IKT440 AI and Learning Systems

Student assistant: Ahmed A. Email: Ahmed.Abouzeid@uia.no

# 31-08: Learning in Random Environments

About the course: 3 parts:

* Lectures
  + Evaluation: Multiple choice quiz
* Assignments
  + Trying out techniques from lectures, hands-on experience (Pass/fail)
  + Two weeks each, with presentation at end
* Project
  + Groups of 2
  + Subjects presented at 14.09
  + Presentation (for feedback) 23.11
  + Submission: 17.12

I: Machine Learning

* Seeks to answer the following question:
  + “How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?”
* Covers a wide range of learning tasks:
  + Robots learning to walk/run etc.
  + Medical data being used to predict which treatment will work for a given patient
  + Search optimization to know what results give the most clicks etc.
* A learning problem is defined in terms of a
  + Task T
  + Performance metric P
  + Type of experience E
  + A machine is learning if the system reliably improves its performance P at task T, following experience E.
* When to use ML:
  + Application is too complex to manually implement
    - Sensor-based perception tasks
    - Image identification
  + The application requires specialization to its environment after release/shipping
    - Speech recognition that learns to recognize the owner/user’s voice
    - Recognizing a specific person in an image, for instance your mother.
  + Network Intrusion Detection
    - See what is normal traffic
    - Learn what intrusions are from sample data
    - Identify intrusions autonomously
  + Bio-surveillance
    - Detect/isolate outbreaks
      * Different covid variants etc.

II: Learning Automata

* Reinforcement learning
  + Don’t say this is a dog, that is a cat
  + Instead, tell the machine when it is correct and when it is wrong
    - Machine learns what is good behaviour
  + Reinforcement learning (RL): An agent that explores and environment
    - The agent: Perceives its current state and takes actions
    - The Environment: Provides a reward or penalty
    - RL Algorithms: Attempt to find an algorithm for maximizing the agent’s cumulative reward over the course of the problem
    - Exploration vs exploitation
      * If you have a favourite restaurant, you might visit it disproportionately, possibly missing another, better restaurant
      * Stop exploration too early – miss rewards
      * Stop exploration too late – waste time on lower rewards
      * **AKA The multi-armed bandit problem:**
        + An agent sequentially pulls the levers of gambling machines
        + Each pull results in a reward or a penalty
        + The reward probabilities are unknown
        + Challenge: Balance exploiting existing knowledge and obtaining new information in order to maximize rewards
* The learning automata
  + Came from psychology (modelling behaviour), statistics (model choice of experiments based on observations), system theory (make rational decisions in random environments)
  + Adaptive decision making devices that can operate in
    - Unknown environments – does not know about the effect of their actions at the start of the operations
    - Random environments: an action does not necessarily produce the same response each time it is performed
  + A powerful property: Systematically improving
    - Combine rapid and accurate convergence
    - Low computational complexity – easy to implement and run
  + Example decisions:
    - Network routing – which path is fastest?
    - Choosing between air travel or car travel to a neighbouring city
  + Outcome of choice is random:
    - Time to reach a destination using one route is assumed to be a random variable depending on traffic conditions.
  + Other applications:
    - Call routing
    - Board games like Go
    - Resources allocation in web polling
    - Pattern recognition (Tsetlin Machine)
  + The environment:
    - Input: Action from Set of Actions
    - Output: Reward or Penalty
    - Penalty probabilities: For action i, there is a probability that the Environment responds with a penalty
      * P(Penalty|Action ai) = ci
    - An environment can be
      * Static: Penalty probabilities do not change over tine
      * Dynamic: Penalty probabilities change
    - Action A 🡪 Environment C 🡪 Response R

III: Game theory and automata games

* Decentralized decision making is a common, often necessary feature of complex natural and man-made systems
  + Arises from the reality that the complete information exchange needed for centralized decision making may not be feasible
  + Formidable problem: Coordination of decentralized decision makers
* Nash equilibrium: Neither player benefits from changing strategy
* Pareto optimal: Neither player can improve their results without hurting another player

IV: Assignment

* Goore game: Judge gives rewards based on number of yes-votes it receives in an attempt to optimize number of yes-votes when voters are simultaneously and independent
* Example code on canvas, but remake environment class