Learning Goals of CS486 Artificial Intelligence

Alice Gao

December 9, 2018

Contents

1 Definitions of AI

- Describe one application of AI.
- Describe each of the four definitions of AI in one sentence.
- Compare and contrast any two definitions of AI. Describe their similarities and differences.

2 Search

- Describe the components of a search problem.
- Formulate a real world problem as a search problem, using the incremental or the complete-state formulation.
- For a successor function, enumerate all the successor states of a given state.
- Given a search problem, estimate the complexity of the search space in terms of number of nodes and paths in the search tree.
- Describe what it means to expand a node and what is a frontier in a search tree.
- Define/trace/implement search algorithms (with/without cost) (dealing with cycles)
- Determine properties of search algorithms: completeness, optimality, time and space complexity.
- Compare and contrast the properties of different search algorithms.
- Given a scenario, explain why it is appropriate or not appropriate to use a particular search algorithm.
- Explain what it means for A* to be optimally efficient.
- Explain what it means for a heuristic function to be admissible. Verify whether a heuristic is admissible.
- Choose the best one among several heuristic functions for a particular problem.
- Construct admissible heuristics for appropriate problems.

3 Constraint Satisfaction Problems

Introduction to CSP and Arc Consistency

- Describe components of a constraint satisfaction problem.
- Formulate a real-world problem as a constraint satisfaction problem.
- Formulate a constraint using the table or the formula format.
- Verify whether a variable is arc-consistent with respect to another variable for a constraint.
- Define/implement/trace the arc consistency algorithm. Describe the possible outcomes of the arc-consistency algorithm.
- Analyze the complexity of the arc consisteny algorithm.

Backtracking search

- Contrast naive depth-first search and backtracking search on a CSP.
- Describe/trace/implement the backtracking search algorithm.
- Describe/trace/implement the backtracking search algorithm with forward checking and/or arc consistency.
- Describe/trace/implement the backtracking search algorithm with forward checking and/or arc consistency and with heuristics for choosing variables and values.

Local search

- Describe the advantages of local search over other search algorithms.
- Formulate a real world problem as a local search problem using complete-state formulations.
- Given a local search problem, verify whether a state is a local optimum.
- Describe/trace/implement the local search algorithms (hill climbing, hill climbing with random restarts, simulated annealing, and genetic algorithms).
- Describe strategies for escaping local optima.
- Compare and contrast the properties of local search algorithms.

4 Reasoning under Uncertainty

A review of probabilities

- Calculate prior, posterior, and joint probabilities using the sum rule, the product rule, the chain rule and Bayes' rule.

Unconditional and Conditional Independence

- Given a description of a domain or a probabilistic model for the domain, determine whether two variables are unconditionally independent.
- Given a description of a domain or a probabilistic model for the domain, determine whether two variables are conditionally independent given a third variable.

Introduction to Bayesian Networks

- Describe reasons for using a Bayesian Network rather than the joint probability distribution to model a domain.
- Describe components of a Bayesian network.
- Compute a joint probability given a Bayesian network.
- Identify the conditional independence assumptions of a Bayesian network.
- Given a Bayesian network, determine if two variables are unconditionally or conditionally independent given a third variable.

Constructing a correct and good Bayesian Network for a domain

- Determine if a Bayesian network is a correct representation of a domain by determining whether the network correctly captures the conditional independence assumptions in the domain.
- Determine whether a Bayesian network is a good representation of a domain by calculating the number of probabilities required to define the Bayesian network.
- Construct a correct Bayesian network representation for a given domain.

Inference in Bayesian Networks - The Variable Elimination Algorithm

- Define factors. Manipulate factors using operations restrict, sum out, multiply and normalize.
- Compute a prior or a posterior probability given a Bayesian network.
- Describe/trace/implement the variable elimination algorithm for calculating a prior or a posterior probability given a Bayesian network.

5 Decision Making under Uncertainty

Introduction to Decision Networks

- Model a one-off decision problem by constructing a decision network containing nodes, arcs, conditional probability distributions, and a utility function.
- Choose the best action by evaluating a decision network.
- Determine the best action given a change in the utility function of a decision network.

Markov Decision Processes

- Describe components of a Markov decision process.
- Explain why the optimal solution to a Markov decision process cannot be a fixed sequence of actions.
- Describe how the solution to the Markov decision process changes depending on whether we choose to model it with a finite or an infinite horizon.
- Describe/trace the value iteration algorithm for calculating true utilities of states and the optimal policy of a Markov decision process.

6 Game Theory

- Determine dominant-strategy equilibria of a 2-player normal form game.
- Determine pure-strategy Nash equilibria of a 2-player normal form game.
- Determine whether one outcome Pareto dominates another outcome of a game. Determine Pareto optimal outcomes of a 2-player normal form game.
- Calculate a mixed strategy Nash equilibrium of a 2-player normal form game.

7 Machine Learning

Introduction to Machine Learning

- Identify reasons for building an agent that can learn.
- Define supervised learning, unsupervised learning, classification, and regression.
- Define and describe over-fitting in terms of how the training and test accuracy changes as the complexity of the model changes.

Decision Trees

- Describe the components of a decision tree.
- Construct a decision tree given an order of testing the features.
- Determine the prediction accuracy of a decision tree on a test set.
- Describe/trace/implement the ID3 algorithm.

Constructing Decision Trees

- Compute the entropy of a probability distribution.
- Compute the expected information gain for testing a feature.
- Describe/trace/implement the ID3 algorithm. Construct a decision tree by selecting a feature for each node using the expected information gain metric.
- Explain why the ID3 algorithm does not necessarily produce the optimal (i.e. the smallest) decision tree.

Extensions of Decision Trees

- Extend the IDS algorithm to handle more than two classes.
- Extend the IDS algorithm to handle real-valued features with binary tests at each node.
- Describe/trace/implement cross-validation for dealing with noisy data and preventing over-fitting.

A brief history of deep learning

- Describe the five major events in the history of deep learning: the initial rise of perceptrons, the first AI winter, the discovery of the back-propagation algorithm, the second AI winter, the triumph of deep learning in recent years.
- Describe the four lessons of deep learning as summarized by Geoff Hinton.

Artificial Neural Networks

- Describe components of a perceptron.
- Construct a perceptron to represent simple linear functions such as AND, OR, and NOT.
- Represent the XOR function using a three-layer feed-forward perceptron network.
- Explain why the back-propagation algorithm can be interpreted as a version of the gradient descent optimization algorithm.
- Execute the back-propagation algorithm given the update rules of the weights.