GeeksforGeeks

GATE फर्रे

DATA SCIENCE & ARTIFICIAL INTELLIGENCE (DA)

AI

SHORT NOTES

ENROLL NOW

TO EXCEL IN GATE
AND ACHIEVE YOUR DREAM IIT OR PSU!

ENROLL NOW

STAR MENTOR CS/DA

KHALEEL SIR
ALGORITHM & OS
29 YEARS OF TEACHING EXPERIENCE

CHANDAN SIR
DIGITAL LOGIC
GATE AIR 23 & 26 / EX-ISRO

SATISH SIR
DISCRETE MATHEMATICS
BE in IT from MUMBAI UNIVERSITY

MALLESHAM SIR
M.TECH FROM IIT BOMBAY
AIR – 114, 119, 210 in GATE
(CRACKED GATE 8 TIMES)
14+ YEARS EXPERIENCE

VIJAY SIR
DBMS & COA
M. TECH FROM NIT
14+ YEARS EXPERIENCE

PARTH SIR
DA
IIIT BANGALORE ALUMNUS
FORMER ASSISTANT PROFESSOR

SAKSHI MA'AM
ENGINEERING MATHEMATICS
IIT ROORKEE ALUMNUS

SHAILENDER SIR
C PROGRAMMING & DATA STRUCTURE
M.TECH in Computer Science
15+ YEARS EXPERIENCE

AVINASH SIR
APTITUDE
10+ YEARS OF TEACHING EXPERIENCE

AJAY SIR
PH.D. IN COMPUTER SCIENCE
12+ YEARS EXPERIENCE

Artificial Intelligence – GATE DA Short Notes

1. Search in AI

Uninformed (Blind) Search

- **Overview:** Uses only problem definition, no domain-specific knowledge; explores state space systematically.

Algorithm	Expansion Order	Complete	Optimal	Space Complexity	Time Complexity	Example Use Case
Breadth-First Search	Level by level	Yes	Yes*	O(b^d)	O(b^d)	Sudoku with small solution depth
Depth-First Search	Deepest node first	No	No	O(bd)	O(b^d)	Maze with long corridors
Uniform Cost Search	Lowest path cost next	Yes	Yes	O(b^d)	O(b^d)	Route planning with road distances

*Optimal if step costs are uniform.

Example:

BFS in a simple graph—Find shortest path from A to D.

Graph: A–B, A–C, B–D, C–D

Expansion order: A → B, C → D

Shortest path: A–B–D or A–C–D.

Informed (Heuristic) Search

- **Overview:** Uses a heuristic function h(n) to estimate cost to goal, speeds up search, can ensure optimality with admissible heuristics.

Algorithm	Evaluation Function	Complete	Optimal	Heuristic Use	Example
Greedy Best-First Search	f(n) = h(n)	No	No	Picks node closest to goal	Solving mazes via Euclidean distance
A* Search	f(n) = g(n) + h(n)	Yes	Yes*	Balances cost and heuristic	Google Maps shortest path with time cost

*Optimal if h(n) is admissible and consistent.

Example: *8-puzzle A**

- g(n) : steps taken so far
- h(n) : number of misplaced tiles
- f(n) = g(n) + h(n) selects most promising configuration.

Adversarial Search

- **Overview:** Applies to multi-agent, competitive environments (e.g., games).

*Assumes perfect knowledge and optimal play by both players.

Algorithm	Applicability	Optimal Play	Main Idea	Example
Minimax	2+ player	Yes*	Maximize own minimum gain	Chess
Alpha-Beta Prune	2+ player	Yes	Prune nodes that can't affect move	Chess (faster search)

Example:

Tic-Tac-Toe Minimax Tree: Explores all possible moves, chooses move that ensures win/draw with best strategy from both players.

2. Logic in AI

a. Propositional Logic

Features:

- **Definition:** Simple logic with statements (propositions) that are true/false.
- **Syntax:** AND (∧), OR (∨), NOT (¬), IMPLIES (→), IFF (↔).
- **Example:**
  - Let P: "It is raining", Q: "The ground is wet."
  - Statement: P→Q

- **Inference:**
  - **Modus Ponens:** If P→Q and P are true, infer Q.
  - **Resolution:** Combining clauses to derive conclusions (used in SAT solvers).

Feature	Purpose	Example
Atomic formula	Basic proposition	P : "It's sunny"
Conjunction	Both are true	P ^ Q
Disjunction	At least one is true	P ∨ Q
Negation	Not true	¬ P
Implication	If-then	P →Q

Example Inference:
Given P →Q , P. Then Q .

b. Predicate Logic (First-Order Logic, FOL)

- **Definition:** Extends propositional logic to reason about objects and their relationships.
- **Syntax:** Predicates (e.g., Likes(Alice,IceCream) $Likes(Alice, IceCream)$), Quantifiers (∀, ∃), variables.
- **Examples:**
- "Every student likes AI": ∀x [Student(x)→Likes(x,AI)]$\forall x$
- "Some student dislikes mathematics": ∃x [Student(x)∧¬Likes(x,Math)]$\exists x$
- **Inference:**
- **Unification:** Matching predicates to make logical deductions.
- **Resolution:** Inference method for first-order logic (FOL).

Features:

- Objects, relations, and quantifiers
- More expressive than propositional logic

Element	Syntax Example	Meaning
Predicate	Student(x)	"x is a student"
Function	Father(x)	"Father of x"
Quantifiers	∀,∃	"For all", "There exists"
Relation	Likes(x, IceCream)	"x likes ice cream"

Statement Type	Formalization
Universal (all)	$\forall x\ Student(x) \rightarrow Likes(x, AI)$
Existential (some)	$\exists x\ Student(x) \wedge \neg Likes(x, Math)$

Examples:
"All humans are mortal": $\forall x\,(Human(x) \rightarrow Mortal(x))$
"Some cat is black": $\exists x\,(Cat(x) \wedge Black(x)$

Logic Table: Quantifiers and Formulas

English Statement	Quantifier Logic Formulation
Everyone loves someone	$\forall x \exists y\ Loves(x,y)$
There is someone everyone loves	$\exists y \forall x\ Loves(x,y)$
Nothing loves everything	$\forall x \neg \forall y\ L(x,y)$
Something loves nothing	$\exists x \forall y \neg L(x,y)$
Only one student took Greek	$\exists x\,[Student(x) \wedge \ldots] \wedge \forall y\,(y \square = x \rightarrow \ldots)$

3. Reasoning Under Uncertainty

a. Conditional Independence & Probabilistic Representation

- **Bayesian Networks:** Directed acyclic graphs where nodes = variables; arcs = dependencies.
- **Conditional Independence Example:**
- In a network: Rain→WetGrass←Sprinkler
- Rain is independent of Sprinkler given WetGrass.
- **Probabilistic Factorization:**
- P(X,Y,Z)=P(X)·P(Y|X)·P(Z|Y)

Exact Inference (Variable Elimination)

- **Goal:** Compute P(Q|e) (probability of query Q given evidence e).
- **Steps:**

1. Write joint distribution in terms of network's factors.
2. Identify hidden variables (not in query/evidence).
3. Sum out hidden variables in correct order.

- **Example:**
- To find probability it rains, given grass is wet, sum over possible states of "Sprinkler".

	Explanation	Example
Bayesian Network	Graph: Nodes = random variables, Arcs = dependencies	Weather (Rain, Sprinkler, WetGrass)
Conditional Independence	$P(A\|B,C)=P(A\|B)$ $P(A\|B,C)=P(A\|B)$ if A ⊥ C given B	$P(Rain\|Sprinkler,WetGrass)=P(Rain\|WetGrass)$ if Rain ⊥ Sprinkler given WetGrass
Factorization	Decompose joint probability	P(A,B,C)=P(A) P(B|A) P(C|B)

b. Approximate Inference (Sampling)

- **Why:** Exact inference is often infeasible for large/loopy networks.
- **Key Methods:**
- **Rejection Sampling:** Generates samples and rejects non-matching evidence.
- **Likelihood Weighting:** Samples variables, weighting by likelihood of evidence.
- **Gibbs Sampling:** Iteratively resamples each variable with others fixed.
- **Example:**
- For a large disease network, estimate probability of "disease" given symptoms using Gibbs Sampling.

Example:

Bayes Net structure:
Rain → WetGrass ← Sprinkler
Conditional independence: Given WetGrass, Rain and Sprinkler are not independent; given Sprinkler, Rain and WetGrass are independent.

c. Exact Inference (Variable Elimination)

- **Purpose:** Compute exact probability of queries in BayesNets.
- **Goal:** Compute P(Q|e) $P(Q|e)$ (probability of query Q Q given evidence e e).
- **Steps:**

Write joint distribution in terms of network's factors.
Identify hidden variables (not in query/evidence).
Sum out hidden variables in correct order.

- **Example:**

To find probability it rains, given grass is wet, sum over possible states of "Sprinkler".

Step	Explanation	Example Calculation
1. Write Joint	Express probability using conditional factors	P(Rain,Sprinkler,WetGrass)= P(Rain)P(Sprinkler)P(WetGrass|Rain, Sprinkler)
2. Sum Out	Marginalize hidden variables	$P(Rain,WetGrass)=\Sigma_{Sprinkler}P(Rain,Sprinkler,WetGrass)$
3. Normalize	Divide by evidence probability to obtain conditionals	P(Rain|WetGrass)=P(Rain,WetGrass) P(WetGrass)

Worked Example:

Calculate P(Rain|Wet Grass) :

- List all probability factors
- Sum over non-evidence variables
- Normalize.

d. Approximate Inference (Sampling)

- Used when exact inference is computationally infeasible.
- **Sampling methods:** Generate samples to estimate required probabilities.
- **Rejection Sampling**
- **Gibbs Sampling**
- **Likelihood Weighting**
- **Trade-off:** Faster, scalable, but involves statistical approximation of the solution

- **Purpose:** Estimate probabilities when exact inference is too costly.

Method	Core Idea	Suitability	Example Situation
Rejection Sampling	Generate full samples, reject those not matching evidence	Simple, inefficient if rare evidence	Diagnostic BayesNets with common findings
Likelihood Weighting	Weight samples by likelihood of evidence	More efficient	Medical diagnosis with known symptoms
Gibbs Sampling	Iteratively sample each variable, given the rest	Efficient for complex nets	Large disease-symptom networks

Example:

Have 3 binary variables, want $P(A=1|C=1)$. Use Gibbs:

- Start with a random assignment
- Repeatedly sample A, B, and C, each given the values of the other two
- After many iterations, estimate probability by frequency.

4. Summary Table – AI Syllabus Coverage

Topic	Key Points	Typical Algorithms/Examples
Uninformed Search	No problem-specific guidance	BFS, DFS, Uniform Cost Search
Informed Search	Uses heuristic to guide	Greedy Search, A*
Adversarial Search	Decision-making under competition	Minimax, Alpha-Beta Pruning
Propositional Logic	Simple true/false statements	Truth tables, Resolution
Predicate Logic	Quantified, relational reasoning	Unification, Quantifiers
Conditional Independence	Efficient probabilistic modeling	Bayesian Networks
Exact Inference	Marginals/conditionals via elimination	Variable Elimination Algorithm
Approximate Inference	Probability estimation by sampling	Rejection, Gibbs, Likelihood Weighting

GeeksforGeeks

GATE CSE BATCH

KEY HIGHLIGHTS:

- 300+ HOURS OF RECORDED CONTENT
- 900+ HOURS OF LIVE CONTENT
- SKILL ASSESSMENT CONTESTS
- 6 MONTHS OF 24/7 ONE-ON-ONE AI DOUBT ASSISTANCE
- SUPPORTING NOTES/DOCUMENTATION AND DPPS FOR EVERY LECTURE

COURSE COVERAGE:

- ENGINEERING MATHEMATICS
- GENERAL APTITUDE
- DISCRETE MATHEMATICS
- DIGITAL LOGIC
- COMPUTER ORGANIZATION AND ARCHITECTURE
- C PROGRAMMING
- DATA STRUCTURES
- ALGORITHMS
- THEORY OF COMPUTATION
- COMPILER DESIGN
- OPERATING SYSTEM
- DATABASE MANAGEMENT SYSTEM
- COMPUTER NETWORKS

LEARNING BENEFIT:

- GUIDANCE FROM EXPERT MENTORS
- COMPREHENSIVE GATE SYLLABUS COVERAGE
- EXCLUSIVE ACCESS TO E-STUDY MATERIALS
- ONLINE DOUBT-SOLVING WITH AI
- QUIZZES, DPPS AND PREVIOUS YEAR QUESTIONS SOLUTIONS

ENROLL NOW

TO EXCEL IN GATE
AND ACHIEVE YOUR DREAM IIT OR PSU!

ENROLL NOW