Course Summary

- Introduction to Artificial Intelligence
- C++ and Objected-Oriented Programming
- Scheduling and State Space Search
- Design and Rule-Based Programming
- Learning, Neural Networks, and Decision Trees
- Diagnosis and Bayes Networks
- Other Topics

Artificial Intelligence

- **Optimal Behavior for the Task**

  Often impossible to specify: (a) lack of knowledge, (b) competing objectives

  Often impossible to achieve: (a) computational complexity

- **The Best Behavior Achievable with the Available Resources**

Best Behavior

- **Scheduling**. Computational complexity requires heuristic algorithms.
- **Design**. Continuous design variables and computational complexity require constraint propagation methods.
- **Learning**. Small training set (uncertainty) makes perfect learning impossible. Computational complexity of finding the best hypothesis makes optimal learning impossible.
- **Diagnosis**. Computational complexity requires heuristic algorithms.

Scheduling and State Space Search: Basic Ideas

- **Problem Spaces**. Especially formulating alternative problem spaces. Completeness and redundancy. Branching Factor, Solution Depth, Solution Density. Experimental methods for estimating.
- **Search Algorithms**. States already expanded, frontier, unknown.
  Depth-first, breadth-first, best-first, A*.
  Greedy Search, Random Probing, Simulated Annealing.
- **Constructive Problem Space**.
- **Repair-Based Problem Space**.

Design: Basic Issues

- **Many Different Kinds of Design Problems**. Routine design.
- **Constructive Problem Space is Infinite**. Continuous design variables (e.g., shaft locations) make a constructive space infeasible.
- **Constraint Analysis**. Identify constraints that can be "propagated" to constrain (or even determine) solutions.
- **Intermix Generate-and-Test and Constraint Propagation**. Apply tests as soon as possible. Then propagate any constraints. Finally, try guessing possible solutions.

Rule-Based Programming

- **A rule is applied *whenever* its condition is satisfied.**
  Programmer does not need to anticipate all possible situations under which a rule should be fired.
- **Implemented by Active Values**. Whenever the value of an active value changes, it notifies all rules that use it.
- **Conflict Set**. Rules that are ready-to-run are not executed immediately. Instead, they are placed in the conflict set. This handles temporary inconsistencies in the parameter values.
- **Backwards Reasoning**. Rule-based programming can also be used for backwards reasoning.

Machine Learning: Basic Ideas

- **Target Function**. A function that classifies cases into one of k classes.
- **Training Data and Input Features**. Careful selection of features is critical.
- **Hypothesis Space**. Determined by representation.
- **Objective Function**. Errors on training set, errors on cross-validation set.

Neural Networks

- **Neural Networks**. Advantages of the sigmoid function (differentiable, linear region, approximates threshold function).
- **Fitting Neural Networks via Gradient Search**. Initialization, momentum, conjugate-gradient search.
- **Overfitting**. Cross-validated early stopping.
- **Advantages.**
  Generally good performance.
  Hidden units can learn new features.
- **Disadvantages.**
  Slow.
  Hard to interpret weights.

Decision Trees

- **Fitting Decision Trees.** Top-down divide-and-conquer.
- **Mutual-Information Splitting Heuristic**.
- **Overfitting**. Control via bottom-up pruning (cross-validated).
- **Advantages**.

  Fast.

  Easy to interpret.
- **Disadvantages**.

  Performance often not quite as good as carefully-trained neural network.

Nearest Neighbor

- **Trivial Algorithm**
- **kd-trees** for faster retrieval.
- **Advantages**.

  Simple to implement, fast learning.

  Often excellent performance.
- **Disadvantages**.

  Slow to classify new points.

  Does not provide any distillation of data.

  Sensitive to distance metric.

Reinforcement Learning

- **States, Actions, Rewards**
- **Policy, Value Function of a Policy**
- **Value Iteration**. Backed-up value.
- **Exploration**.
- **Approximating the Value Function via Neural Networks**

Diagnosis

- **Observations, Repairs**
- **Goal: Minimize total cost of repair**
- **Optimal diagnostic policy.**

  Computing via dynamic programming or reinforcement learning.

  Optimal repair-only policy.
- **Approximations to Optimal Policy for Multiple-Faults and Auxiliary Observations**

Reasoning Under Uncertainty

- **Joint Probability Distribution**. Explosive size.
- **Conditional Probability Tables**
- **Conditional Independence**
- **Bayes Networks**
- **Computing Failure Probabilities**
- **Factoring Algorithm for Bayes Networks**

Other Topics in AI

- **Computer Vision**. Edge detection, segmentation, depth maps, object identification.
- **Language Processing**. Language understanding, representing meaning, discourse structure, dialogue structure, language translation, language summarization.
- **Planning**. Designing a sequence of actions to achieve a goal. E.g., planning milling operations. Related to reinforcement learning.
- **Qualitative Simulation**. Predicting the future behavior of a system based on incomplete or qualitative information about the system.
- **Robotics**. Robot control, navigation, planning, learning, sensing, etc.
- **Distributed Intelligence**. Coordination of multiple intelligent agents in problem solving, planning, negotiating.

Final Exam

Monday, December 4, 12:00.