CS 561: Artificial Intelligence

Instructor: Sofus A. Macskassy, macskass@usc.edu
TAs: Nadeesha Ranashinghe (nadeeshr@usc.edu)
William Yeoh (wyeoh@usc.edu)
Harris Chiu (chiciu@usc.edu)

Lectures: MW 5:00-6:20pm, OHE 122 / DEN
Office hours: By appointment
Class page: http://www-rcf.usc.edu/~macskass/CS561-Spring2010/

This class will use http://www.uscden.net/ and class webpage

- Up to date information
- Lecture notes
- Relevant dates, links, etc.

Course material:
[AIMA] Artificial Intelligence: A Modern Approach,
by Stuart Russell and Peter Norvig. (2nd ed)

Review

- Intro
- Intelligent agents
- Problem solving and search
- Adversarial game search
- Constraint satisfaction problems
- Logical agents
- First-order logic
- Knowledge representation
- ~~Logical reasoning~~
- Planning
- Uncertainty
- Probabilistic reasoning and inference
- Probabilistic reasoning over time
- Rational decision-making
- Learning
- Communication and language

Intro

- Turing test
- AI Research
  - Theoretical and experimental
  - Two lines
    - Biological – based on human analogy psychology/physiology
    - Phenomonal – formalizing common-sense
- We have studied theoretical - phenomonal

Intelligent agents

- **Intelligent agents**
  - Anything that can be *viewed as* **perceiving** its **environment** through **sensors** and **acting** upon that environment through its **actuators** to maximize progress towards its **goals**
  - **PAGE** (Percepts, Actions, Goals, Environment)
  - The environment types largely determine the agent design
  - Described as a Perception (sequence) to Action Mapping:

    $$f : P^* \rightarrow A$$
  - Using look-up-table, closed form, etc.

- **Agent Types:** Reflex, state-based, goal-based, utility-based

- **Rational Action:** The action that maximizes the expected value of the performance measure <u>given the percept sequence to date</u>

Problem solving and search - uninformed

- Uninformed
  - Breadth-first, Uniform-cost, Depth-first, Depth-limited, Iterative deepening
- Problem formulation usually requires abstracting away real-world details to define a state space that can be explored using computer algorithms.
  - **Single-state problem:** deterministic, accessible
  - **Multiple-state problem:** deterministic, inaccessible
  - **Contingency problem:** nondeterministic, inaccessible
  - **Exploration problem:** unknown state space
- Once problem is formulated in abstract form, complexity analysis helps us picking out best algorithm to solve problem.
- Variety of uninformed search strategies; difference lies in method used to pick node that will be further expanded.
- Iterative deepening search only uses linear space and not much more time than other uniformed search strategies.
- Graph search can be exponentially more efficient than tree search.

Problem solving and search - heuristic

- Heuristic
  - Best first, A*, Hill-climbing, Simulated annealing

- Time complexity of heuristic algorithms depend on quality of heuristic function. Good heuristics can sometimes be constructed by examining the problem definition or by generalizing from experience with the problem class.

- Iterative improvement algorithms keep only a single state in memory.

- Can get stuck in local extrema; simulated annealing provides a way to escape local extrema, and is complete and optimal given a slow enough cooling schedule.

Adversarial game search

- **Game playing**
  - Perfect play
    - The minimax algorithm, alpha-beta pruning
  - Elements of chance
  - Imperfect information
- **Complexity:** many games have a huge search space
  - **Chess:** *$b = 35, m=100 \Rightarrow nodes = 35^{100}$*
    if each node takes about 1 ns to explore
    then each move will take about ***10^{50}* millennia**
    to calculate.
- **Resource (e.g., time, memory) limit:** optimal solution not feasible/possible, thus must approximate

1. **Pruning:** makes the search more efficient by discarding portions of the search tree that cannot improve quality result.
2. **Evaluation functions:** heuristics to evaluate utility of a state without exhaustive search

Constraint satisfaction problems

- CSPs are a special kind of problem:
  - states defined by values of a fixed set of variables
  - goal test defined by constraints on variable values
- Backtracking = depth-first search with one variable assigned per node
- Variable ordering and value selection heuristics help significantly
- Forward checking prevents assignments that guarantee later failure
- Constraint propagation (e.g., arc consistency) does additional work
- to constrain values and detect inconsistencies
- The CSP representation allows analysis of problem structure
- Tree-structured CSPs can be solved in linear time
- Iterative min-conflicts is usually effective in practice

Logics in general

Language	Ontological Commitment	Epistemological Commitment
Propositional logic	**facts**	**true/false/unknown**
First-order logic	**facts, objects, relations**	**true/false/unknown**
Temporal logic	**facts, objects, relations, times**	**true/false/unknown**
Probability logic	**facts**	**degree of belief 0…1**
Fuzzy logic	**facts, degree of truth**	**known interval value**

Logical agents – propositional logic

- Logical agents apply inference to a knowledge base
  to derive new information and make decisions
- 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
- Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.
- Forward, backward chaining are linear-time, complete for Horn clauses
- Resolution is complete for propositional logic
- Propositional logic lacks expressive power

First-order logic

- First-order logic:
  - objects and relations are semantic primitives
  - syntax: constants, functions, predicates, equality, quantifiers
- Increased expressive power: sufficient to define wumpus world
- Quantification – universal and existential
- ~~Situation calculus~~

Inference in first-order logic

- Reducing first-order inference to propositional inference
- Unification
- Generalized Modus Ponens
- Forward and backward chaining
- Logic programming
- Resolution

Knowledge representation

- Knowledge engineering: principles and pitfalls
- Ontologies
- Examples

Planning

- Search vs. planning
- STRIPS operators
- Partial-order planning
- Types of planners
  - Situation space planner: search through possible situations
  - Progression planner: start with initial state, apply operators until goal is reached
  - Regression planner: start from goal state and apply operators until start state reached
  - Partial order planner: some steps are ordered, some are not
  - Total order planner: all steps ordered (thus, plan is a simple list of steps)
- Simple planning agent
  - Use percepts to build model of current world state
  - IDEAL-PLANNER: Given a goal, algorithm generates plan of action
  - STATE-DESCRIPTION: given percept, return initial state description in format required by planner
  - MAKE-GOAL-QUERY: used to ask KB what next goal should be

Uncertainty

- Probability is a rigorous formalism for uncertain knowledge
- Joint probability distribution specifies probability of every atomic event
- Queries can be answered by summing over atomic events
- For nontrivial domains, we must find a way to reduce the joint size
- Independence and conditional independence provide the tools

Probabilistic reasoning

- Syntax and Semantics
- Parameterized distributions
- Bayes nets provide a natural representation for (causally induced) conditional independence
- Topology + CPTs = compact representation of joint distribution
- Canonical distributions (e.g., noisy-OR) = compact representation of CPTs
- Continuous variables $\Rightarrow$ parameterized distributions (e.g., linear Gaussian)

Probabilistic inference

- Exact inference by variable elimination
  - polytime on polytrees, NP-hard on general graphs
  - space = time, very sensitive to topology
- Approximate inference by LW, MCMC:
  - LW does poorly when there is lots of (downstream) evidence
  - LW, MCMC generally insensitive to topology
  - Convergence can be very slow with probabilities close to 1 or 0
  - Can handle arbitrary combinations of discrete and continuous variables

Probabilistic reasoning over time

- Temporal models use state & sensor variables replicated over time
- Markov assumptions and stationarity assumption, so we need
  - transition model $\mathbf{P}(\mathbf{X}_t | \mathbf{X}_{t-1})$
  - sensor model $\mathbf{P}(\mathbf{E}_t | \mathbf{X}_t)$
- Tasks are filtering, prediction, smoothing, most likely sequence;

  all done recursively with constant cost per time step
- Hidden Markov models have a single discrete state variable
- Dynamic Bayes nets subsume HMMs, ~~Kalman filters;~~ exact update intractable
- Particle filtering is a good approximate filtering algorithm for DBNs

Rational decision-making

- Rational preferences
- Utilities
- Money
- Multi-attribute utilities
- Decision networks
- Value of information

Learning

- Learning needed for unknown environments, lazy designers
- Learning agent = performance element + learning element
- Learning method depends on type of performance element, available
- feedback, type of component to be improved, and its representation
- For supervised learning, the aim is to nd a simple hypothesis
- that is approximately consistent with training examples
- Decision tree learning using information gain
- Learning performance = prediction accuracy measured on test set

Statistical Learning

- Bayes learning
  - Full Bayesian learning gives best possible predictions but is intractable
  - MAP learning balances complexity with accuracy on training data
  - Maximum likelihood assumes uniform prior, OK for large data sets
    - Choose a parameterized family of models to describe the data
    - Search for model parameters which best fit the data

- Neural nets
  - Most brains have lots of neurons; each neuron linear-threshold unit
  - Perceptrons (one-layer networks) insufficiently expressive
  - Multi-layer networks are sufficiently expressive; can be trained by gradient descent, i.e., error back-propagation
  - Many applications: speech, driving, handwriting, fraud detection, etc.
  - Engineering, cognitive modelling, and neural system modelling subfields have largely diverged

Communication and language

- Communication
- Grammar
- Syntactic analysis
- Problems