Final Exam Review

CMPUT 366: Intelligent Systems

Weeks 1-13

Lecture Structure

1. Imperfect information extensive form games
2. Exam structure and details
3. Learning objectives walkthrough
    - **Clarifying questions** are the point of this class
4. Other questions, clarifications

Final Exam Details

- The final exam is **Wednesday, April 28** via **eClass**
- There will be a **6 hour** time limit for the exam
  - Starting at **any time** between 12:01am and 11:59pm Mountain time
  - It should *not* take anywhere near this long (I aimed for it to take **2 hours**)
- You may use a **single**, **handwritten cheat sheet** if you wish
- You may use a non-programmable calculator if you wish
- **All course material** is included
  - Weeks 8-13 will be more heavily weighted than weeks 1-7

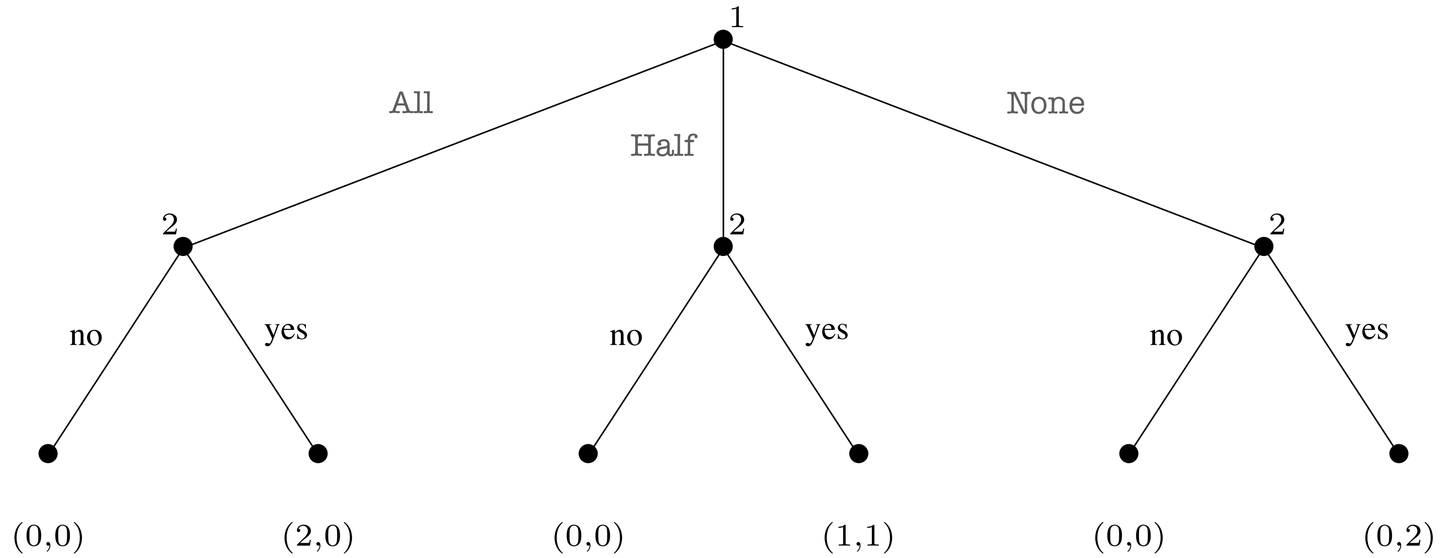
Final Exam Structure

- There will be **120 marks** total
- There will be **15 short answer** questions with 1-2 sentence answers
  - The rest will be more in-depth
- There will be **no coding** questions
  - But you may be asked to **execute a few steps** of an algorithm
- Every question will be based on the **learning objectives** that we are about to walk through
- **There will be five marks for uploading a picture of your cheat sheet**

Recap: Perfect Information Extensive Form Game

Definition:
A **finite perfect-information game in extensive form** is a tuple $G = (N, A, H, Z, \chi, \rho, \sigma, u)$, where

- N is a set of n **players**,
- A is a single set of **actions**,
- H is a set of nonterminal **choice nodes**,
- Z is a set of **terminal nodes** (disjoint from H),
- $\chi : H \to 2^A$ is the **action function**,
- $\rho : H \to N$ is the **player function**,
- $\sigma : H \times A \to H \cup Z$ is the **successor function**,
- $u = (u_1, u_2, \ldots, u_n)$ is a **utility function** for each player, $u_i : Z \to \mathbb{R}$



Imperfect Information, informally

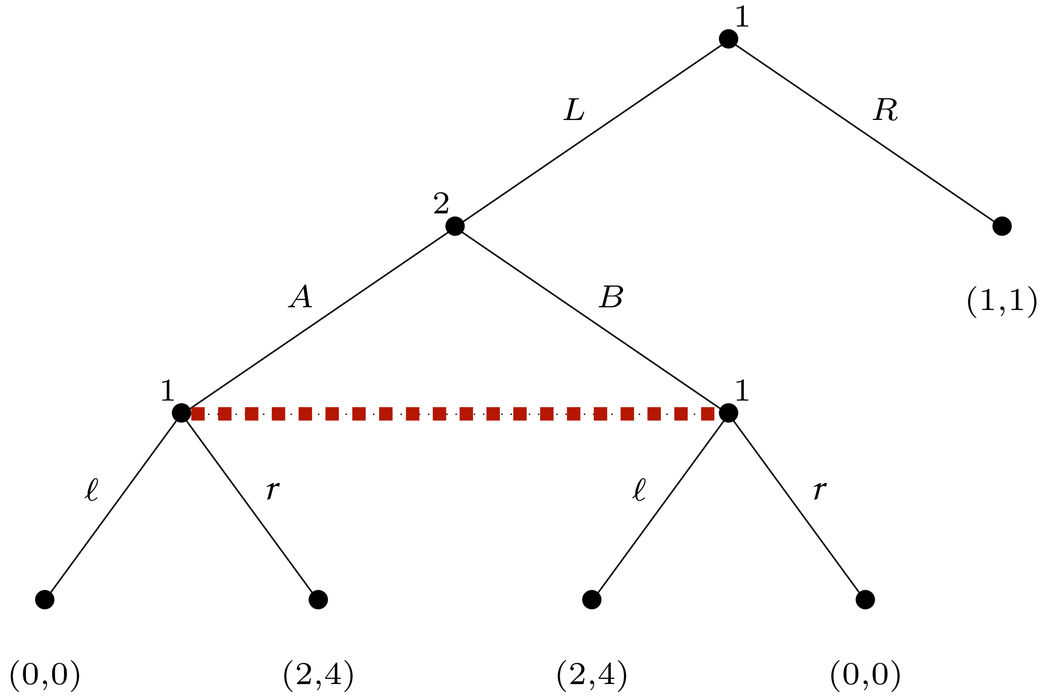
- **Perfect information** games model **sequential** actions that are **observed by all players**
  - **Randomness** can be modelled by a special **Nature** player with constant utility and known mixed strategy
- But many games involve **hidden** actions
  - Cribbage, poker, Scrabble
  - Sometimes actions of the **players** are hidden, sometimes **Nature's** actions are hidden, sometimes both
- **Imperfect information extensive form games** are a model of games with sequential actions, some of which may be **hidden**

Imperfect Information Extensive Form Game

Definition:
An **imperfect information game in extensive form** is a tuple $G = (N, A, H, Z, \chi, \rho, \sigma, u, I)$, where

- $(N, A, H, Z, \chi, \rho, \sigma, u)$ is a perfect information extensive form game, and
- $I = (I_1, \ldots, I_n)$, where $I_i = (I_{i,1}, \ldots, I_{i,k_i})$ is an **equivalence relation** on (i.e., partition of) $\{h \in H : \rho(h) = i\}$ with the property that $\chi(h) = \chi(h')$ and $\rho(h) = \rho(h')$ whenever there exists a $j \in N$ for which $h \in I_{i,j}$ and $h' \in I_{i,j}$.

Imperfect Information Extensive Form Example



- The members of the equivalence classes are also called **information sets**
- Players **cannot distinguish** which **history** they are in within an information set
- **Question:** What are the information sets for each player in this game?

Pure Strategies

Question: What are the **pure strategies** in an **imperfect information** extensive-form game?

Definition:

Let $G = (N, A, H, Z, \chi, \rho, \sigma, u, I)$ be an imperfect information game in extensive form. Then the **pure strategies of player i** consist of the cross product of actions available to player i at each of their **information sets**, i.e.,

$$\prod_{I_{i,j} \in I_i} \chi(h)$$

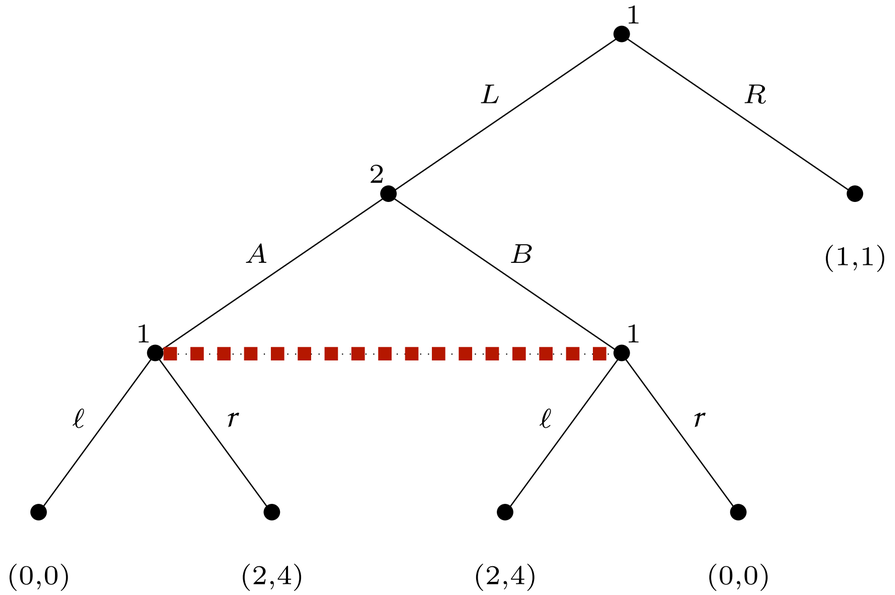
- A pure strategy associates an action with **each** information set, even those that will **never be reached**

Questions:

In an imperfect information game:

1. What are the **mixed strategies**?
2. What is a **best response**?
3. What is a **Nash equilibrium**?

Induced Normal Form



	A	B
L,ℓ	0,0	2,4
L,r	2,4	0,0
R,ℓ	1,1	1,1
R,r	1,1	1,1

Question:
Can you represent an arbitrary **perfect information** extensive form game as an **imperfect information** game?

- Any pair of pure strategies uniquely identifies a **terminal node**, which identifies a **utility** for each agent
- We have now defined a set of **agents**, **pure strategies**, and **utility functions**
- Any extensive form game defines a corresponding **induced normal form game**

Extensive Form Games Summary

- **Extensive form games** model **sequential** actions
- **Pure strategies** for extensive form games map **choice nodes** to **actions**
  - **Induced normal form:** normal form game with these pure strategies
  - Notions of mixed strategy, best response, etc. **translate directly**
- **Perfect information:** Every agent **sees all actions** of the other players
  - **Backward induction** computes a **pure strategy Nash equilibrium** for any perfect information extensive form game
- **Imperfect information:** Some actions are **hidden**
  - Histories are partitioned into **information sets**; players **cannot distinguish** between histories in the same information set

Introduction to AI

- characterize simplifying assumptions made in building AI systems
- determine what simplifying assumptions particular AI systems are making
- suggest what assumptions to lift to build a more intelligent system than an existing one
- define the major representational dimensions
- classify problem statements by representational dimensions

Search

- define a directed graph
- represent a problem as a state-space graph
- explain how a generic searching algorithm works

Search (2)

- demonstrate how depth-first search will work on a graph
- demonstrate how breadth-first search will work on a graph
- demonstrate how iterative deepening DFS will work
- demonstrate how least cost first search will work on a graph
- predict the space and time requirements for depth-first and breadth-first searches

Search (3)

- devise a useful heuristic function for a problem
- demonstrate how best-first and A* search will work on a graph
- predict the space and time requirements for best-first and A* search
- justify why and when depth-bounded search is useful
- demonstrate how iterative-deepening works for a particular problem
- demonstrate how depth-first branch-and-bound works for a particular problem

Search (4)

- define hill climbing, random step, random restart
- explain why hill climbining is not complete
- explain why adding random restarts to hill climbing makes it complete
- justify when local search is appropriate for a given problem

Search (5)

- list the elements of a local search problem
- recognize a local search problem
- explain how the generic local search algorithm works
- define hill climbing and stochastic local search
- trace an execution of hill-climbing and stochastic local search
- define improving step, random step, and random restart
- explain the benefits of random steps and random restarts

Uncertainty

- define a random variable
- describe the semantics of probability
- apply the chain rules
- apply Bayes' theorem

Uncertainty (2)

- define a belief network
- build a belief network for a domain
- build a correct belief network for a given joint distribution
- compute marginal and conditional probabilities from a joint distribution
- describe the semantics of a belief network
- identify the independence guarantees encoded by a belief network

Uncertainty (3)

- define the factor objects and factor operations used in variable elimination
- explain the origins of the efficiency improvements of variable elimination
- define the high-level steps of variable elimination
- trace an execution of variable elimination

Uncertainty (4)

- justify why a belief network is a correct encoding of a joint distribution
- identify the factorization of a joint distribution encoded by a belief network
- answer queries about independence based on a belief network
- answer queries about independence based on a joint distribution

Causality

- define observational and causal query
    - explain the difference
- explain why causal queries on observational distributions can go wrong
- construct the post-intervention distribution for a causal query from an observational distribution
- evaluate a causal query given an observational distribution
- justify whether a causal model is valid
- define selection effect

Supervised Learning

- define supervised learning task, classification, regression, loss function
- represent categorical target values in multiple ways (indicator variables, indexes)
- identify an appropriate loss function for different tasks
- explain why a separate test set estimates generalization performance
- define 0/1 error, absolute error, (log-)likelihood loss, mean squared error, worst-case error

Supervised Learning (2)

- define generalization performance
- construct a decision tree using given features, splitting conditions, and stopping conditions
- define overfitting
- explain how to avoid overfitting

Supervised Learning (3)

- explain how to use the Beta and Bernoulli distributions for Bayesian learning
- derive the posterior probability of a model using Bayes' rule
- define conjugate prior
- demonstrate model averaging

Supervised Learning (4)

- estimate expectations from a finite sample
- apply Hoeffding's inequality to derive PAC bounds for given quantities
- demonstrate the use of rejection sampling and importance sampling

Deep Learning

- define an activation
- define a rectified linear unit and give an expression for its value
- describe how the units in a feedforward network are connected
- give an expression in matrix notation for a layer of a feedforward network
- explain at a high level what the Universal Approximation Theorem means
- explain at a high level how feedforward neural networks are trained
- identify the parameters of a feedforward neural network

Deep Learning (2)

- trace an execution of forward-mode automatic differentiation
- trace an execution of backward-mode automatic differentiation
- explain why automatic differentiation is more efficient than symbolic differentiation or the method of finite differences
- explain why backward-mode automatic differentiation is more efficient for typical deep learning applications than forward-mode
- explain how gradient descent is used to train a neural network

Deep Learning (3)

- define sparse interactions and parameter sharing
- define the convolution operation
- define the pooling operation
- explain why convolutional networks are more efficient to train
- describe how the units/layers in a convolutional neural network are connected

Deep Learning (4)

- demonstrate unfolding a recurrent expression
- explain the problems with handling sequence input using non-recurrent dense or convolutional multi-layer neural networks
- describe the high-level idea behind recurrent neural networks
- explain why recurrence through outputs is strictly less general than recurrence through hidden layers

Reinforcement Learning

- define a Markov decision process and a policy
- define and give expressions for the **state-value function** and the **action-value function**
- state the Bellman optimality equations
- define returns and give expressions for episodic and discounted continuing returns
- represent a problem as a Markov decision process

Reinforcement Learning (2)

- justify why a policy is weakly better than another
- trace an execution of iterative policy evaluation
- state the **Policy Improvement Theorem** and explain **why it is important**
- trace an execution of the Value Iteration algorithm

Reinforcement Learning (3)

- explain how Monte Carlo estimation for state values works
- trace an execution of the first-visit Monte Carlo Prediction algorithm
- explain the difference between prediction and control

Reinforcement Learning (4)

- define on-policy vs. off-policy learning
- define a behaviour policy
- explain what exploring starts are and why they are necessary
- define an ε-soft policy
- explain when and why ε-soft policies are useful

Reinforcement Learning (5)

- trace an execution of the TD(0) algorithm
- trace an execution of the Q-learning algorithm
- trace an execution of the Sarsa algorithm
- define bootstrapping
- explain why bootstrapping is useful

Reinforcement Learning (6)

- explain why function approximation is useful
- explain the difference between action-value and policy gradient methods
- trace an execution of the REINFORCE algorithm

Multiagent Systems

- define **best response** and **Nash equilibrium**
- define Pareto dominance and Pareto optimality
- explain the difference between pure strategy and mixed strategy Nash equilibria

Multiagent Systems (2)

- trace an execution of backward induction
- explain the difference between imperfect information and perfect information extensive form games
- define an information set
- identify the pure strategies in an extensive form game

Questions?