CS 1571 Introduction to AI
Review

Course review

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Administrative

- **Homeworks**
  - HW 1-10 graded (ready for pickup)
    - see the TA – Tomas Singliar
  - Solutions available on-line
- **Remaining office hours:**
  - Will be posted on the course web page

Final

- **Final:**
  - **Wednesday, December 11, 2002**
  - **5502 Sennot Square, at 2:00-3:50pm**

- **Form of the final exam:**
  - Closed book
  - Cumulative (material for all semester)
  - Lecture notes + corresponding chapters in the textbook

- **Practice exam**

CS 1573: Artificial Intelligence Application Development (Spring 2003)

- **Time Tu Th 2:30-3:45**
- **Place 5129 Sennott Square**
- **Prof: Dr. Diane Litman**
- **http://www.cs.pitt.edu/~litman**

OBJECTIVES

- **Ability to build simple versions of a few AI applications**
- **Familiarity with full-scale versions of the same applications**
- **Mastery of Python, a popular programming language useful for AI applications and rapid prototyping**

DESCRIPTION

This course will focus on the development of artificial intelligence applications. It will cover symbolic data structures, advanced control structures, and advanced prototyping and data exploration techniques. Multiple areas of artificial intelligence will be covered, with a focus on areas (e.g. natural language processing, reinforcement learning) and applications (e.g. expert systems, email filtering, the semantic web) that were not covered during CS 1571.

PREREQUISITES: CS 1571, or consent of the instructor

Review

Search

- **Basic definition of the search problem**
  – Search space, operators, initial state, goal condition
- **Path vs. configuration search**
- **Search tree**
- **Search methods properties :**
  – Completeness, Optimality, Space and time complexity.
- **Complexities**
  – measured in terms of a branching factor (b), depth of the optimal solution (d), maximum depth of the state space (m)

Search

- **Uninformed methods:**
  - Breadth first search, Depth first search, Iterative deepening, Bi-directional search, Uniform cost search (for the weighted path search)
- **Informed methods:**
  - **Heuristic function** (*h*): potential of a state to reach the goal
  - **Evaluation function (*f*) :** desirability of a state to be expanded next
  - **Best first search:**
    - Greedy $f(n) = h(n)$
    - A*: $f(n) = g(n) + h(n)$

      the role of admissible heuristics, optimality

Search

- **Constraint satisfaction problem (CSP)**
  - Variables, constraints on values (reflect the goal)
  - Formulation of a CSP as search
  - Methods and heuristics for CSP search
    - Backtracking, most constrained variable, least constrained value
- **Iterative improvement algorithms:**
  - **Methods:** Hill climbing, Simulated annealing, Genetic algorithms
  - **Advantage**: memory !!

Search

- **Adversarial search (game playing)**
  - Specifics of a game search, game problem formulation
  - rational opponent
- **Algorithms:**
  - Minimax algorithm
    - Complexity bottleneck for large games
  - Alpha-Beta pruning: prunes branches not affecting the decision of players
  - Cutoff of the search tree and heuristics

KR and logic

- **Knowledge representation:**
  - **Syntax** (how sentences are build), **Semantics** (meaning of sentences), **Computational aspect** (how sentences are manipulated)
- **Logic:**
  - A formal language for expressing knowledge and ways of reasoning
  - **Three components:**
    - A set of sentences
    - A set of interpretations
    - The valuation (meaning) function

Propositional logic

- A language for symbolic reasoning
- **Language:**
  - Syntax, Semantics
- **Satisfiability** of a sentence: at least one interpretation under which the sentence can evaluate to ***True***.
- **Validity** of a sentence: ***True*** in all interpretations
- **Entailment:** $KB \models \alpha$

  α is true in all worlds in which KB is true
- **Inference procedure**
  - Soundness If $KB \vdash_i \alpha$ then $KB \models \alpha$
  - Completeness If $KB \models \alpha$ then $KB \vdash_i \alpha$

Propositional logic

- **Logical inference problem:** $KB \models \alpha$?
  - Does KB entail the sentence α ?
- Logical inference problem for the propositional logic is **decidable**.
  - A procedure (program) that stops in finite time exists
- **Approaches:**
  - Truth table approach
  - Inference rule approach
  - Resolution refutation

$KB \models \alpha$ if and only if

$(KB \wedge \neg \alpha)$ is **unsatisfiable**

- **Normal forms**: DNF, CNF, Horn NF (conversions)

First order logic

- Deficiencies of propositional logic
- **First order logic (FOL):** allows us to represent objects, their properties, relations and statements about them
  - Variables, predicates, functions, quantifiers
  - Syntax and semantics of the sentences in FOL
- **Logical inference problem** $KB \models \alpha$?
  - **Undecidable.** No procedure that can decide the entailment for all possible input sentences in a finite number of steps.
- **Inference approaches:**
  - Inference rules
  - Resolution refutation

First order logic

- **Methods for making inferences work with variables:**
  - **Variable substitutions**
  - **Unification** process that takes two similar sentences and computes the substitution that makes that makes them look the same, if it exists
- **Conversions to CNF** with universally quantified variables
  - Used by resolution refutation
    - The procedure is refutation- complete

Knowledge-based systems with HNF

- **KBs in Horn normal form:**
  - Not all sentences in FOL can be translated to HNF
  - Modus ponens is complete for Horn databases
- **Inferences** with KBs in Horn normal form (HNF)
  - Forward chaining
  - Backward chaining
- **Retrieval of information for KBs**
- **Production systems**
  - Conflict resolution

Planning

- **Find a sequence of actions** that lead to a goal
  - Much like path search, but for a very large domains
  - Need to represent the world and change
- **Two basic approaches** planning problem representation:
  - **Situation calculus**
    - Explicitly represents situations (extends FOL)
    - **Solving:** theorem proving
  - **STRIPS**
    - Add and delete list
    - **Solving:** Search

      (Goal progression, Goal regression)
- **Frame problem**

Planning

- **Divide and conquer approach** (Sussman's anomaly)
- **State space vs. plan space search**
- Partial order (non-linear) planners:
  - Search the space of partially build plans
  - Progressive or regressive mode
- **Hierarchical planners**

Uncertainty

- **Basics of probability:**
  - random variable, values, probability distribution
- **Joint probability distribution**
  - Over variables in a set, **full joint** over all variables
  - Marginalization (summing out)
- **Conditional probability distribution**

$$P(A \mid B) = \frac{P(A,B)}{P(B)} \quad \text{s.t. } P(B) \neq 0$$

- **Product rule** $P(A,B) = P(A \mid B)P(B)$
- **Chain rule**
- **Bayes rule**

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Uncertainty

Full joint probability distribution

– Over variables in a set, **full joint** over all variables

Two important things to remember:

- Any probabilistic query can be computed from the full joint distribution
- Full joint distribution can be expressed as a product of conditionals via the chain rule

Bayesian belief networks

- **Full joint distribution** over all random variables defining the domain can be very large
  – Complexity of a model, inferences, acquisition
- **Solution:** Bayesian belief networks (BBNs)
- **Two components of BBNs:**
  – Structure (directed acyclic graph)
  – Parameters (conditional prob. distributions)
- **BBN** build upon conditional independence relations:

$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$

- **Joint probability distribution for BBNs**:
  – Product of local (variable-parents) conditionals

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

Bayesian belief networks

- **Model of joint distribution:**
  - Reduction in the number of parameters
- **Inferences:**
  - Queries on joint probabilities
  - Queries on conditionals expressed as ratios of joint probabilities
  - Joint probabilities can be expressed in terms of full joints
  - Full joints are product of local conditionals
- **Smart way to do inferences:**
  - Interleave sums and products (variable elimination)

Decision-making in the presence of uncertainty

- **Decision tree:**
  - Decision nodes (choices are made)
  - Chance nodes (reflect stochastic outcome)
  - Outcomes (value) nodes (value of the end-situation)
- **Rational choice:**
  - Decision-maker tries to optimize the expected value
- **Utility theory:**
  - Utilities express preferences in terms of numeric quantities
  - Monetary values may not be equal to utility values
  - Choices based on expected utility (and its maximization)
- **Value of information**
  - Additional information (via test) can reduce the expected utility

Learning

- **Three types of learning:**
  - Supervised, unsupervised, reinforcement
- **Main learning steps:**
  - Select model, select error function (measures the misfit between the model and data), optimize error function
- **Overfitting**
- **Supervised learning:**
  - Regression, classification
- **Unsupervised learning:**
  - Density estimation

Learning

- **Linear regression** $f(\mathbf{x}) = w_0 + \sum_{j=1}^{d} w_j x_j$
  - **Error:** mean square error $J_n = \frac{1}{n} \sum_{i=1,..n} (y_i - f(\mathbf{x}_i))^2$
  - **Optimization:**
    - a set of linear equations, Gradient descent
- **Logistic regression**: $f(\mathbf{x}) = p(y=1 \mid \mathbf{x}, \mathbf{w}) = g(w_0 + \sum_{j=1}^{d} w_j x_j)$
  - **Error:** negative log likelihood of data
  - **Optimization:** gradient descent
- **Online version** of the gradient descent:
  - Update rule, learning rate, algorithm

$$w_j \leftarrow w_j + \alpha(y - f(\mathbf{x}))x_j$$

Learning

- **Linear models**
  - Linear regression model: Linear functions
  - Logistic regression: Linear boundaries for binary classification
- **Adding nonlinearities:**
  - Feature functions
  - Multilayer neural networks
    - Online gradient descent: backpropagation