TDT4136 Introduction to Artificial Intelligence

Lecture 13: Summary

Pinar Öztürk

Norwegian University of Science and Technology
2021

Today's plan

- summarization of each lecture in one page
- a couple of example questions - solving them
- practical things/info about the exam.

Syllabus: As specified in the Blackboard, Chapters in the textbook + Lecture slides

Lectures during the semester

1. Introduction.
2. Intelligent agents
3. Search and Problem solving, Uninformed Search
4. Informed Search + Local Search
5. Adversarial Search
6. Constraint Satisfaction Problems
7. Propositional Logic
8. First order logic
9. Inference in FOL
10. Planning
11. Knowledge Representation + Ethical issues in AI
12. Multiagent environments and Game Theory

Lectures 1 - Introduction

- History of AI. The Darthmout Summer School organised by John McCarthy coined "Artificial Intelligence". Learn the other few important people as well.
- Turing test and Chineese room of Searle. The latter, about consciousness/understanding of machines, has become very hot topic these days again)
- Nature of Different types of AI. The field started with Good Old Fashion AI, and SEAI evolved in later years.
- Knowledge (and its representation is the backbone of GOFAI while SEAI relies on perceived information/data and bodily (hence embodied) action.

Lectures 1 - Introduction

- History of AI. The Darthmout Summer School organised by John McCarthy coined "Artificial Intelligence". Learn the other few important people as well.
- Turing test and Chineese room of Searle. The latter, about consciousness/understanding of machines, has become very hot topic these days again)
- Nature of Different types of AI. The field started with Good Old Fashion AI, and SEAI evolved in later years.
- Knowledge (and its representation is the backbone of GOFAI while SEAI relies on perceived information/data and bodily (hence embodied) action.
- Russel & Norving investigates different agents along thinking vs acting, and rational vs human-like - the matrix.
- Many disciplines contributed to/"interacts with" AI. Sometimes AI helps them to understand/investigate their research questions through e.g., simulations, other times AI is inspired from their models/methods.
- Many application areas for AI, and many killer applications.

- Notion of agent and rationality. Agents defined through PEAS - Performance, Environment, Actuators, Sensors
- Different types of environments and their characteristics.
  - partial vs full observable, deterministic vs stochastic (and strategic environments with multiple agents), discrete vs continuous, episodic vs sequential, static vs dynamic, known vs unknown.
  - is prisoner's dilemma environment episodic or sequential? Depends on whether one-shot (played only once) or repeated.
  - is Solitaire partially or fully observable? Find in the textbook.

Lecture 2 - Intelligent Agents

- Notion of agent and rationality. Agents defined through PEAS - Performance, Environment, Actuators, Sensors
- Different types of environments and their characteristics.
  - partial vs full observable, deterministic vs stochastic (and strategic environments with multiple agents), discrete vs continuous, episodic vs sequential, static vs dynamic, known vs unknown.
  - is prisoner's dilemma environment episodic or sequential? Depends on whether one-shot (played only once) or repeated.
  - is Solitaire partially or fully observable? Find in the textbook.
- Different types of agents(programs). Rationality of agents depends on the combination of the environment and the task they are given
  - simple/pure reflex, model-based reflex, goal-based, utility-based, learning agents
- From the book: "What is rational at any given time depends on 4 things: The performance measure....., The agent's prior knowledge of the environment,......".
  - What kind of knowledge is this? How can it be represented?
    - atomic, states are not elaborated, indivisible state with no internal structure (e.g., in search, chaps 3, 4)
    - factored, (CSP, propositional logic, planning) properties of the states represented as attribute-value pairs.
    - structured representations, with relationships between objects

Lecture 3 - Problem solving through search

- Represent a problem as a search problem. Definition of a problem: initial state, actions, transition model (i.e., successor fn), goal test. Path cost (i.e, cost function capturing performance measure).
- Uninformed search algorithms, Breath-first, depth-first, uniform-cost, iterative deepening, and bidirectional search are the most known examples.
- Tree versus graph search. Time and memory complexity, completeness and optimality of algorithms.
- Important characteristics of an algorithm: Memory and time complexity, completeness, and optimality.
- Memory and time complexity of algorithms differ - expressed in terms of branching factor and depth of the shallowest goal.

Lecture 3 - Problem solving through search

- Represent a problem as a search problem. Definition of a problem: initial state, actions, transition model (i.e., successor fn), goal test. Path cost (i.e, cost function capturing performance measure).
- Uninformed search algorithms, Breath-first, depth-first, uniform-cost, iterative deepening, and bidirectional search are the most known examples.
- Tree versus graph search. Time and memory complexity, completeness and optimality of algorithms.
- Important characteristics of an algorithm: Memory and time complexity, completeness, and optimality.
- Memory and time complexity of algorithms differ - expressed in terms of branching factor and depth of the shallowest goal.
- Goal testing time differs across algorithms, e.g., breath first search does goal test early, upon generation of children i.e, before putting into the frontier not when popping out of it.
- Main difference between search you saw in Algorithm and Data structures course and in this course

  - the search tree/graph is not given explicitly. It evolves as problem solving process proceeds.

  - not necessary to traverse the entire tree, often stops when a goal found, unless explicitely asked to find all goals.

Lecture 4 - Informed search and local search

- Informed search using domain-specific heuristics
- Requirements on heuristic function/values.
  - Admissible heuristic in In A* tree search, while consistency required in graph-search. What is the main point with consistency. What kind of overhead if inconsistent heuristic is used in graph search?

Lecture 4 - Informed search and local search

- Informed search using domain-specific heuristics
- Requirements on heuristic function/values.
  - Admissible heuristic in In A* tree search, while consistency required in graph-search. What is the main point with consistency. What kind of overhead if inconsistent heuristic is used in graph search?
- Local Search - Simulating annealing, Hill-climbing, genetic algorithms...
  - Unknown environments
  - State representation: starts with a possibly incorrect but complete-state solution - to be iteratively modified towards a better one
  - Hill climbing - Moves to a next state/neighbour that is best with respect to either objective function or heuristic cost function - to be maximized and minimized respectively.
    - incomplete - i.e., stuck in local max/min.
    - stochastic hill climbing with random choice of next move, random restart.
  - Simulating annealing. choses the next move randomly instead of the best move. if this move is not better than the current it accepts it with a probability that decreases exponentially with $\Delta E=$ (next value - current value), and with temperature T that decreases by time.

Lecture 4 - Informed search and local search

- Informed search using domain-specific heuristics
- Requirements on heuristic function/values.
  - Admissible heuristic in In A* tree search, while consistency required in graph-search. What is the main point with consistency. What kind of overhead if inconsistent heuristic is used in graph search?
- Local Search - Simulating annealing, Hill-climbing, genetic algorithms...
  - Unknown environments
  - State representation: starts with a possibly incorrect but complete-state solution - to be iteratively modified towards a better one
  - Hill climbing - Moves to a next state/neighbour that is best with respect to either objective function or heuristic cost function - to be maximized and minimized respectively.
    - incomplete - i.e., stuck in local max/min.
    - stochastic hill climbing with random choice of next move, random restart.
  - Simulating annealing. choses the next move randomly instead of the best move. if this move is not better than the current it accepts it with a probability that decreases exponentially with $\Delta E =$ (next value - current value), and with temperature T that decreases by time.
- Search with nondeterministic actions(e.g., erratic VC), and Searching in Partially observable E. $\Rightarrow$ contingency plans. AND-OR search trees on Belief state space
- Sensorless/conformant problems plans with sequence of actions.

Lecture 5 - Adversarial search

- Search when more than one agent take action in the environment. 2-agent play focus in the textbook but it can be more than two agents.
- Sequential, deterministic, fully observable environment
- Solution:a strategy specifying a move for every possible opponent reply
- Minimax search algorithm - terminal test and utility function
- Alpha-Beta alg for pruning/limiting search tree. effectiveness dependent on the order in which the states are examined. Searches up to the terminal states.

- Search when more than one agent take action in the environment. 2-agent play focus in the textbook but it can be more than two agents.
- Sequential, deterministic, fully observable environment
- Solution:a strategy specifying a move for every possible opponent reply
- Minimax search algorithm - terminal test and utility function
- Alpha-Beta alg for pruning/limiting search tree. effectiveness dependent on the order in which the states are examined. Searches up to the terminal states.
- Approaches to the efficiency problem:
  - use cutoff-test instead of terminal-test: e.g., depth limit
  - use a heuristics evaluation function instead of utility function that estimates desirability of position
  - properties of a good evaluation function:
    - it should order terminal states the same way as the true utility function
    - for non-terminal states, the evaluation function should be strongly correlated with the true chance of winning
    - computation of values must not take too long

Lecture 6- Constraint Satisfaction Problem

- CSP is a special kind of problem: - states defined by values of a fixed set of variables- factored representation. Goal test defined by constraints on variable values
- Reasoning mechanism: search, inference or combination of them
- Inference mechanism: various types of constraint propagation, eg., node-, arc-,path- and k-consistency . Unary, binary and global constraints.

Lecture 6- Constraint Satisfaction Problem

- CSP is a special kind of problem: - states defined by values of a fixed set of variables- factored representation. Goal test defined by constraints on variable values
- Reasoning mechanism: search, inference or combination of them
- Inference mechanism: various types of constraint propagation, eg., node-, arc-,path- and k-consistency . Unary, binary and global constraints.
- Inference:
  - Forward checking (typically used with search) does 1-step arc consistency and checks the consistency of the current variable with its immediate neighbours
  - AC-3: propagates arc consistency until either no more arcs in the queue or detects there is no solution (when the size of a domain becomes zero). May find out that there is no solution, or finds the solution of all domains have size 1. Or, serves as preprocessing for search: reduces the CSP so that search is more effective.

Lecture 6- Constraint Satisfaction Problem

- CSP is a special kind of problem: - states defined by values of a fixed set of variables- factored representation. Goal test defined by constraints on variable values
- Reasoning mechanism: search, inference or combination of them
- Inference mechanism: various types of constraint propagation, eg., node-, arc-,path- and k-consistency . Unary, binary and global constraints.
- Inference:
  - Forward checking (typically used with search) does 1-step arc consistency and checks the consistency of the current variable with its immediate neighbours
  - AC-3: propagates arc consistency until either no more arcs in the queue or detects there is no solution (when the size of a domain becomes zero). May find out that there is no solution, or finds the solution of all domains have size 1. Or, serves as preprocessing for search: reduces the CSP so that search is more effective.
- Heuristics for efficiency: Variable ordering (e.g., Minimum remaining values, and degree heuristics) and value selection (e.g., Least-constraining value) heuristics - these are domain-independent heuristics

Lecture 6- Constraint Satisfaction Problem

- CSP is a special kind of problem: - states defined by values of a fixed set of variables- factored representation. Goal test defined by constraints on variable values
- Reasoning mechanism: search, inference or combination of them
- Inference mechanism: various types of constraint propagation, eg., node-, arc-,path- and k-consistency . Unary, binary and global constraints.
- Inference:
  - Forward checking (typically used with search) does 1-step arc consistency and checks the consistency of the current variable with its immediate neighbours
  - AC-3: propagates arc consistency until either no more arcs in the queue or detects there is no solution (when the size of a domain becomes zero). May find out that there is no solution, or finds the solution of all domains have size 1. Or, serves as preprocessing for search: reduces the CSP so that search is more effective.
- Heuristics for efficiency: Variable ordering (e.g., Minimum remaining values, and degree heuristics) and value selection (e.g., Least-constraining value) heuristics - these are domain-independent heuristics
- Backtracking search, a depth-first search with one variable assigned at a time, backtracks when inconsistency. Can integrate inference (e.g., forward checking).
- Local search, where states are complete assignments, with possibly some conflicted variables. Search changes the value of one of the conflicted variables. Uses "minimum conflict" heuristic

Lecture 7 -Propositional Logic

- Basic concepts of logic:
  - syntax: formal structure of sentences
  - semantics: truth of sentences wrt models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
  - soundness: derivations produce only entailed sentences
  - completeness: derivations can produce all entailed sentences
- Precedence of operators/connectives: $\neg$, $\wedge$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.

Lecture 7 -Propositional Logic

- Basic concepts of logic:
  - syntax: formal structure of sentences
  - semantics: truth of sentences wrt models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
  - soundness: derivations produce only entailed sentences
  - completeness: derivations can produce all entailed sentences
- Precedence of operators/connectives: $\neg$, $\wedge$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
- Inference: truth of a sentence given a KB, i.e. whether the KB entails the sentence

  - soundness of inference algorithm: produces for a given KB only the sentences entailed by the KB - important notions e.g., entailment, validity, satisfiability - model checking by truth table

Lecture 7 -Propositional Logic

- Basic concepts of logic:
  - syntax: formal structure of sentences
  - semantics: truth of sentences wrt models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
  - soundness: derivations produce only entailed sentences
  - completeness: derivations can produce all entailed sentences
- Precedence of operators/connectives: $\neg$, $\wedge$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
- Inference: truth of a sentence given a KB, i.e. whether the KB entails the sentence

  - soundness of inference algorithm: produces for a given KB only the sentences entailed by the KB - important notions e.g., entailment, validity, satisfiability - model checking by truth table

  - Forward, backward chaining - uses Horn clauses

  - Resolution refutation, uses CNF sentences,
- Propositional logic lacks expressive power.

- More expressiveness,
  - can express partial information, e.g., "there exist a person in the room" without needing to tell which concrete person/object.
  - objects and relations between them, properties of objects
  - uses quantifications and variables for such expressive.
- Figure 8.3 in the textbook shows what atomic/complex sentence, term, constant, predicate, function, etc means

- More expressiveness,
  - can express partial information, e.g., "there exist a person in the room" without needing to tell which concrete person/object.
  - objects and relations between them, properties of objects
  - uses quantifications and variables for such expressive.
- Figure 8.3 in the textbook shows what atomic/complex sentence, term, constant, predicate, function, etc means
- Typical wrong use of universal and existential quantifications.
  - Correct use (mostly): Universal/all quantifiers dominantly with $\rightarrow$,
  - Existential quant. with $\wedge$.

- More expressiveness,
  - can express partial information, e.g., "there exist a person in the room" without needing to tell which concrete person/object.
  - objects and relations between them, properties of objects
  - uses quantifications and variables for such expressive.
- Figure 8.3 in the textbook shows what atomic/complex sentence, term, constant, predicate, function, etc means
- Typical wrong use of universial and existential quantifications.
  - Correct use (mostly): Universal/all quantifiers dominantly with $\rightarrow$,
  - Existential quant. with $\wedge$.
- Relationship between Universal and Existential Quantifiers
- Nested Quantifiers

Lectures 9 - Inference in First order Logic

- Inference rules for quantifiers
  - Universal Instantiation - substitution of variables with ground terms (i.e, terms without variables)
  - Existential Instantiation - a variable is substituted with a unique(not seen/used before) constant/fn, skolem constant/fn
- The use of Quantifiers and variables require some extra work/syntactical operations (e.g., when converting to CNF (for resolution) and to Definite clausal form (for Backward Chaining) such as unification, skolemization for existential quantifiers, standardization of variable names

- Inference rules for quantifiers
  - Universal Instantiation - substitution of variables with ground terms (i.e, terms without variables)
  - Existential Instantiation - a variable is substituted with a unique(not seen/used before) constant/fn, skolem constant/fn
- The use of Quantifiers and variables require some extra work/syntactical operations (e.g., when converting to CNF (for resolution) and to Definite clausal form (for Backward Chaining) such as unification, skolemization for existential quantifiers, standardization of variable names
- Inference through reduction to propositional inference - propositionalize and use propositional logic inference. Not efficient.
- Inference without propositionalization. Using "generalized" versions of the inference rules in PropLogic,
  - e.g., generalized/lifted modus ponens. A sound inference.
  - involves Substitution. Advantage of inference without propositionalization: makes only substitution required for particular inferences required for solving the problem, not all possible substitutions

Lectures 9 - Inference in First order Logic

- Inference rules for quantifiers
  - Universal Instantiation - substitution of variables with ground terms (i.e, terms without variables)
  - Existential Instantiation - a variable is substituted with a unique(not seen/used before) constant/fn, skolem constant/fn
- The use of Quantifiers and variables require some extra work/syntactical operations (e.g., when converting to CNF (for resolution) and to Definite clausal form (for Backward Chaining) such as unification, skolemization for existential quantifiers, standardization of variable names
- Inference through reduction to propositional inference - propositionalize and use propositional logic inference. Not efficient.
- Inference without propositionalization. Using "generalized" versions of the inference rules in PropLogic,
  - e.g., generalized/lifted modus ponens. A sound inference.
  - involves Substitution. Advantage of inference without propositionalization: makes only substitution required for particular inferences required for solving the problem, not all possible substitutions
- Inference with Backward/Forward chaining
- inference through Resolution Refutation

Lecture 11 - Planning

- Uses factored representations. KR languages specific to planning, eg., PDDL that builds on logical representation.
- Special representation for actions - action schemas. State representations without variables and functions.

Lecture 11 - Planning

- Uses factored representations. KR languages specific to planning, eg., PDDL that builds on logical representation.
- Special representation for actions - action schemas. State representations without variables and functions.
- Planning as state-space search versus plan-space search
- State space search. Each node represents a state of the world. A plan is a path through the space.
  - Forward or backward state-space search

Lecture 11 - Planning

- Uses factored representations. KR languages specific to planning, eg., PDDL that builds on logical representation.
- Special representation for actions - action schemas. State representations without variables and functions.
- Planning as state-space search versus plan-space search
- State space search. Each node represents a state of the world. A plan is a path through the space.
  - Forward or backward state-space search
- Plan-space search. Partial order planning. Refinement of a partially ordered plan.
  - Can produce a plan like this : Action1 ; (Action2 AND Action3) ; Action4; (Action 5 AND Action 6). Meaning that Action1 is followed by (Action 2 and Action3 which can be executed in either order, no ordering between them), followed by Action4, which in turn followed by (Action5 and Action6 which can be executed in either order).
- Plan operators are adding actions, causal links, and ordering links.
- Heuristics for planning: ignore precondition, ignore-delete-list heuristic....

Lecture 10 - KR. AI-ethics

- Chapter12: Knowledge representation. Ontologies, Category and objects.
- Hierarchical organisation of knowledge, inheritance-based inference.
- Structural representation languages: Semantic network, Frames (combines factored and structural)

Lecture 10 - KR. AI-ethics

- Chapter12: Knowledge representation. Ontologies, Category and objects.
- Hierarchical organisation of knowledge, inheritance-based inference.
- Structural representation languages: Semantic network, Frames (combines factored and structural)

- Ethical aspects of AI, Regulations by various institutions including EU
- Common in these are:
  - autonomy (people should be able to make their own decisions, e.g. human-in-the-loop, privacy protection)
  - beneficence (society at large should benefit)
  - non-maleficence (harmful consequences should be avoided, e.g.systems should be robust)
  - justice (diversity, non-discrimination and fairness)
  - explicability (transparency and explainability)

Lecture 10 - KR. AI-ethics

- Chapter12: Knowledge representation. Ontologies, Category and objects.
- Hierarchical organisation of knowledge, inheritance-based inference.
- Structural representation languages: Semantic network, Frames (combines factored and structural)

- Ethical aspects of AI, Regulations by various institutions including EU
- Common in these are:
  - autonomy (people should be able to make their own decisions, e.g. human-in-the-loop, privacy protection)
  - beneficence (society at large should benefit)
  - non-maleficence (harmful consequences should be avoided, e.g.systems should be robust)
  - justice (diversity, non-discrimination and fairness)
  - explicability (transparency and explainability)
- Ethics in Philosophy
  - deontological (principles about actions, duties or rules)
  - virtue (what kind of person, moral characters)
  - consequentialism (goodness of the outcome)

Lecture 12 - Multiagent environments and Game theory

- Utility-based agents. Notions of outcome and utility. Payoff matrix representation of states. One-shot games. Strategical behaviour,
- Rationality: all agents are rational (aims to maximize their expected utility). This important because it help an agent to guess what the opponent(s) will do.

- Utility-based agents. Notions of outcome and utility. Payoff matrix representation of states. One-shot games. Strategical behaviour,
- Rationality: all agents are rational (aims to maximize their expected utility). This important because it help an agent to guess what the opponent(s) will do.
- Notions of Dominance, action profiles, dominant and dominated actions.
- Elimination of dominated actions is rational. Applying these in different cardinal games, such as prisoners dilemma, stag hunt, etc.
- Notion of Nash equilibrium
- Pareto optimality and social welfare and their use for solving problems. When can be used?

Exam

- Home exam with pass/fail grade
- Note that this is home exam but with 4 hours - as the not-home exam. You must study everything thoroughly because you will not have time to read and learn during the exam.
- All written material is allowed during the exam. Videos will not be available.
- The work load of the last years exam was too high. This year it will be close to the original exams.
- Working together or getting the answers from another person is forbidden
- There will not be multiple choice questions