

Midterm Review

CMPUT 366: Intelligent Systems

Weeks 1-8

Lecture Structure

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

Midterm Details

- The midterm is **Friday, March 11** via **eClass** (no lecture that day)
- There will be a **1 hour** time limit for the midterm
 - Starting at **any time** between 12:01am and 11:59pm Mountain time
 - But if you start after 10:59pm you will get less than an hour
- You may use a **single, handwritten cheat sheet** if you wish (double-sided)
 - You will get 2 bonus marks for uploading a picture of it!
- You may use a non-programmable calculator if you wish
- All material from lectures before the midterm are included
 - Everything up to and including Recurrent Neural Networks
- **Assignment solutions are posted to eclass**

Midterm Structure

- There will be **60 marks** total
- There will be **10 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
- Every question will be answerable in eClass
 - I.e., text-box or multiple-choice; no need to upload pictures (except cheat sheet)

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 search problem formally
- 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**
- define an **admissible heuristic**
- 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 climbing 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 rule
- 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
- execute the convolution operation
- execute 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

Questions?