J|A)\cdot P(M|A)\cdot P(A|BE)\cdot P(B)\cdot P(E)$$

$$= \sum_{A} \{P(J|A)\cdot P(M|A)\cdot P(A|BE)\cdot P(B)\cdot P(E)\} + \{P(J|A)\cdot P(M|A)\cdot P(A|B\neg E)\cdot P(B)\cdot P(\neg E)\}$$

$$= [P(J|A)\cdot P(M|A)\cdot P(A|BE)\cdot P(B)\cdot P(E)] + [P(J|A)\cdot P(M|A)\cdot P(A|B\neg E)\cdot P(B)\cdot P(\neg E)] + [P(J|\neg A)\cdot P(M|\neg A)\cdot P(\neg A|BE)\cdot P(B)\cdot P(E)] + [P(J|\neg A)\cdot P(M|\neg A)\cdot P(\neg A|B\neg E)\cdot P(B)\cdot P(\neg E)]$$

Example s

3. What is the probability that John calls given earthquake occurred?



$$P(J \mid E) = \frac{P(J, E)}{P(E)}$$

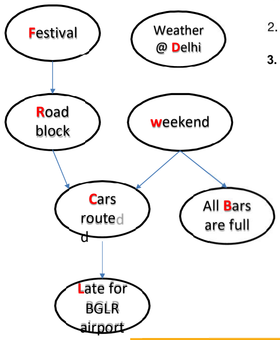
$$P(J \mid E) = \frac{\sum_{M, A, B} P(J, M, A, B, E)}{\sum_{J, M, A, B} P(J, M, A, B, E)}$$

Inferences in Bayesian Nets

Variable Elimination

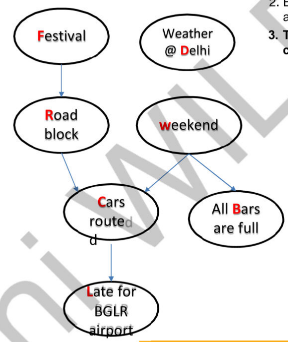
Reduce Guaranteed Independent nodes

D-Connectedness Vs D-Separation



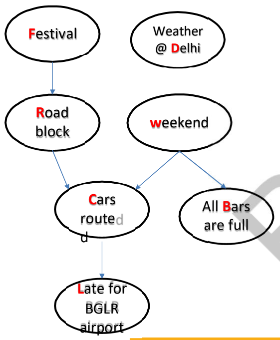
1. Each variable is conditionally independent of its non-descendants, given its parents
2. Eliminate the hidden variables that is neither a query nor an evidence
3. **Two variables are d-separated if they are conditionally independent given evidences**

D-Connectedness Vs D-Separation



1. Each variable is conditionally independent of its non-descendants, given its parents
2. Eliminate the hidden variables that is neither a query nor an evidence
3. **Two variables are d-separated if they are conditionally independent given evidences**

Try it & Test



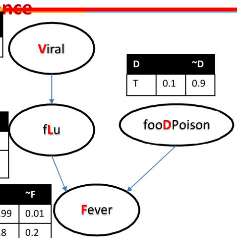
X	Y	Evidence Z	d-sep?
F	W	C	No
L	W	R	No
R	**L**	**C**	**Yes**
B	R	C	No

- P(R | L , C) = P(R| L)

R & L are d-separated ie., conditionally independent

D-Separation in Inference

V		~V
T	0.6	0.4



D		~D
T	0.1	0.9

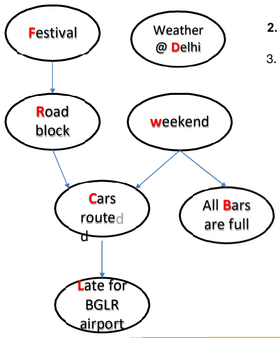
V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

X	Y	Evidence Z	d-sep?
V	F	L	Yes
V	D	L	Yes

- P(V | F , L)
- P(V | D , L)

Variable Elimination



1. Each variable is conditionally independent of its non-descendants, given its parents
2. **Eliminate the hidden variables that is neither a query nor evidence**
3. Two variables are d-separated if they are conditionally independent given evidences

- $P(B) = \sum_{L,B,W,R,F} P(L, C, B, W, R, F)$
  $= \sum_L \sum_B P(L|C) \cdot P(B | W) \cdot \sum_W P(C| W, R) \cdot \sum_R P(R | F) \cdot \sum_F P(F)$
  $= \mathbf{P(B | W)}$

All other variables are hidden w.r.t to B as (L, C, R, F) are neither evidence nor query nor (L, C, R, F) ∈ Ancestors(W, B)

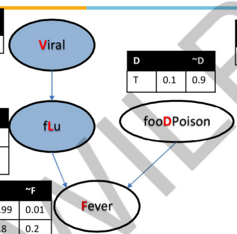
This is variable elimination example targeting irrelevant nodes

Inference

Variable Elimination: V

V		~V
T	0.6	

0.4



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

P(V)
P(L|V)
P(D)
P(F|L,D)

V		~V
T	0.6	0.4

V	L	~L
T	0.8	0.2
F	0.01	0.99

V	L	
T	T	0.48
T	F	0.12
F	T	0.004
F	F	0.396

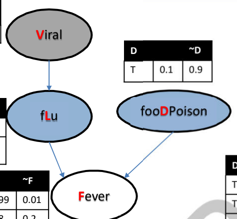
L	
T	0.484
F	0.516

Inference

Variable Elimination: L,D

V		~V
T	0.6	

0.4



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

P(L)
P(D)
P(F|L,D)

L	
T	
F 0.484	0.516

D		~D
T	0.1	0.9

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

D	L	F	
T	T	T	
T	F	T	
F	T	T	
F	F	T	
T	T	F	
T	F	F	
F	T	F	
F	F	F	

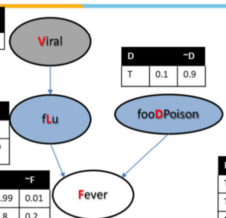
L	F	
T	T	0.08928
F	T	0.37174
T	F	0.0108
F	F	0.52818

F	
T	0.46102
F	0.53898

Inference

Variable Elimination: L,D

V		~V
T	0.6	0.4



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

P(L)
P(D)
P(F|L,D)

L	
T	0.484
F	0.516

D		~D
T	0.1	0.9

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

D	L	F	
T	T	T	0.048
T	F	T	0.04128
F	T	T	0.34852
F	F	T	0.02322
T	T	F	0.00048
T	F	F	0.01032
F	T	F	0.087
F	F	F	0.44118

L	F	
T	T	0.39652
F	T	0.0645
T	F	0.08748
F	F	0.4515

F	
T	0.46102
F	0.53898

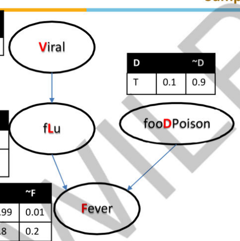
Approximate Inferences in Bayesian Nets
Introduction

Prior Sampling

Sample Generation by Randomization

V		~V
T	0.6	

0.4



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

V	L	D	F

0.3, 0.2, 0.6, 0.58, 0.9,0.55…………..

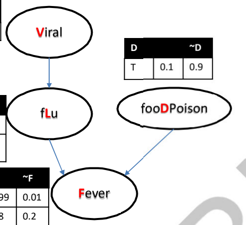
0.73, 0.87, 0.15, 0.6, 0.57, 0.85, 0.12, 0.004, 0.93 , 0.0002,

Prior Sampling

Sample Generation by Randomization

V		~V
T	0.6	

0.4



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

V	L	D	F
T	T	F	T
F	F	F	F
T	F	F	T
F	T	F	T
..			
…..			

0.3, 0.2, 0.6, 0.58, 0.9,0.55…………..

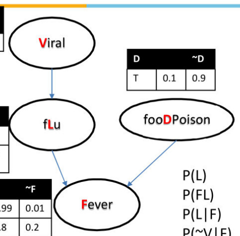
0.73, 0.87, 0.15, 0.6, 0.57, 0.85, 0.12, 0.004, 0.93 , 0.0002,

Prior Sampling

Inference

V		~V
T	0.6	

0.4



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

V	L	D	F
T	T	F	T
F	F	F	F
T	F	F	T
F	T	F	T
T	T	F	T
T	F	F	F
F	F	F	T
T	F	F	F

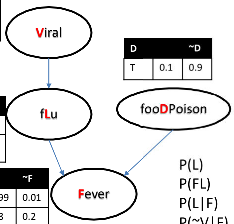
P(L) = 3/8
P(FL) = 3/8
P(L|F) = 3/5
P(~V|F) = 2/5
P(L|V~F) = 0
P(F|D) = ?????

Rejection Sampling

Sample Generation by Randomization

V		~V
T	0.6	

0.4



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

V	L	D	F
…..			

P(L) = 3/8
P(FL) = 3/8
P(L|F) = 3/5
P(~V|F) = 2/5
P(L|V~F) = 0
P(F|D) = ?????

0.3, 0.2, 0.6, 0.58, 0.9,0.555,0.38…………..

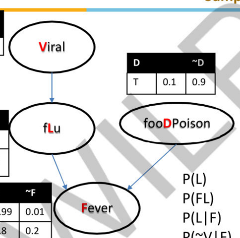
0.73, 0.87, 0.15, 0.6, 0.57, 0.85, 0.12, 0.004, 0.93 , 0.0002,

Rejection Sampling

Sample Generation by Randomization

V		~V
T	0.6	

0.4



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

V	L	D	F
T	T	F	
T	T	F	
T	T	F	
F	F	T	F
T	F	T	T
…..			

P(L) = 3/8
P(FL) = 3/8
P(L|F) = 3/5
P(~V|F) = 2/5
P(L|V~F) = 0
P(F|D) = ?????

0.3, 0.2, 0.6, 0.58, 0.9,0.555,0.38…………..

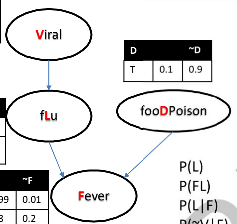
0.73, 0.87, 0.15, 0.6, 0.57, 0.85, 0.12, 0.004, 0.93 , 0.0002,

Rejection Sampling

Inference

V		~V
T	0.6	

0.4



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

V	L	D	F
T	T	T	T
F	F	T	F
T	F	T	T
F	T	T	T
T	T	T	T
T	F	T	F
F	F	T	T
T	F	T	F

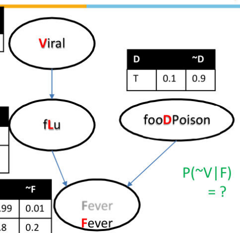
P(L) = 3/8
P(FL) = 3/8
P(L|F) = 3/5
P(~V|F) = 2/5
P(L|V~F) = 0
P(F|D) = 5/8

Rejection Sampling

Sample Generation by Randomization

V		~V
T	0.6	

0.4



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

V	L	D	F
T	T	F	T
F	F	F	
T	F	F	T
F	T	F	T
…..			

P(~V|F) = ?

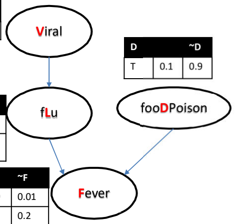
0.3, 0.2, 0.6, 0.58, 0.9,0.555,………..

0.73, 0.87, 0.15, 0.6, 0.57, 0.85, 0.12, 0.004, 0.93 , 0.0002,

Likelihood Weighing

Sample Generation by Randomization

V	~V
0.6	
0.4	



D		~D
T	0.1	0.9

L	~L
0.8	0.2
0.01	0.99

D	F	~F
T	0.99	0.01
F	0.8	0.2
T	0.8	0.2
F	0.05	0.95

V	L	D	F	wgt
F		T		
F		T		
F		T		
F		T		
F		T		
F		T		
F		T		

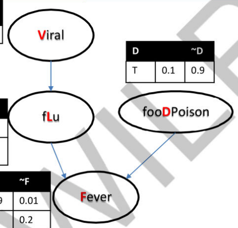
P(~F|D,~V) = 0.04 / 7*0.04

0.3, 0.2, 0.58, 0.73, 0.87, 0.15, 0.6, 0.57, 0.85, 0.12, 0.004, 0.93 , 0.0002, 0.99,,..........

Likelihood Weighing

Sample Generation by Randomization

V		~V
T	0.6	
	0.4	



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

V	L	D	F	wgt
F	F	T	T	0.4*1* 0.1 *1=
F	F	T	T	
F	F	T	T	
F	F	T	T	
F	F	T	T	
F	T	T	T	
F	T	T	F	

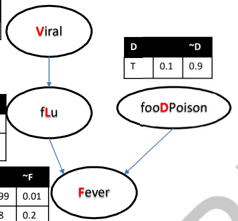
P(~F|D,~V) = 0.04 / 7*0.04

0.3, 0.2, 0.58, 0.73, 0.87, 0.15, 0.6, 0.57, 0.85, 0.12, 0.004, 0.93 , 0.0002, 0.99,,..........

Likelihood Weighing

Inference

V	~V
0.6	
0.4	



D		~D
T	0.1	0.9

L	~L
0.8	0.2
0.01	0.99

D	F	~F
T	0.99	0.01
F	0.8	0.2
T	0.8	0.2
F	0.05	0.95

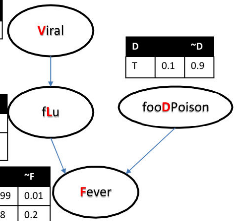
V	L	D	F	wgt
F	F	T	F	0.4*1* 0.1 *1=
F	T	T	T	0.4*1* 0.1 *1=
F	F	T	T	0.4*1* 0.1 *1=
F	F	T	F	0.4*1* 0.1 *1=

P(F|D,~V) = 0.04+0.04 / 4*0.04

Likelihood Weighing

Inference

V		~V
T	0.6	
	0.4	



D		~D
T	0.1	0.9

V	L	~L
T	0.8	0.2
F	0.01	0.99

L	D	F	~F
T	T	0.99	0.01
T	F	0.8	0.2
F	T	0.8	0.2
F	F	0.05	0.95

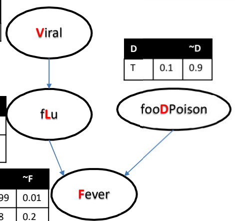
V	L	D	F	wgt
F	F	T	F	
F	F	T	T	
F	F	T	T	
T	F	T	F	

P(F|D,~L) = 0.099+0.099 / (3*0.099 + 0.02)

Likelihood Weighing

Inference

V	~V
0.6	
0.4	



D		~D
T	0.1	0.9

L	~L
0.8	0.2
0.01	0.99

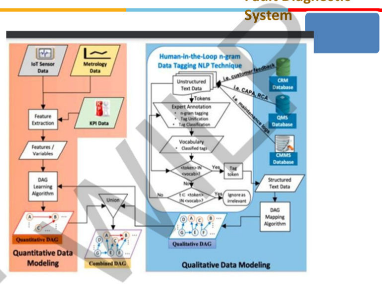
D	F	~F
T	0.99	0.01
F	0.8	0.2
T	0.8	0.2
F	0.05	0.95

V	L	D	F	wgt
F	F	T	F	1*0.99* 0.1 *1=
F	F	T	T	1*0.99* 0.1 *1=
F	F	T	T	1*0.99* 0.1 *1=
T	F	T	F	1*0.2* 0.1 *1=

P(F|D,~L) = 0.099+0.099 / (3*0.099 + 0.02)

Bayesian Network

Fault Diagnostic System

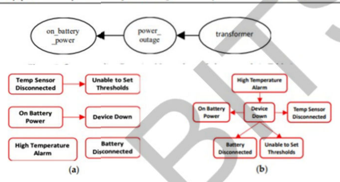


Source Credit : Sensors 2021 : Fusion-Learning of Bayesian Network Models for Fault Diagnostics

Bayesian Network

Fault Diagnostic System

Raw Data	Short Description		Resolution Notes			
	On battery power		Power outage due to transformer fire			
Classified Tags	Symptom		Cause(s)		Link	
	on_battery_power		power_outage, transformer_fire		due_to	
BN Mapping	Child Variable	Child State	Parent Variable	Parent State	Ancestor Variable	Ancestor State
	on_battery_power	yes	power_outage	yes	transformer	Fire



Source Credit : Sensors 2021 : Fusion-Learning of Bayesian Network Models for Fault Diagnostics

Bayesian Network

Fault Diagnostic System

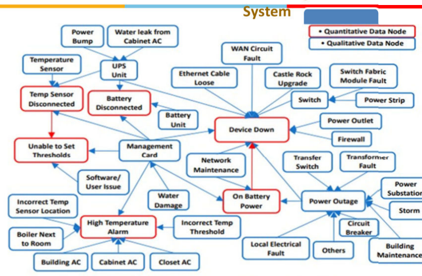
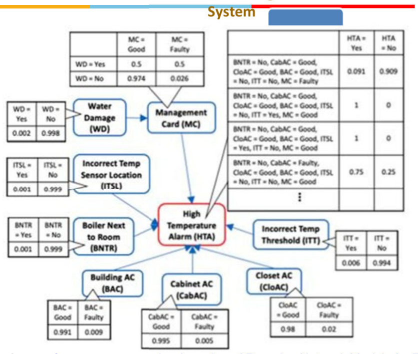


Figure 8. Fused Bayesian Network structure for top six occurring UPS messages.

Source Credit : Sensors 2021 : Fusion-Learning of Bayesian Network Models for Fault Diagnostics

Bayesian Network

Fault Diagnostic System



Source Credit : Sensors 2021 : Fusion-Learning of Bayesian Network Models for Fault Diagnostics

Required Reading: AIMA - Chapter # 4.1, #4.2, #5.1, #9 Refer to the handout

Next Session Plan:

- Hidden Markov Models

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

M1	Introduction to AI
M2	Problem Solving Agent using Search
M3	Game Playing
M4	Knowledge Representation using Logics
M5	Probabilistic Representation and Reasoning
M6	Reasoning over time
M7	Ethics in AI

Module 6: Reasoning over time

Reasoning Over Time

A. Time and Uncertainty

B. Inference in temporal models

C. Overview of HMM

D. Learning HMM Parameters using EM Algorithm

E. Applications of HMM

Learning Objective

1. Understand the relationship between Time & Uncertainty
1. Recognize the transition model of Markov Model
1. Relate to the application of the Hidden Markov Model

Module 6: Reasoning over time

Reasoning Over Time

A. Time and Uncertainty
B. Inference in temporal models
C. Introduction to Hidden Markov Model
D. Applications of HMM

Sequential Decision Problems & Markov Decision Process

Markov Decision Process

Sequential Problem | Partial Observability | Belief System

Modelling sequences of random events and transitions between states over time is known as Markov chain

Agents in partially observable environment should keep a track of current state to the extent allowed by sensors
E.g., Robot moving in a new maze

Agent maintains a **belief state** representing the current possible world states

Transition Model / Probability Matrix :
Using belief state and transition model,
the agent can how the world might evolve in
next time step. To capture the degree of belief
we will use Probability Theory. We model the change in world
using a variable for each aspect of state and
at each point in time.
Current state depends only finite number of previous states.

C	M	
0.40	0.20	C
0.60	0.80	M

Markov Decision Process

Time - Uncertainty | States - Observations

Static World: Each random variable would have a single fixed value
E.g., Diagnosing a broken car

Dynamic World: The state information keeps changing with time
E.g., treating a diabetic patient, tracking the location of robot, tracking economic activity of a nation

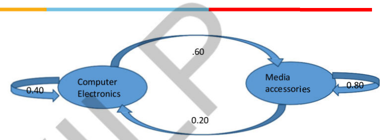
Time slices: World is observed in time slices. Each slice has a set of random variables, some observable and some not.

Assumption: We will assume same subset of random variables are observable in each time slice

E_t - set of observable random variables at time t
X_t - set of unobserved random variables at time t

C	M	
0.40	0.20	C
0.60	0.80	M

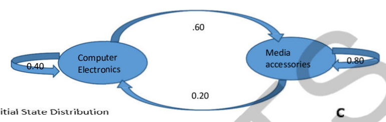
Markov Model- Example 1



Transition Model

C	M	
0.40	0.20	C
0.60	0.80	M

Markov Model



Current State: Initial State Distribution

1	C
0	M

C	M	
0.40	0.20	C
0.60	0.80	M

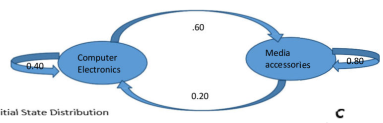
Next State : Likely to buy Media accessories on next visit

0.40	C
0.60	M

Next State : Likely to buy Media accessories on next visit

0.28	C
0.72	M

Markov Model



Current State: Initial State Distribution

1	C
0	M

C	M	
0.40	0.20	C
0.60	0.80	M

Next State : Likely to buy Media accessories on next visit

0.40	C
0.60	M

Next State : Likely to buy Media accessories on next visit

0.28	C
0.72	M

Markov Process

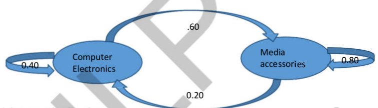
States | Observations | Assumptions

Modelling sequences of random events and transitions between states over time is known as Markov chain

Transition Model / Probability Matrix :
Current state depends only finite number of previous states. :

C	M	
0.40	0.20	C
0.60	0.80	M

Markov Model



Current State: Initial State Distribution

1	C
0	M

C	M	
0.40	0.20	C
0.60	0.80	M

Next State : Likely to buy Media accessories on next visit

0.40	C
0.60	M

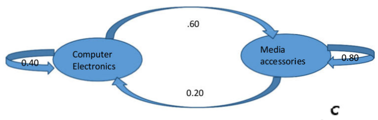
Next State : Likely to buy Media accessories on next visit

0.28	C
0.72	M

Inference in temporal Models

Markov Model

Inference Type 1



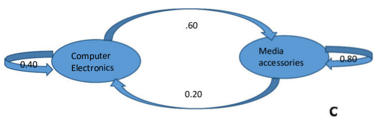
C	M	
0.40	0.20	C
0.60	0.80	M

What is the probability that the purchasing behaviour of the customer is in below sequential order only?Initial Probability Matrix is P(C) = 1 , P(M) = 0
(Computer, Media, Media, Computer)

Apply Bayes chain rule:
P(Computer, Media, Media, Computer) = P(C) * P(M|C) * P(M|M) * P(C|M) = 0.096

Markov Model

Inference Type 2



C	M	
0.40	0.20	C
0.60	0.80	M

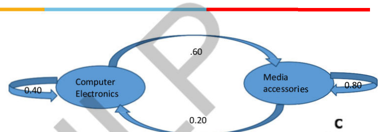
What is the probability that the customer who purchased Media accessories will keep coming back to purchase media accessories in the next 2 consecutive visits only?

Derive Initial prob values & Apply Bayes chain rule on the pattern exhibited:
Initial Probability Matrix is P(M) = 1 , P(C) = 0

P(Media, Media, Media, Computer) = P(M) * P(M|M) * P(M|M) * P(C|M) = 0.128

Markov Model

Inference Type 3



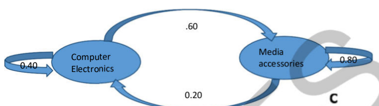
C	M	
0.40	0.20	C
0.60	0.80	M

Given the evidence that the customer walked into the store and bought a computer electronics, find the expected purchase pattern in the next 3 visits

Derive Initial prob values & Apply Bayes chain rule and reverse predict the combination on the most likely pattern (Similar to Viterbi Algorithm):
Initial Probability Matrix is P(C) = 1 , P(M) = 0
P(Computer , X, Y, Z) = P(Computer) * P(X|Computer) * P(Y|X) * P(Z|X) =
1*0.6*0.8*0.8 → Produces max values
Ans : Pattern = (Computer, Media, Media, Media)

Markov Model

Inference Type 3



C	M	
0.40	0.20	C
0.60	0.80	M

Given the evidence that the customer walked into the store and bought a computer electronics, find the expected purchase pattern in the next 3 visits

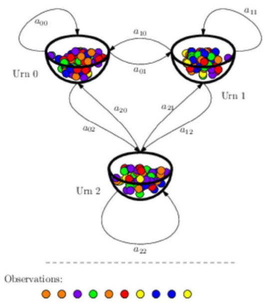
Derive Initial prob values & Apply Bayes chain rule and reverse predict the combination on the most likely pattern (Similar to Viterbi Algorithm):
Initial Probability Matrix is P(C) = 1 , P(M) = 0
P(Computer , X, Y, Z) = P(Computer) * P(X|Computer) * P(Y|X) * P(Z|X) =
1*0.6*0.8*0.8 → Produces max values
Ans : Pattern = (Computer, Media, Media, Media)

HMM

Markov Process

States | Observations | Assumptions

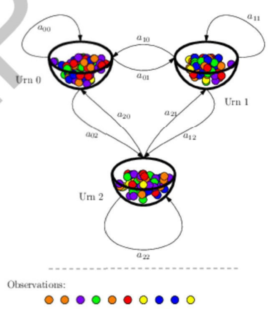
Standard Mathematical Example:
Urn & Ball Model



Markov Process

States | Observations | Assumptions

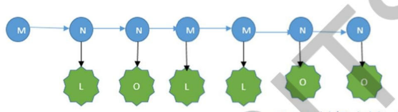
Standard Mathematical Example:
Urn & Ball Model



Hidden Markov Model

States | Observations | Assumptions

Time Slice **(t)**	0	1	2	3	4	5	6	P(**O_t | O_{t-1}**)
Observed Evidence **(O_t/ E_t)**	-	Late	OnTime	Late	Late	Ontime	Ontime	……
Unobserved State **(U_t /X_t/ Q_t)**	Meeting	No Meeting	No Meeting	Meeting	Meeting	No Meeting	No Meeting	……



Transition Model / Probability Matrix

P(**U_{t-1} = No Meeting)**	P(**U_{t-1} = Meeting)**	←Previous
0.5	0.67	P(**U_t = No Meeting)**
0.5	0.33	P(**U_t = Meeting)**

Evidence / Sensor Model/ Emission Probability Matrix

P(**U_t = No Meeting)**	P(**U_t = Meeting)**	←Unobserved Evidence v
0.9	0.3	P(**O_t = OnTime)**
0.1	0.7	P(**O_t = Late)**

Hidden Markov Process

States | Observations | Assumptions

Modelling sequences of random events and transitions between states over time is known as Morkov chain

Hidden Markov Process models events as the state sequences that are not directly observable but only be approximated from the sequence of observations produced by the system

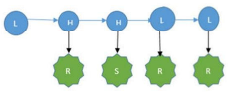
Transition Model / Probability Matrix :
Current state depends only finite number of previous states. :

Evidence / Sensor Model/ Emission Probability Matrix :
Current Evidence or Observation depends Current State of the world. Given the Current State Knowledge of the world, observation doesn't depend on history:

Hidden Morkov Model

States | Observations | Assumptions

Time Slice **(t)**	0	1	2	3	4	…………	P(O_t | O_{t-1}, O_{t-2})
Observed Evidence **(O_t)**	-	Rainy	Sunny	Rainy	Rainy		
Unobserved State**(U_t)**	Low Pressure	High Pressure	High Pressure	Low Pressure	Low Pressure		



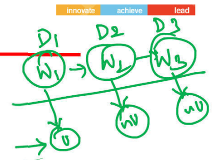
Transition Model / Probability Matrix

P(**U_{t-2} = LP, U_{t-1} =HP)**	P(**U_{t-2} = HP,U_{t-1} = HP)**	P(**U_{t-2} = HP,U_{t-1} = LP)**	P(**U_{t-2} = LP,U_{t-1} = LP)**	←Previous
0.2	0.40	0.85	0.5	P(**U_t = LP)**
0.8	0.60	0.15	0.5	P(**U_t = HP)**

Evidence / Sensor Model/ Emission Probability Matrix

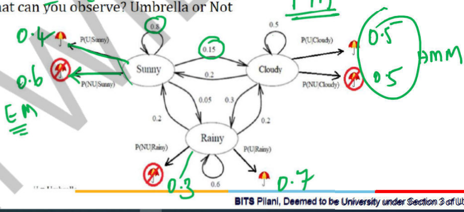
P(**X_t = LP)**	P(**X_t = HP)**	←Unobserved Evidence v
0.8	0.4	P(**E_t = Rainy)**
0.2	0.6	P(**E_t = Sunny)**

3rd



- Imagine: You were locked in a room for several days and you were asked about the weather outside. The only piece of evidence you have is whether the person who comes into the room bringing your daily meal is carrying an umbrella or not.
- What is hidden? Sunny, Rainy, Cloudy
- What can you observe? Umbrella or Not

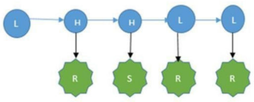
TM



Hidden Morkov Model

Filtering	Prediction	Smoothing	Most Likely Explanation
P(L $_3$ | R-S-R-R) P(X_t | $E_{1..t}$)	P(L $_3$ | R-S) P(X_{t+k} | $E_{1..t}$)	P(H $_2$ | R-S-R-R) P($X_{k,\ 0>k>t}$ | $E_{1..t}$)	P(H-H-L-L | R-S-R-R) argmax $X_{1..t}$: P($X_{1..t}$ | $E_{1..t}$)

In your Text book another example for these inferences is explained "Task of predicting the weather condition by a security personnel sitting in an underground secret installation by observing the state of an employee who either umbrella or don't" Kindly check it and work it out as additional practice

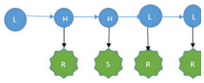


Hidden Morkov Model

Inference: Type -1

Sequence Evaluation : Likely hood Computation : Forward Algorithm
Find the probability of occurrence of
this weather sequence observation: **S-S-R**

Intuition:$P(E_{1...t}) = \sum_{i=1}^{N} P(E_{1...t} \mid X_{1...t}) * P(X_{1...t}) =$
$= \sum_{i=1}^{N} \prod_{j=1}^{t} P(Ej \mid Xj) * P(Xj \mid Xj_{-1})$



Transition Model / Probability Matrix

P(**U_{t-1} = HP)**	P(**U_{t-1} = LP)**	←Previous
0.2	0.5	P(**U_t = LP)**
0.8	0.5	P(**U_t = HP)**

Evidence / Sensor Model/ Emission Probability Matrix

P(**X_t = LP)**	P(**X_t = HP)**	←Unobserved Evidence v
0.8	0.4	P(**E_t = Rainy)**
0.2	0.6	P(**E_t = Sunny)**

P(SSR)

$$= \sum_X P(SSR,\ X) = \sum_X P(SSR, X_1X_2X_3)$$

$$= \sum_X P(R, X_3, S, X_2, S, X_1) = \sum_X P(R \mid X_3) * P(X_3 \mid X_2) * P(S \mid X_2) * P(X_2 \mid X_1) * P(S \mid X_1) * P(X_1 \mid X_0)$$

$$= \Sigma_X\ P(R \mid X_3) * P(S \mid X_2) * P(S \mid X_1) * P(X_3 \mid X_2) * P(X_2 \mid X_1) * P(X_1 \mid X_0)$$

$$= \sum_X \prod_{j=1}^{t} P(Ej \mid X_j) * P(Xj \mid X_{j-1})$$

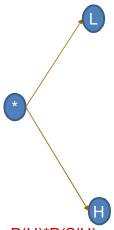
Hidden Morkov Model

Forward Propagation Algorithm

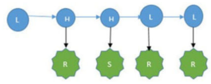
Find the probability of occurrence of this Pressure sequence observation: **S-S-R**

Initialization Phase:

P(L)*P(S|L) = 0.5*0.2 = 0.1 →0.25



P(H)*P(S|H) = 0.5*0.6 = 0.3 → 0.75



Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

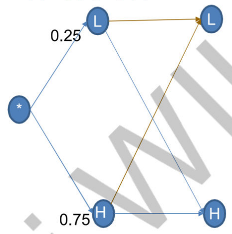
$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

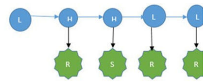
Hidden Morkov Model

Forward Propagation Algorithm : S-S-R

P(L)*P(L|L)*P(S|L) = 0.25*0.5*0.2 = **0.025**

P(H)*P(L|H)*P(S|L) = 0.75*0.2*0.2 = **0.03**





Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

P(L)*P(H|L)*P(S|H) = 0.25*0.5*0.6 = **0.075**

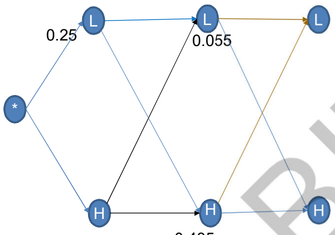
P(H)*P(H|H)*P(S|H) = 0.75*0.8*0.6 = **0.36**

Hidden Morkov Model

Forward Propagation Algorithm : S-S-R

P(L)*P(L|L)*P(R|L) = 0.055*0.5*0.8 = **0.022**

P(H)*P(L|H)*P(R|L) = 0.435*0.2*0.8 = **0.0696**



Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

P(L)*P(H|L)*P(R|H) = 0.055*0.5*0.4 = **0.011**

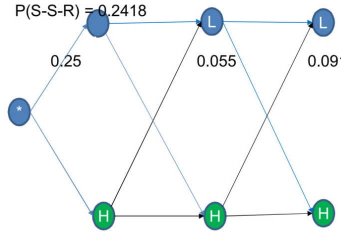
P(H)*P(H|H)*P(R|H) = 0.435*0.8*0.4 = **0.1392**

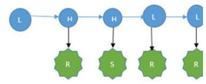
Hidden Morkov Model

Forward Propagation Algorithm : S-S-R

Termination Phase:

P(S-S-R) = 0.2418





Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

Hidden Morkov Model

Filtering : P(SecondUrnIsSelected $_3$ | Red-Blue-Blue-Yellow)

$P(X_t \mid E_{1...t})$

Prediction: P(FirstUrnWillbeSelected $_3$ | Red-Yellow)

$P(X_{t+k} \mid E_{1...t})$

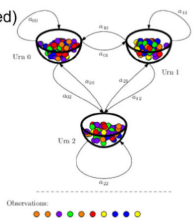
Smoothing: P(ThirdUrnWasSelected $_2$ | Red-Yellow-Red-Red)

$P(X_{k,\ o>k>t} \mid E_{1...t})$

Most Likely Explanation (or) Viterbi Algorithm

P(Urn1-Urn2-Urn1 | Red-Yellow-Yellow)

argmax $X_{1...t}$: $P(X_{1...t} \mid E_{1...t})$



Hidden Morkov Model

Inference: Type -2

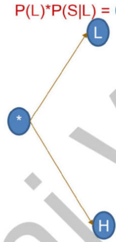
Most Likely Explanation : Veterbi Algorithm

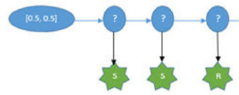
Find the pattern in pressure that might have caused this observation: **S-S-R**

argmax $X_{1...t}$: $P(X_{1...t} \mid E_{1...t})$

MM Inf

P(L)*P(S|L) = 0.5*0.2 = 0.1 → 0.25





Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

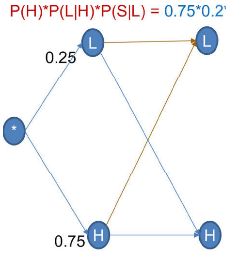
P(H)*P(S|H) = 0.5*0.6 = 0.3 → 0.75

Hidden Morkov Model

Veterbi Algorithm : S-S-R

P(L)*P(L|L)*P(S|L) = 0.25*0.5*0.2 = 0.025

P(H)*P(L|H)*P(S|L) = 0.75*0.2*0.2 = **0.03**



Transition Model / Probability Matrix

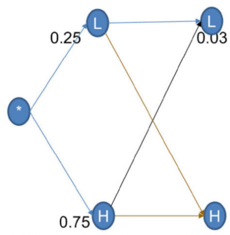
$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

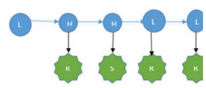
Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

Hidden Morkov Model

Veterbi Algorithm : S-S-R





Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

P(L)*P(H|L)*P(S|H) = 0.25*0.5*0.6 = 0.075

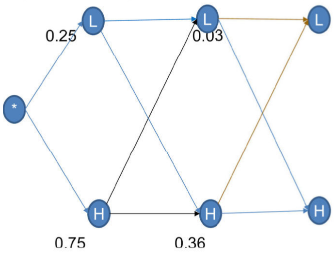
P(H)*P(H|H)*P(S|H) = 0.75*0.8*0.6 = **0.36**

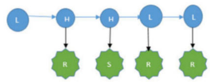
Hidden Morkov Model

Veterbi Algorithm : S-S-R

P(L)*P(L|L)*P(R|L) = 0.03*0.5*0.8 = 0.012

P(H)*P(L|H)*P(R|L) = 0.36*0.2*0.8 = **0.0576**





Transition Model / Probability Matrix

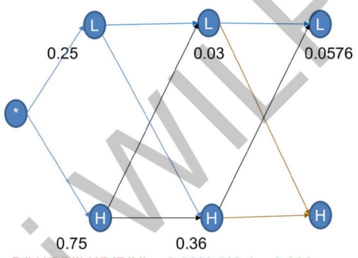
$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

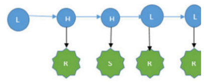
Hidden Morkov Model

Veterbi Algorithm : S-S-R



P(L)*P(H|L)*P(R|H) = 0.03*0.5*0.4 = 0.006

P(H)*P(H|H)*P(R|H) = 0.36*0.8*0.4 = **0.1152**



Transition Model / Probability Matrix

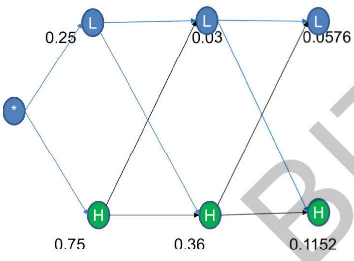
$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

Hidden Morkov Model

Veterbi Algorithm : S-S-R



Transition Model / Probability Matrix

$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

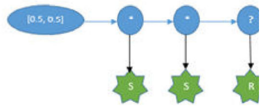
Hidden Morkov Model

Inference: Type -3

Filtering : Forward Propagation Algorithm

Find the Current Pressure if sequence of weather observations recorded are: **S-S-R**

Intuition: $P(E_{1...t}) = \sum_{i=1}^{N} P(E_{1...t} \mid X_{1...t}) * P(X_{1...t}) = \sum_{i=1}^{N} \prod_{j=1}^{t} P(Ej|\, Xj) * P(Xj|\, Xj_{-1})$



Transition Model / Probability Matrix

$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

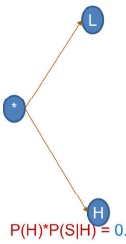
Hidden Morkov Model

Forward Propagation Algorithm

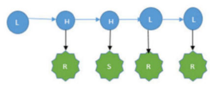
Pressure sequence observation: **S-S-R**

Initialization Phase:

P(L)*P(S|L) = 0.5*0.2 = 0.1 → 0.25



P(H)*P(S|H) = 0.5*0.6 = 0.3 → 0.75



Transition Model / Probability Matrix

$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

Evidence / Sensor Model/ Emission Probability Matrix

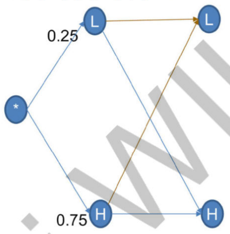
$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

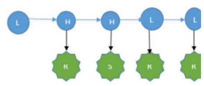
Hidden Morkov Model

Forward Propagation Algorithm : S-S-R

P(L)*P(L|L)*P(S|L) = 0.25*0.5*0.2 = **0.025**

P(H)*P(L|H)*P(S|L) = 0.75*0.2*0.2 = **0.03**





Transition Model / Probability Matrix

$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

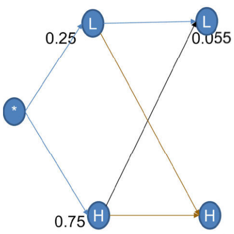
Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

Recursion Phase:

Hidden Morkov Model

Forward Propagation Algorithm : S-S-R



P(L)*P(H|L)*P(S|H) = 0.25*0.5*0.6 = **0.075**

P(H)*P(H|H)*P(S|H) = 0.75*0.8*0.6 = **0.36**

Transition Model / Probability Matrix

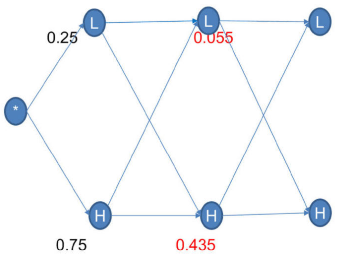
$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

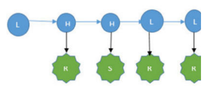
Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

Hidden Morkov Model

Forward Propagation Algorithm : S-S-R





Transition Model / Probability Matrix

$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

Evidence / Sensor Model/ Emission Probability Matrix

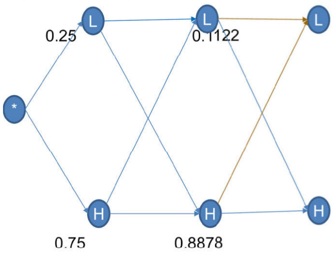
$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

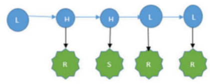
Hidden Morkov Model

Forward Propagation Algorithm : S-S-R

$P(L)*P(L|L)*P(R|L) = 0.1122*0.5*0.8 = \mathbf{0.04488}$

$P(H)*P(L|H)*P(R|L) = 0.8878*0.2*0.8 = \mathbf{0.142048}$





Transition Model / Probability Matrix

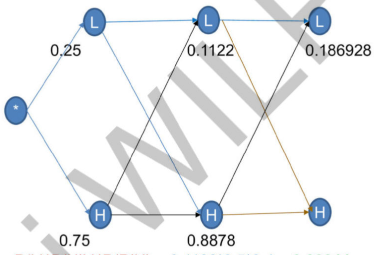
$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

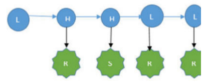
Hidden Morkov Model

Forward Propagation Algorithm : S-S-R



$P(L)*P(H|L)*P(R|H) = 0.1122*0.5*0.4 = \mathbf{0.02244}$

$P(H)*P(H|H)*P(R|H) = 0.8878*0.8*0.4 = \mathbf{0.284096}$



Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

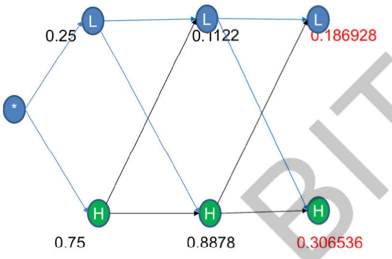
Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

Hidden Morkov Model

Forward Propagation Algorithm : S-S-R

Termination Phase:



Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

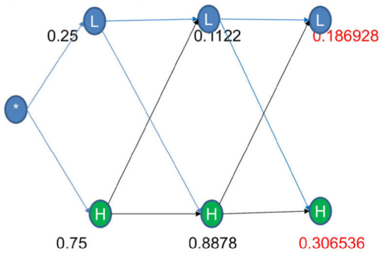
$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

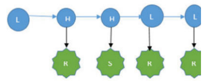
Hidden Morkov Model

Forward Propagation Algorithm : S-S-R

Termination Phase:

(0.37881,**0.62119**)





Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

Hidden Morkov Model

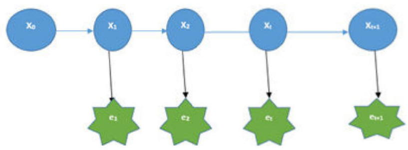
Inference: Type -3

Filtering : Forward Propagation Algorithm

Find the Current Pressure if sequence of weather observations recorded are: **S-S-R**

Intuition: $P(Xt_{+1} | E_{1...t+1}) = \boldsymbol{\alpha}\, P(et_{+1} | Xt_{+1}) * \sum_{X_t} P(X_{t+1}|X_t) * \boldsymbol{P}(\mathbf{X_t} | \mathbf{E_{1..t}})$

$$P(Xt_{+1} | E_{1...t+1}) = \boldsymbol{\alpha}\, P(e_{t+1} | Xt_{+1}) * \sum_{X_t} P(X_{t+1}|X_t) * \boldsymbol{P}(\mathbf{X_t} | \mathbf{E_{1..t}})$$



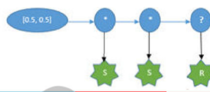
Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

Hidden Morkov Model



-3

Filtering : Forward Propagation Algorithm

Find the Current Pressure if sequence of weather observations recorded are: **S-S-R**

Intuition: $P(Xt_{+1} | E_{1...t+1}) = \boldsymbol{\alpha}\, P(et_{+1} | Xt_{+1}) * \sum_{X_t} P(X_{t+1}|X_t) * \boldsymbol{P}(\mathbf{X_t} | \mathbf{E_{1..t}})$

$P(X_3 | SSR) = P(X_3 | S,S,R)$

$$= \frac{\boldsymbol{P}(\boldsymbol{R} | \mathbf{X_3}\, \boldsymbol{S,S}) * P(X_3 | S,S)}{P(R)}$$

$$= \frac{P(R | X_3) * \boldsymbol{P}(\mathbf{X_3} | \boldsymbol{S,S})}{P(R)}$$

$$= \frac{P(R | X_3) * \{\sum_{X_2} P(X_3|X_2) * \boldsymbol{P}(\mathbf{X_2} | \boldsymbol{S,S})\}}{P(R)}$$

$$= \frac{P(R | X_3) * \{\sum_{X_2} P(X_3|X_2) * P(R | X_3) * \{\sum_{X_1} P(X_2|X_1) * \boldsymbol{P}(\mathbf{X_1} | \boldsymbol{S})\}\}}{P(R) * P(S)}$$

Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

$$P(Xt_{+1} | E_{1...t+1}) = \boldsymbol{\alpha}\, P(e_{t+1} | Xt_{+1}) * \sum_{X_t} P(X_{t+1}|X_t) * \boldsymbol{P}(\mathbf{X_t} | \mathbf{E_{1..t}})$$

Evidence / Sensor Model/ Emission Probability Mat

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

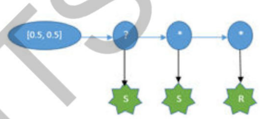
Hidden Morkov Model

Inference: Type -4

Smoothing : Backward Propagation Algorithm (Most Likely State Estimation)

Find the Pressure in past instance of time if sequence of following future weather observations recorded are: **S-S-R**

Intuition: $P(E_{1...t}) = \sum_{i=1}^{N} P(E_{1...t} | X_{1...t}) * P(X_{1...t}) = \sum_{i=1}^{N} \prod_{j=1}^{t} P(Ej | Xj) * P(Xj | Xj_{-1})$



Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

Hidden Morkov Model

Inference: Type -4

Smoothing : Backward Propagation Algorithm

Find the Pressure in past instance of time if sequence of following future weather observations recorded are: **S-S-R**

Intuition: $P(Xt_{+1} | E_{1...t+1}) = \boldsymbol{\alpha}\, P(et_{+1} | Xt_{+1}) * \sum_{X_t} P(X_{t+1}|X_t) * \boldsymbol{P}(\mathbf{X_t} | \mathbf{E_{1..t}})$

$P(X_1 | SSR) = P(X_1 | S,S,R)$

$$= \frac{\boldsymbol{P}(\boldsymbol{SR} | \mathbf{X_1}\, \boldsymbol{S}) * P(X_1 | S)}{P(SR)}$$

$$= \frac{P(X_1 | S) * \{\sum_{X_2} P(X_2|X_1) * \boldsymbol{P}(\mathbf{SR} | \mathbf{X_2}\, \mathbf{X_1})\}}{P(SR)}$$

$$= \frac{P(X_1 | S) * \{\sum_{X_2} P(X_2|X_1) * \boldsymbol{P}(\mathbf{SR} | \mathbf{X_2})\}}{P(SR)}$$

$$= \frac{P(X_1 | S) * \{\sum_{X_2} P(X_2|X_1) * P(S | X_2) * \boldsymbol{P}(\mathbf{R} | \mathbf{X_2})\}}{P(SR)}$$

$$= \frac{P(X_1 | S) * \{\sum_{X_2} P(X_2|X_1) * P(S | X_2) * \{\sum_{X_3} P(X_3|X_2) * P(R | X_3) * \boldsymbol{P}(. | \mathbf{X_3})\}\}}{P(SR)}$$

Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previou
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP$

Evidence / Sensor Model/ Emission Pro

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserve Evidence v
0.8	0.4	$P(E_t = Rainy)$
0.2	0.6	$P(E_t = Sunny)$

$$P(Xt | E_{t+1,\, t+2\, ...\, z}) = \boldsymbol{\alpha} * \text{fwd msg} * \sum_{X_{t+1}} P(X_{t+1}|X_t) * P(e_{t+1}|X_{t+1}) * \boldsymbol{P}(\mathbf{E_{t+2..z}} | \mathbf{X_{t+1}})$$

Hidden Morkov Model

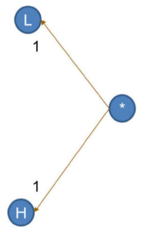
Backward Propagation Algorithm

Pressure sequence observation: **S-S-R**

Initialization Phase: Set value 1 for the terminal state

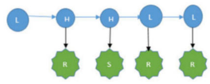
P(L|L)*P(R|L)*P(.|L) = 0.5*0.8 * 1= 0.40

P(H|L)*P(R|H)*P(.|H) = 0.5*0.4 *1= 0.2



P(L|H)*P(R|L)*P(.|L) = 0.2*0.8 * 1= 0.16

P(H|H)*P(R|H)*P(.|H) = 0.8*0.4 *1= 0.32



Transition Model / Probability Matrix

$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

Evidence / Sensor Model/ Emission Probability Matrix

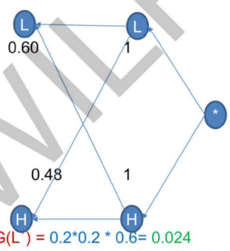
$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

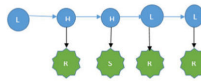
Hidden Morkov Model

Backward Propagation Algorithm : S-S-R

P(L|L)*P(S|L)*MSG(L`) = 0.5*0.2 * 0.60= 0.06

P(H|L)*P(S|H)*MSG(H`) = 0.5*0.6*0.48= 0.144





P(L|H)*P(S|L)*MSG(L`) = 0.2*0.2 * 0.6= 0.024

P(H|H)*P(S|H)*MSG(H`) = 0.8*0.6 *0.48= 0.2304

Recursion Phase:

Transition Model / Probability Matrix

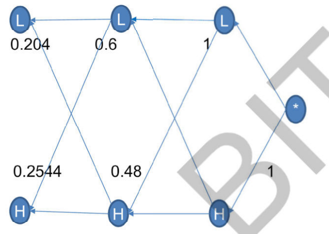
$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

Hidden Morkov Model

Backward Propagation Algorithm : S-S-R



Recursion Phase: If it continues if needed !!!!

Transition Model / Probability Matrix

$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

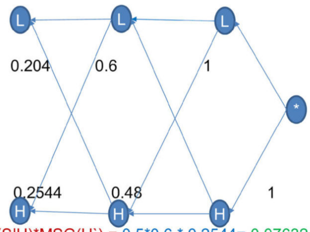
Evidence / Sensor Model/ Emission Probability Matrix

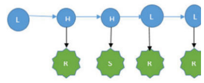
$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

Hidden Morkov Model

Backward Propagation Algorithm : S-S-R

P(L)*P(S|L)*MSG(L`) = 0.5*0.2 * 0.204= 0.0204





P(H)*P(S|H)*MSG(H`) = 0.5*0.6 * 0.2544= 0.07632

Termination Phase: (0.2109,0.7891)

Normalize :Initial value * Emission at start* backMsg

Transition Model / Probability Matrix

$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

Evidence / Sensor Model/ Emission Probability Matrix

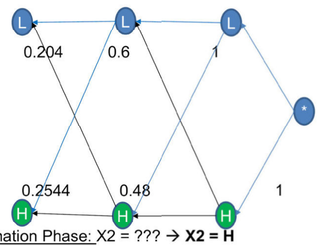
$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

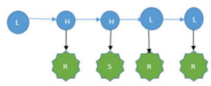
Hidden Morkov Model

Forward Backward Propagation Algorithm : S-S-R

P(X2 | SSR) = α * P(X2|SS) *P(R|X2)

P(X2 | SSR) = α * (0.1122 , 0.8878)* (0.6, 0.48) = (0.06732, 0.426144) = (0.14,0.86)





Termination Phase: X2 = ??? → **X2 = H**

Transition Model / Probability Matrix

$P(U_{t-1}=HP)$	$P(U_{t-1}=LP)$	←Previous
0.2	0.5	$P(U_t=LP)$
0.8	0.5	$P(U_t=HP)$

Evidence / Sensor Model/ Emission Probability Matrix

$P(X_t=LP)$	$P(X_t=HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t=Rainy)$
0.2	0.6	$P(E_t=Sunny)$

Forward Path Probability

$\alpha_t(j) = \sum_i \alpha_{t-1}(i)\, a_{ij}\, b_j(o_t)$

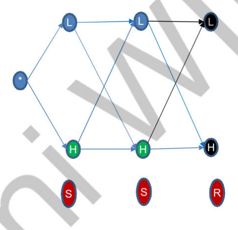
$P(O_{1..t} \mid \lambda)$

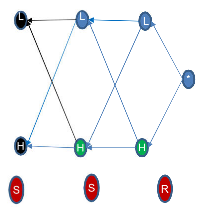
Backward Path Probability

$\beta_t(i) = \sum_j \beta_{t+1}(j)\, a_{i,j}\, b_j(o_{t+1})$

$P(O_{t+1,\, t+2..t+k} \mid \lambda)$

$\gamma_t(i) = P(X_t \mid O_{1\ldots t\ldots t+1,\, t+2..t+k} \mid \lambda)$: Forward – Backward Algorithm






```
function FORWARD-BACKWARD(ev, prior) returns a vector of probability distributions
  inputs: ev, a vector of evidence values for steps 1, . . . , t
          prior, the prior distribution on the initial state, P(X0)
  local variables: fv, a vector of forward messages for steps 0, . . . , t
                   b, a representation of the backward message, initially all 1s
                   sv, a vector of smoothed estimates for steps 1, . . . , t

  fv[0] ← prior
  for i = 1 to t do
     fv[i] ← FORWARD(fv[i − 1], ev[i])
  for i = t downto 1 do
     sv[i] ← NORMALIZE(fv[i] × b)
     b ← BACKWARD(b, ev[i])
  return sv
```

Figure 15.4 The forward–backward algorithm for smoothing: computing posterior probabilities of a sequence of states given a sequence of observations. The FORWARD and BACKWARD operators are defined by Equations (15.5) and (15.9), respectively.

Text & Natural Language Processing

HMM Application

Text & Natural Language Processing

HMM Application

Initial	Prob
Noun	
aDverb	
Verb	
adJective	
Preposition	
Determiner	

N	D	V	J	P	
					N
					D
					V
					J
					A
					P
					E

- Boys are taller.
  N V J
- This is the tree.
  D V D N
- She is a tall girl.
  N V D J N
- Trees are more.
  N V D
- Girls are more than boys
  N V D P N
- The tall tree is falling.
  D J N V V

Given the corpus with tags to build training data:

1. Create initial probability matrix.
2. Transition probability matrix
3. Emission probability matrix
4. Use HMM Veterbi algorithm to predict the sequence of PoS Tags for given test data / sentence.

In the HMM model , the PoS tags act as the hidden states and the word in the given test sentence as the observed states.

Initial	Prob
N	0.67
D	0.33
V	0
J	0
A	0
P	0

N	D	V	J	A	P	
	01675		0.67		1	N
		0.571				D
0.63	0.1675	0.143				V
	0.33	0.143				J
						A
	0.1675					P
0.37	0.1675	0.143	0.33			E

N	D	V	J	A	P	
0.25						Boy s
		0.43				Are
			1			Tall
	0.17					This
		0.43				Is
	0.33					The
0.375						Tree
0.125						She
	0.17					A
0.25						Girl
	0.33					Mor e
					1	Tha n
		0.14				fall

- Boys are taller.
  N V J
- This is the tree.
  D V D N
- She is a tall girl.
  N V D J N
- Trees are more.
  N V D
- Girls are more than boys.
  N V D P N
- The tall tree is falling.
  D J N V V

Initial	Prob
N	0.67
D	0.33
V	0
J	0
A	0
P	0

N	D	V	J	A	P	
	01675		0.67		1	N
		0.571				D
0.63	0.1675	0.143				V
	0.33	0.143				J
						A
	0.1675					P
0.37	0.1675	0.143	0.33			E

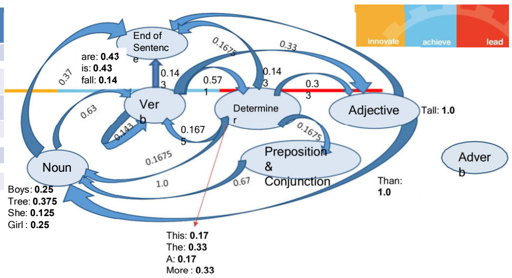
Exercise :

For the below test data/sentence, using the tables constructed using training data, predict the PoS tags.

"Girls are falling"

N	D	V	J	A	P	
0.25						Boy s
		0.43				Are
			1			Tall
	0.17					This
		0.43				Is
	0.33					The
0.375						Tree
0.125						She
	0.17					A
0.25						Girl
	0.33					Mor e
					1	Tha n
		0.14				fall

Initi	
N	0.67
D	0.33
V	0
J	0
A	0
P	0



"Girls are falling"

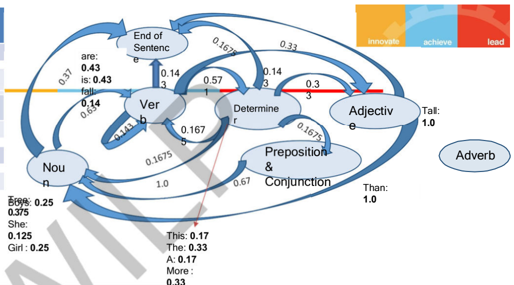
P(Girls , Noun) = P(Girl | Noun) * P(Noun | StartState) = 0.25 * 0.67 = **0.1675**

P(Girls , Verb) = P(Girl | Verb) * P(Verb | StartState) = 0 * 0 = 0 (Ideally with better corpus and the KB , for
most cases it might not be 0 but too low like 0.00000000000…..001.)

P(Girls , Determiner) = P(Girls , Adverb) = P(Girls , Adjective) =P(Girls , Preposition/Conjunction) = 0

StartState → Noun

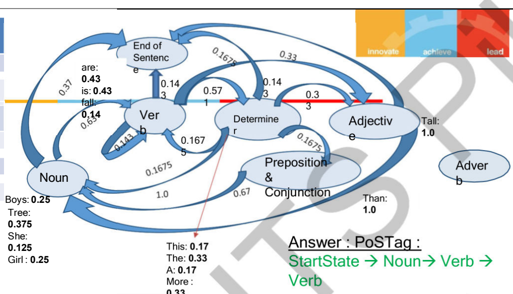
Initi	
N	0.67
D	0.33
V	0
J	0
A	0
P	0



"Girls are falling"

If Sequence is StartState → Noun	If Sequence is StartState → Verb
P(are , Verb) = P(are| Verb) * P(Verb | Noun) * **P(Girls | Noun) * P(Noun | StartState)** = 0.43 * 0.63 * **0.1675** **= 0.04537** P(are, Noun) = P(are| Noun) * P(Noun | Noun) * **Noun** = 0 * 0 = 0 P(are, Determiner) = P(are, Adverb) = P(are , Adjective) =P(are , Preposition/Conjunction) = 0 StartState→ Noun→ Verb	P(are , Verb) = P(are| Verb) * P(Verb | Verb) * **P(Girls | Verb)** * **P(Verb | StartState)** = 0.43 * 0.143 * **0 = 0** P(are, Noun) = P(are| Noun) * P(Noun | Noun) * **Verb**= 0 * 0 = 0 P(are, Determiner) = P(are, Adverb) = P(are , Adjective) =P(are , Preposition/Conjunction) = 0

Initi	
N	0.67
D	0.33
V	0
J	0
A	0
P	0



"Girls are falling"

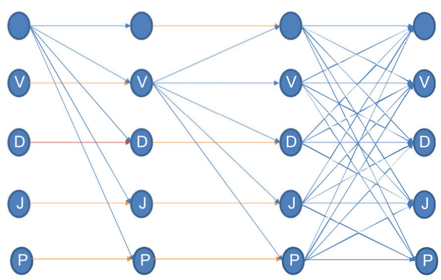
Answer : PoSTag :
StartState → Noun→ Verb → Verb

If Sequence is StartState→ Noun →Verb	If Sequence is StartState→ Verb ->Adjective
P(falling, Verb) = P(falling|Verb)*P(Verb|Verb)***P(are| Verb) * P(Verb | Noun)** * **P(Girls | Noun)** * **P(Noun | StartState)** = 0.14 * 0.143 * **0.04537** **= 0.000908** P(are, Noun) = P(are| Noun) * P(Noun | Noun) * **Noun** = 0 * 0 = 0 P(are, Determiner) = P(are, Adverb) = P(are , Adjective) =P(are , Preposition/Conjunction) = 0	P(falling , Verb) = P(falling | Verb) * P(Verb | Verb) * P(are| Adjective) * P(Adjective | Verb) * P(Girls | Verb) * P(Verb | StartState) = 0.14 * 0.143 * **0 = 0** P(falling, Noun) = P(falling| Noun) * P(Noun | Adjective) * **Verb**= 0 * 0 = 0 P(falling, Determiner) = P(falling, Adverb) = P(falling , Adjective) =P(falling , Preposition/Conjunction) = 0

Sample Sequence under Test: Start → Noun → Verb

- … … … .

Assume Noun→ Verb is the maximum Value



Learning HMM Parameters

Parameter Estimation by EM Algorithm

(Baum-Welch re-estimation procedure)

Parameter Estimation

Learning Approach

Baum-Welch re-estimation procedure: Backward Propagation Algorithm

Given an observation sequence O(Evidence) and the set of possible states in the HMM, learn the HMM parameters A(Transition) and B(Emission).

Given set of weather observations recorded estimate the PARAMATERS:

{SS, SR, RR}

	HH	HL	LH	LL
SS	**(0.5).(0.6).(0.8)(0.6) = 0.1440**	0.0120	0.03	0.01
SR	**0.0960**	0.048	0.02	0.04
RR	0.064	0.032	0.12	**0.16**
Total	0.304	0.092	0.17	0.21

Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Prob

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = \text{Rainy})$
0.2	0.6	$P(E_t = \text{Sunny})$

Parameter Estimation

Learning Approach

Baum-Welch re-estimation procedure: Backward Propagation Algorithm

Given an observation sequence O(Evidence) and the set of possible states in the HMM, learn the HMM parameters A(Transition) and B(Emission).

Given set of weather observations recorded estimate the PARAMATERS:

{SS, SR, RR}

HP	LP	
0.232323232	0.5526316	LP
0.767676768	0.4473684	HP

	HH	HL	LH	LL	Best Seq	P(Best)
SS	**0.1440**	0.0120	0.03	0.01	HH	0.144
SR	**0.0960**	0.048	0.02	0.04	HH	0.096
RR	0.064	0.032	0.12	**0.16**	LL	0.16
Total	0.304	0.092	0.17	0.21		0.4
Normalize	0.76	0.23	0.425	0.525		

Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probabi

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = \text{Rainy})$
0.2	0.6	$P(E_t = \text{Sunny})$

Parameter Estimation

Learning Approach

Baum-Welch re-estimation procedure: Backward Propagation Algorithm

Given an observation sequence O(Evidence) and the set of possible states in the HMM, learn the HMM parameters A(Transition) and B(Emission).

Find set of weather observations recorded estimate the parAMATERS:

{SS, SR, RR}

LP	HP	
0.615384615	0.4	R
0.384615385	0.6	S

	H→S	L→S	H→R	L→R	Best Seq	P(Seq)
SS	**0.1440**	0.01			HH	0.144
SR	0.0960	0.04	0.096	0.048	HH	0.096
RR			0.064	0.0320	LL	0.16
Total	0.24	0.05	0.16	0.08		
Normalize	0.6	0.125	0.4	0.2		

Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previou
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Prob

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = \text{Rainy})$
0.2	0.6	$P(E_t = \text{Sunny})$

Parameter Estimation

Learning Approach

Baum-Welch re-estimation procedure: Backward Propagation Algorithm

Given an observation sequence O(Evidence) and the set of possible states in the HMM, learn the HMM parameters A(Transition) and B(Emission).

Find set of weather observations recorded estimate the parAMATERS:

{SS, SR, RR}

After this step for the second iteration
Use the optimized tables
(Initial, Transition , Emission)
and repeat the algorithm till convergence

HP	LP
0.56929	0.4307

	Start(H)	Start(L)	Best Seq	P(Best)
SS	**0.1440**	0.03	HH	0.144
SR	0.0960	0.04	HH	0.096
RR	0.064	0.16	LL	0.16
	0.304	0.23		
Normalize	0.76	0.575		

Transition Model / Probability Matrix

$P(U_{t-1} = HP)$	$P(U_{t-1} = LP)$	←Previous
0.2	0.5	$P(U_t = LP)$
0.8	0.5	$P(U_t = HP)$

Evidence / Sensor Model/ Emission Probability

$P(X_t = LP)$	$P(X_t = HP)$	←Unobserved Evidence v
0.8	0.4	$P(E_t = \text{Rainy})$
0.2	0.6	$P(E_t = \text{Sunny})$

HMM in Prevention of Network Security Threat

(Interesting Case Studies)

Hidden Morkov Model

Cyber Security

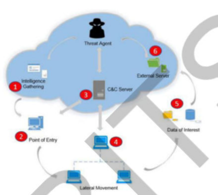
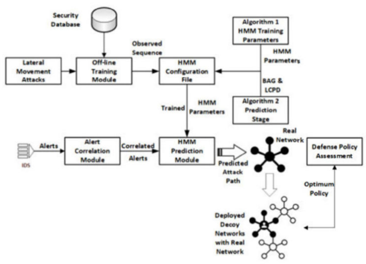


FIGURE 1. Typical stages of APT attack.

Source Credit : [2021 : Hidden Markov Model and Cyber Deception for the Prevention of Adversarial Lateral Movement]

Hidden Morkov Model

Cyber Security



Source Credit : [2021 : Hidden Markov Model and Cyber Deception for the Prevention of Adversarial Lateral Movement]

Hidden Morkov Model

Cyber Security

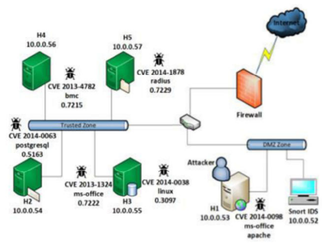


FIGURE 9. Experimental network topology.

Source Credit : [2021 : Hidden Markov Model and Cyber Deception for the Prevention of Adversarial Lateral Movement]

Hidden Morkov Model

Cyber Security

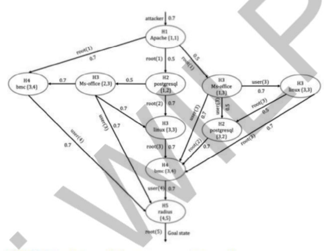


FIGURE 10. Attack graph of the experimental network.

Attack states description.

State	Description
S_1	Initial State
S_2	$(H_1$,root)
S_3	$(H_2$,root)
S_4	$(H_3$,user)
S_5	$(H_3$,root)
S_6	$(H_4$,user)
S_7	$(H_5$,root)

Source Credit : [2021 : Hidden Markov Model and Cyber Deception for the Prevention of Adversarial Lateral Movement]

Hidden Morkov Model

Cyber Security

Attack states description.

TABLE 6. Possible attack paths.

Path Number	Attack Path
1	$S_1 \to S_2 \to S_6 \to S_7$
2	$S_1 \to S_2 \to S_3 \to S_4 \to S_7$
3	$S_1 \to S_2 \to S_3 \to S_6 \to S_7$
4	$S_1 \to S_2 \to S_3 \to S_4 \to S_5 \to S_6 \to S_7$
5	$S_1 \to S_2 \to S_3 \to S_5 \to S_6 \to S_7$
6	$S_1 \to S_2 \to S_4 \to S_6 \to S_7$
7	$S_1 \to S_2 \to S_4 \to S_3 \to S_6 \to S_7$
8	$S_1 \to S_2 \to S_4 \to S_5 \to S_3 \to S_6 \to S_7$
9	$S_1 \to S_2 \to S_4 \to S_5 \to S_6 \to S_7$

State	Description
S_1	Initial State
S_2	$(H_1$,root)
S_3	$(H_2$,root)
S_4	$(H_3$,user)
S_5	$(H_3$,root)
S_6	$(H_4$,user)
S_7	$(H_5$,root)

Source Credit : [2021 : Hidden Markov Model and Cyber Deception for the Prevention of Adversarial Lateral Movement]

Required Reading: AIMA - Chapter #15.1, #15.2, #15.3, #20.3.3

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials

Artificial & Computational Intelligence

AIMLCZG557

Contributors & Designers of document content : Cluster Course Faculty Team

M7 : Ethics in AI

BITS Pilani
Pilani Campus

Presented by
Faculty Name
BITS Email ID

Artificial and Computational Intelligence

Disclaimer and Acknowledgement

Disclaimer

- Few content for these slides may have been obtained from prescribed books and various other source on the Internet
- I hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge people others who made their course materials freely available online.
- .I have provided source information wherever necessary
- This is not a full fledged reading materials. Students are requested to refer to the textbook w.r.t detailed content of the presentation deck that is expected to be shared over e-learning portal - taxilla.
- I have added and modified the content to suit the requirements of the class dynamics & live session's lecture delivery flow for presentation

- **Slide Source / Preparation / Review:**
- From BITS Pilani WILP: Prof.Raja vadhana, Prof. Indumathi, Prof.Sangeetha
- From BITS Oncampus & External : Mr.Santosh GSK

Course Plan

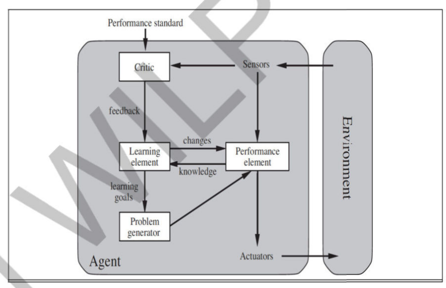
- M1 Introduction to AI
- M2 Problem Solving Agent using Search
- M3 Game Playing
- M4 Knowledge Representation using Logics
- M5 Probabilistic Representation and Reasoning
- M6 Reasoning over time
- M7 Ethics in AI

Learning Objective

At the end of this class , students Should be able to:

1. Understand the explainabilty/interpretability of Intelligent systems
2. Relate the use of logics in the explainability of complex systems
3. Understand the connect between the ethical impact and design of Intelligent agents

Shortcomings of AI



Recommendation System


Insight - Amazon scraps secret AI recruiting tool that showed bias against women

Amazon's Edinburgh engineering hub's goal was to develop AI that could rapidly crawl the web and spot candidates worth recruiting

Fairness : The absence of bias towards an individual or a group

Are the predictions _____ ?
- Fair
- Unbiased

Object Recognition System


Google Photos Tags Two African-Americans As Gorillas Through Facial Recognition Software

-

Are the Inferences _____ ?
- Correct
- Unbiased

Are the Predictions _____?
- Fair
- Universally Applicable

Natural Language Processing system


Microsoft silences its new A.I. bot Tay, after Twitter users teach it racism [Updated]

Tay is Microsoft's conversational bot powered by NLP & ML.

Are the interpretations _____ ?
- Fair
- Legal
- Socially Ethical

Building a Fair Model

No artificial model is a perfect one. But every model significantly influence the social , economic, cultural ethics impacting humanity.

Justify the design modelled & metric used to validate the model, is in fact the right choices fit in the context.

1. Is it fair to make an AI-ML system?
2. Is there a better technical approach to convert an existing AI system fair?
3. Are the results obtained by the AI system fair?

Building a Fair Model

1. Is it fair to make an AI-ML system?
2. Is there a better technical approach to convert an existing AI system fair?

Interpretable Results | Statistical Independence | Customized Metric

Interpretable Models

Are the results obtained by the AI system fair?

Interpretable models helps to trust the AI system by answering transparently to the specific questions like "Why the system is behaving under certain scenarios?"

- If a loan gets rejected, do we know the reasons?
- If a job application is accepted, is it biased towards a gender?
- If a bail is granted to an accused, is it based on their race?
- If a patient is diagnosed with a disease, what factors made the algorithm to classify it?

Interpretable Models

Example Based Explanations:
If SymptomInX ≡ SymptomInY
if DiseaseA infected X
then probably DiseaseB might have infected Y

If CustomerX ≡ CustomerY
if CustomerX purchased P1
then probably CustomerY will purchase P1

Counterfactual Explanations:
If customerX's income level had not been less than L3
then the customer's Loan might not have been rejected

Autonomous AI

Challenges

Environment & Agent:
- Knowledge availability
- Background knowledge

Learning Element:
- Autonomy
- Dynamics of the Environment

XAI

- Bayesian Networks

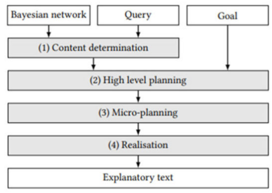
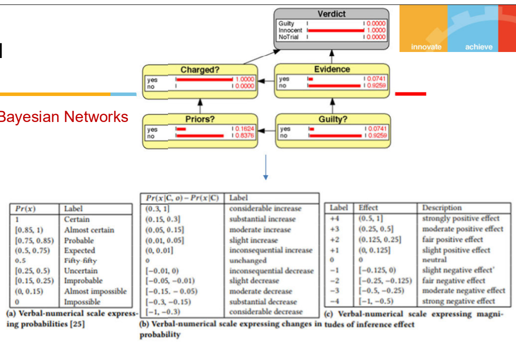


Figure 4: Pipeline architecture of the explanation system

Source Credit :
June 2019: Explainable Bayesian Network Query Results via Natural Language Generation Systems
International Conference on Artificial Intelligence and Law
https://doi.org/10.1145/3322640.3326716

XAI

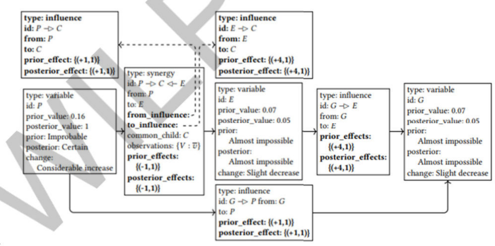
- Bayesian Networks



Source Credit :
June 2019: Explainable Bayesian Network Query Results via Natural Language Generation Systems
International Conference on Artificial Intelligence and Law
https://doi.org/10.1145/3322640.3326716

XAI

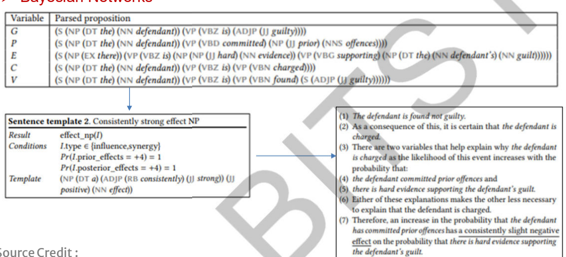
- Bayesian Networks



Source Credit :
June 2019: Explainable Bayesian Network Query Results via Natural Language Generation Systems
International Conference on Artificial Intelligence and Law
https://doi.org/10.1145/3322640.3326716

XAI

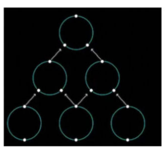
- Bayesian Networks

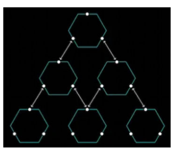


Source Credit :
June 2019: Explainable Bayesian Network Query Results via Natural Language Generation Systems
International Conference on Artificial Intelligence and Law
https://doi.org/10.1145/3322640.3326716

XAI

- Logic based Neural Network (LNN) in KBQA

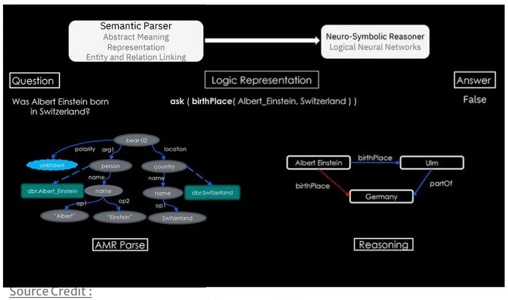




Source Credit :
2020: https://research.ibm.com/blog/ai-neurosymbolic-common-sense

XAI

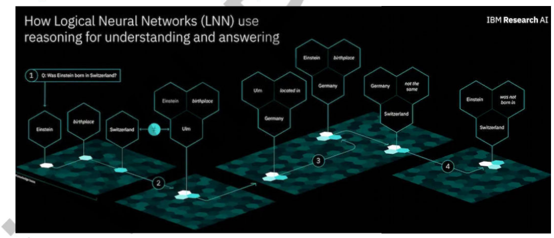
➢ Logic based Neural Network (LNN) in KBQA



Source Credit :
2020: https://research.ibm.com/blog/ai-neurosymbolic-common-sense

XAI

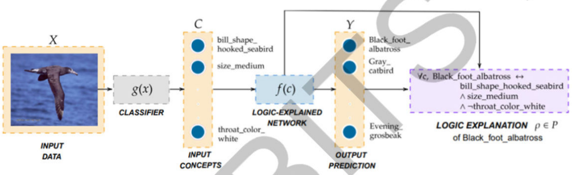
➢ Logic based Neural Network (LNN) in KBQA



Source Credit :
2020: https://research.ibm.com/blog/ai-neurosymbolic-common-sense

XAI

➢ Logic Explained Network in Deep Learning



Source Credit: [2021: Logic Explained Networks]

Required Reading: AIMA - Chapter #15.1, #15.2, #15.3, #20.3.3

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials