CS 604: Artificial Intelligence
Fall 2018

Projects

General Notes

- Each project can be worked on by a single student alone, or by at most two students together.
- To run project code for simulations and such, it is strongly recommended that Google Cloud or Amazon AWS, both of which offer free trial credits, be used. (Cloud credits with AWS or Microsoft Azure will also be available to some students.)

Topics

1. AI for Reading Diagrams with Text

   Many years ago, Jill H. Larkin and Herbert A. Simon (*Why a Diagram is (Sometimes) Worth Ten Thousand Words*, Cognitive Science vol. 11, pp. 65–99, 1987, doi:10.1111/j.1551-6708.1987.tb00863.x) gave a framework for parsing text as well as a certain class of diagrams, and indicated how diagrams can pack a lot of information.

   (a) Do one or both:

      i. Create a software solution that can "read" such diagrams as described by Larkin and Simon, and generate human-readable text descriptions.

      ii. Create a software solution that can draw a diagram given a natural-language description.

   (b) Create a software solution that can "read" text interspersed with diagrams.

2. Multiagent Systems and Agent-Based Modeling

   (a) Cognitive self-stabilization in autonomous agents

   One problem currently faced by autonomous systems, learning agents, and other cognitive systems is their extreme susceptibility to bad information and malicious misdirection. Specifically, there is absolutely no guarantee in general that an autonomous system will recover to a valid cognitive state after a transient exposure to bad information.

   Task: Develop a formulation based on self-stabilization or control theory to describe desirable cognitive design of autonomous systems so that they have the ability to unlearn after transient exposure to bad information, and also have some resistance to incorrect guidance after first having been trained on good information.

   (b) Multi-Agent System Modeling and Simulation of Pedestrian Evac

   Task: Extend prior work on agent-based modeling for pedestrian evacuation, to account for clustering, dissimilar agents, special circumstances (e.g., assisted/directed evacuation, fire, active shooter).

(c) Agent-Based Modeling and Simulation of Drug-Resistant Diseases

Drug resistance is a major problem in the treatment and cure of disases; several important drugs such as penicillin are now ineffective against some infectious pathogens while some diseases such as TB have drug-resistant strains that require special treatments.

Task: Develop a model and simulation of the onset of drug resistance in a disease by mal-administration of antibiotics, and also the subsequent spread of such drug-resistant disease strains.

3. Dynamic Auctions in Cloud

Auctions are widely applied in cloud computing for allocating resources. Each buyer submits bids for the set of resources and the winner is selected based on the provider's criteria. This kind of auction is called a single-shot auctions. If single-shot auctions are repeated over time, the decision space also grows exponentially, which leads to unintended repercussions. Hence, bank account mechanisms are proposed to address this issue (see Mirrokni, Vahab and Leme, Renato Paes and Tang, Pingzhong and Zuo, Song, *Dynamic Auctions with Bank Accounts*, Proceedings of IJCAI 16 – available online at https://www.ijcai.org/Proceedings/16/Papers/062.pdf).

Task: Implement bank account mechanism for dynamic auctions in cloud. This does not require you to work with OpenStack, etc., or even CloudSim, but it does require Python. You can start with small simulations and can then extend the simulation to Google traces.

4. Reinforcement Learning

Reinforcement learning (RL) is a paradigm in artificial intelligence.

(a) Query RL: the system has a budget and can learn specific types of information through direct queries rather than system feedback. It should minimize the time to obtain a solution given a budgetary constraint.

(b) Collaborative RL: different parts of a complex problem are learned in parallel by different agents, and their individual learnings must be combined into a global solution.

(c) RL under Category Theory: learning and policy can be posed in the framework of category theory.

5. Neuromorphic Computing

Neuromorphic engineering is an ongoing attempt to achieve high performance computing by mimicking the structure and functioning of the human brain.

Look up CARLsim (`http://www.socsci.uci.edu/~jkrichma/CARLsim/`) or Brian (`http://briansimulator.org/`). You may also want to look up the Nengo library (`https://github.com/nengo/nengo`) and the Neural Engineering Framework (`http://compneuro.uwaterloo.ca/research/nef.html`), as well as Nupic HTM (`https://numenta.org/`).

(a) Computer algebra: create a basic computer algebra system that has the ability of MACSYMA 1982, SHEEP, or similar classical system.

(b) Floating point arithmetic: extend existing work on carrying out floating-point arithmetic (IEEE 754) addition, to also allow subtraction and multiplication using a neuromorphic chip.

(c) LISP interpreter: create a neuromorphic interpreter for LISP or Scheme.

(d) Logic processing: create a neuromorphic system for processing expressions in sentential logic, similar to the Logic Theory Machine (Newell and Simon, 1956) or such.

6. Drug Discovery and Properties

   Drugs are constantly being sought to fight diseases. However, drugs have serious downsides, in terms of side-effects as well as interactions among drugs. Testing all drugs under all possible conditions is impractical and unethical also, hence machine learning is sought to be used.

   (a) Using available data, create a model to analyze how drugs interact with other drugs, and use it to predict how a new drug would interact with a known one.
   (b) Using available data, create a model to analyze drug side-effects, and use it to predict what the side-effects of a new drug are likely to be.

   For drug data, see the following:

   https://www.drugbank.ca/releases/latest
   http://crdd.osdd.net/chemdatabase.php
   https://www.fda.gov/drugs/informationondrugs/ucm079750.htm

   For side effects, interactions, etc., see the following:

   http://seeslab.info/downloads/drug-drug-interactions/
   http://sideeffects.embl.de/download/
   https://www.pharmgkb.org/downloads
   https://catalog.data.gov/dataset/drug-interaction-api
   http://www.dgidb.org/downloads

7. 3D Path Planning

   3D path planning is required in various applications such as robotics, self-driving cars, protein folding etc. Various algorithms such as A*, Dijkstra exist in the literature, which provide optimality guarantees. In large cluttered spaces, however, this is a computationally intensive problem to solve deterministically. Hence, the dominant approach in the last two decades has been via sampling based methods. Examples of such methods are rapidly-exploring random trees (RRTs), probabilistic roadmaps (PRM) and so on. Such techniques quickly arrive at feasible paths from an initial configuration to a goal configuration.

   However, the paths so produced are jagged, with several unnecessary branches. They need to be pruned/smoothened. One approach is to do a Dijkstra search (which is much better, because the search is over a smaller set of waypoints as opposed to the original path planning problem which is a huge search space). Another approach is spline fitting over the waypoints, which produces a smooth path.

   (a) Path pruning with constraint satisfaction: during the pruning operation, it is important to ascertain that the new path is safe and collision free. In addition, various constraints need to be accounted for. The height constraint is paramount, as is the curvature/turn rate constraint in all three directions.

       Reference: Yang et al. (2013). *Spline-Based RRT Path Planner for Non-Holonomic Robots*, https://link.springer.com/content/pdf/10.1007/s10846-013-9963-y.pdf.
   (b) 3D potential fields which avoid local minima: The artificial potential field method is an intuitive and effective motion planning technique. In this approach, an agent is considered to be moving under the influence of an attractive field (goal), and repulsive fields (obstacles and constraints). However, it is notorious for getting caught in local minima. In the current problem, spatial constraints (such as approach corridors and height bounds) are considered, which can bring further complexities. Multiple soft goals may be considered as well.

One approach is to consider an augmented 3d potential field cost function, and apply evolutionary techniques to find an optimal constraint satisfying path.

Reference: Montiel, et al. (2015). *Optimal Path Planning Generation for Mobile Robots using Parallel Evolutionary Artificial Potential Field*, https://link.springer.com/content/pdf/10.1007/s10846-014-0124-8.pdf.

srao@iiitb.ac.in, 12:42, Wednesday 1st August, 2018