3.5 Uniformed Search Strategies

DFS

1. What is the data structure for the frontier?

   LIFO stack

2. What its the time and space complexity?

   Time is exponential

   Space is linear

3. What is the worst case time complexity? Why?

   It might be infinite

4. Why is DFS said to involve backtracking?

   The algorithm selects a first alternative at each node, and it backtracks to the next alternative when it has pursued all of the paths from the first selection.

5. When is DFS useful?

   When the space is limited

   Many solutions exist, perhaps with long path lengths, particularly for the case where nearly all paths lead to a solution

6. When is DFS a poor method?

   Infinite branch

   Infinite paths

   Solutions exist at shallow depth

BFS

1. What is the data structure for frontier?

Queue

2. What is the time and space complexity?

   Both are exponential

3. When is BFS useful?

   When the space is not an issue

   The goal nodes are shallow

   You want to find the solution containing the fewest arcs

4. When is BFS a poor method?

   When the space is limited

Lowest-Cost-First Search

1. What is the difference between LCFS and DFS/BFS?

   LCFS is used when the edges have unequal weights

2. If the costs of the arcs are bounded below by a positive constant, and the branching factor is finite, is LCFS guaranteed to find an optimal solution?

   Yes

3. What is the time and space complexity of LCFS?

   Both are exponential

3.6 Heuristic Search

Heuristic Search

1. What does it mean when we say a function $h(n)$ is an underestimate?

   When it is always less than the actual cost from node n to goal

2. Describe Heuristic DFS. How is it different from DFS?

   The nodes are added to the frontier in the order of their heuristic

   Local min $h(n)$

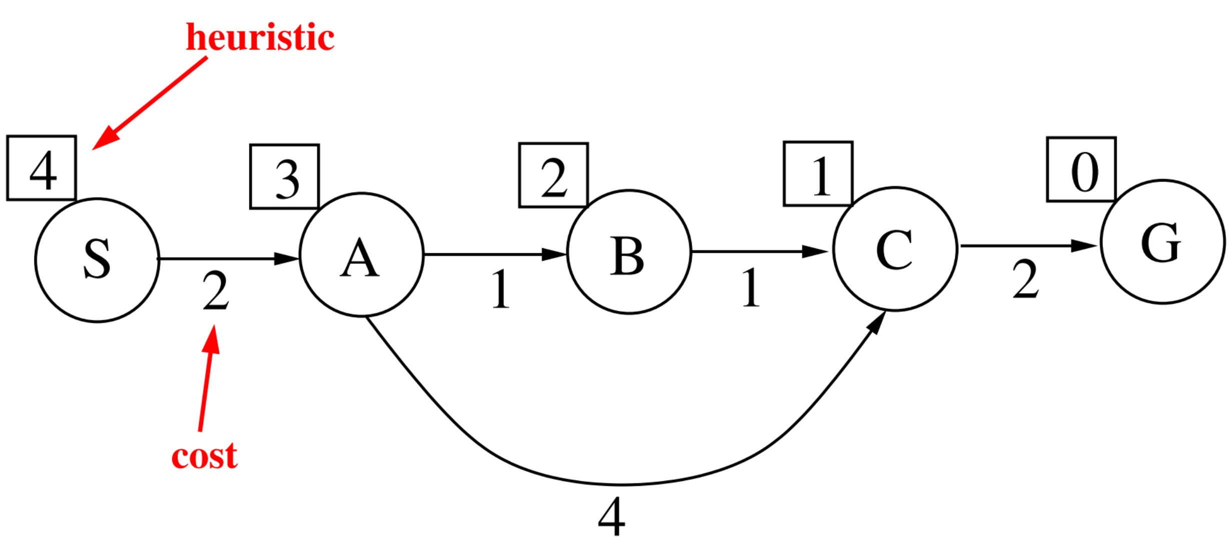
3. Describe best-first search

   Global min $h(n)$

4. Why is best-first search not used in practice?

   It can follow paths that look promising because they are close to the goal, but the costs of the paths may keep increasing

5. Give the path that best-first search will select. Is it optimal?



S-A-C-G

A* Search

1. Which two searches is A* search a combination of? Why?

   LCFS and best-first search

2. How does A* search estimate the total path cost from a start node to a goal node?

   $cost + h(n)$

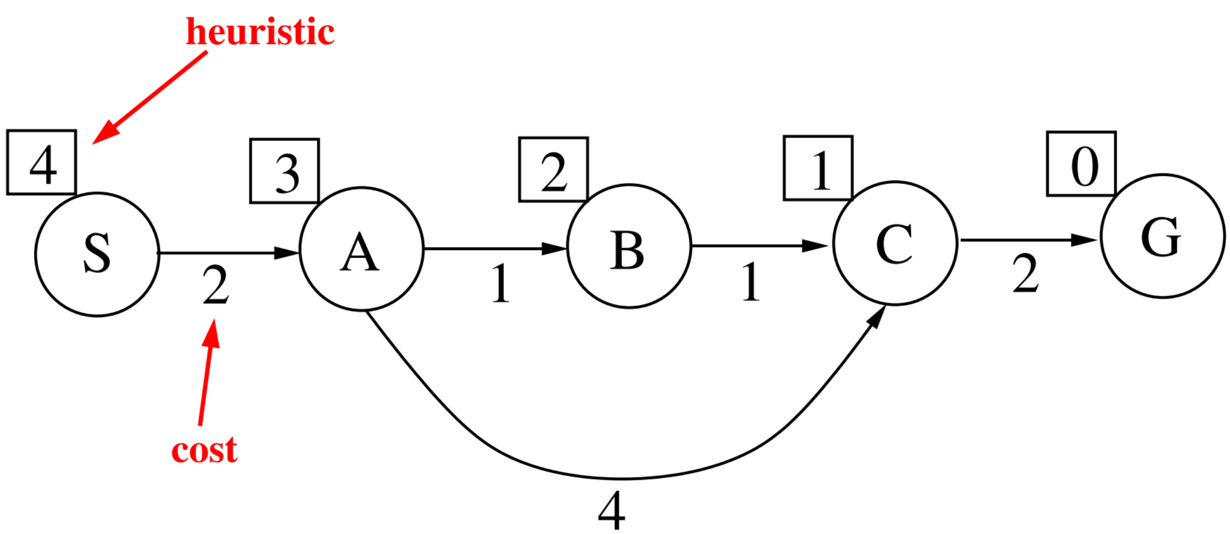
3. What data structure is used for A* search frontier?

   Priority queue ordered by $f(n) = cost + h(n)$

4. What is admissibility?

   $h(n)$ is always an underestimate of the cost from n to goal node

5. Give the path that A* search will select. Is it optimal?



6. What are the time and space complexities of A* Search?

   Both are exponential

3.7 More Sophisticated Search

Multiple path pruning

1. Briefly describe how it can be implemented.

   unordered_set of nodes

2. Does it necessarily guarantee that the shortest path is not discarded?

   No

3. What can we do to ensure an optimal path to a goal?

   Ensure that the heuristic function is monotone restricted

   Make sure that the first path found to any node is a lowest-cost path to that node, then prune all subsequent paths found to that node

   Remove all paths that used the higher-cost path to the node

   Whenever the search finds a lower-cost path to a node than a path to that already found, it can incorporate a new initial section on the paths on the paths that have extended the initial path

4. In LCFS, can pruning subsequent paths to a node remove a lower-cost path to that node?

   No

5. In A*, can pruning subsequent paths to a node remove a lower-cost path to that node?

   Yes

6. What is a monotone restriction on h? Why do we need that?

   $|h(n\prime) - h(n)| \leq d(n\prime, n)$. To ensure that we never prune a lower-cost path to a node

7. What is preferred for breadth-first methods? Multiple-path pruning or cycle check?

   Multiple-path pruning

8. What is preferred for depth-first methods? Multiple-path pruning or cycle check?

   Cycle check

Iterative Deepening

1. Describe iterative deepening

   Multiple depth-bound DFS where the depth increases in each iteration.

2. What is the complexity of iterative deepening?

   Exponential

Bidirectional Search

1. Briefly describe bidirectional search

   One search from the start node and another search from the goal node at the same time

Island-Driven search

1. Briefly describe island-driven search

   Split the problem into simpler, sub-problems called “islands”

4.5 Consistency Algorithms

1. What is domain consistent?

   When there is no value in the domain that violates the domain constraint

2. What is arc consistent?

   $\langle X, c(X, Y)\rangle$ is arc consistent if for every value in the domain of X, there is a value in the domain of Y that satisfies the constraint between X and Y. This is a constraint on X.

3. When AC-3 terminates. What does it mean if...

   a. every domain is empty?

      No solution

   b. every domain has a single value?

      One solution

   c. some domain has more than one value?

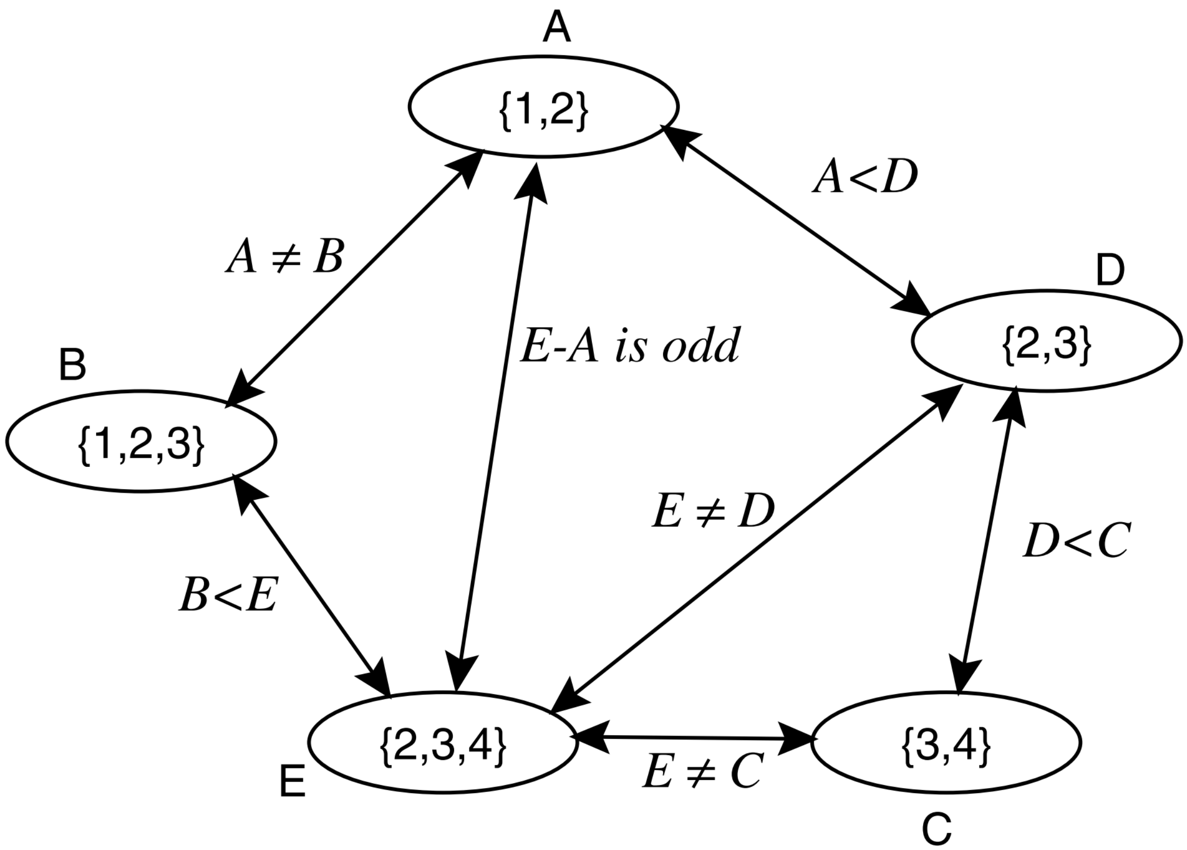
      Multiple solutions, one solution or no solution

4. What is the time complexity of AC-3?

   $O(cd^3)$ where c is the number of binary constraints, and the size of each domain is at most d. Each arc can be added to the queue at most d times because we can delete at most d values from X_i. Checking consistency of each arc can be done in $O(d^2)$ time.

4.7 Variable Elimination

1. Consider the arc-consistent network. Perform a variable elimination on C.



2. Consider a CSP that contains the variables A, B, and C, each with domain {1, 2, 3, 4}. Suppose the constraints that involve B are A < B and B < C. Perform a variable elimination on B.

A	B
1	2
1	3
1	4
2	3
2	4

3	4

B	**C**
1	2
1	3
1	4
2	3
2	4
3	4

Join the tables:

A	**B**	**C**
1	2	3
1	2	4
1	3	4
2	3	4

Project onto A and C

A	**C**
1	3
1	4
2	4

Remove variable B and add the last table as a new constraint on A and C.

Solve the problem recursively and join the result with the second last table above.

4.8 Local Search

1. What is a plateau?

   An area where heuristic function is not informative

2. What is a ridge?

   A local minimum where an n-step look ahead might be useful

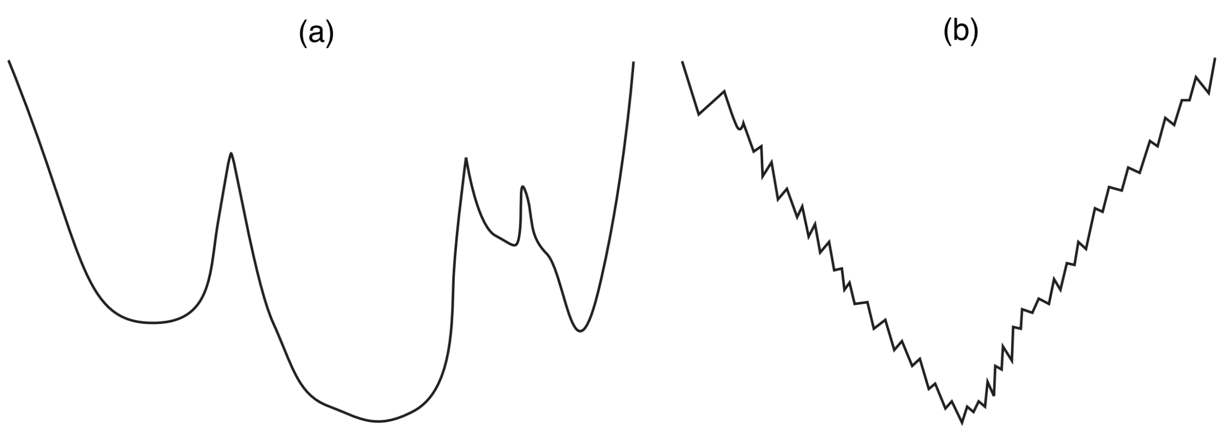
3. What is random step?

   Move to a random neighbour

4. What is random restart?

   Reassign random values for all variables

5. For these two search spaces. Which method would most easily find the global minimum?



(a) random restart (b) random step

6. What is stochastic local search a mix of?

   Greedy descent, random walk, random restart

7. Describe simulated annealing

   Start with a temperature T. Choose a random variable and a random new value. If there is an improvement, then accept it, otherwise accept it with probability proportional to T and $h(n') - h(n)$

8. Describe tabu list

   Keep track of the last k assignments. Don't allow an assignment that is already on the tabu list.

9. Describe parallel search

   Perform multiple searches at the same time

10. Describe beam search

    Like parallel search, but chooses the best k individuals

11. Describe stochastic beam search (asexual reproduction)

    Like beam search, but it probabilistically chooses the best k individuals at the next generation. The probability that a neighbor is chosen is proportional to its heuristic value

5.2 Propositional Definite Clauses

Proofs

1. Consider the following:

$$\texttt{rain} \leftarrow \texttt{clouds} \land \texttt{wind}.$$
$$\texttt{clouds} \leftarrow \texttt{humid} \land \texttt{cyclone}.$$
$$\texttt{clouds} \leftarrow \texttt{near\_sea} \land \texttt{cyclone}.$$
$$\texttt{wind} \leftarrow \texttt{cyclone}.$$
$$\texttt{near\_sea}.$$
$$\texttt{cyclone}.$$

   a. Do a bottom-up proof

   b. Do a top-down proof

2. For the following:

$$a \leftarrow b \wedge c.$$

$$b \leftarrow d \wedge e.$$

$$b \leftarrow g \wedge e.$$

$$c \leftarrow e.$$

$$d.$$

$$e.$$

$$f \leftarrow a \wedge g.$$

a. Do a bottom-up proof

{d, e}

{d, e, b}

{d, e, b, c}

{d, e, b, c, a}

b. Do a top-down proof

yes ← a

yes ← b ∧ c

yes ← d ∧ e ∧ c

yes ← d ∧ e ∧ e

yes ← d ∧ e

yes ← e

yes ← .

6.4.2 Approximate Inference Through Stochastic Simulation

Forward Sampling

1. Describe forward sampling.

   It samples the parent node, then its children.

2. Suppose $X_1, \ldots, X_n$ is a total ordering of the variables so that the parents of a variable come before the variable in the total order. How does forward sampling draw a sample for each variable?

   It samples X_1, then X_2,...

3. What is Hoeffding's inequality? How can it be used?

   It can be used to compute the minimum number of samples needed if you want an error bigger than x in only a certain proportion of the cases.

Rejection Sampling

1. Describe rejection sampling.

   It is like forward sampling, but if any example contains a variable that is assigned a different value than the one in the evidence of our query, it will be rejected.

2. What does the error in the probability of h depends on in rejection sampling?

   It depends on the number of samples that are not rejected.

3. Under what circumstance does rejection sampling not work well?

   When the evidence is unlikely.

Importance Sampling

1. Describe importance sampling.

Each sample is assigned a weight.

2. Describe two ways in which importance sampling differs from rejection sampling.

   There is no rejection in importance sampling.

   Importance sampling can use any proposal distribution, whereas rejection sampling essentially uses $q(a) = P(a)$.

   Importance sampling does not sample all variables, only some of them. The variables that are not sampled and are not observed are summed out (i.e., some exact inference is carried out).

3. Suppose $P(a) = 0.98$ and suppose $A = true$ is sampled 50% of the time, what will be the weight of each sample with $A = true$ and $A = false$?

   0.98/0.5=1.96 and 0.02/0.5=0.04.

Particle Filtering

1. Describe particle filtering.
2. When does this algorithm have an advantage?
3. Describe resampling.

6.5 Probability and Time

Markov Chains

1. State the Markov assumption.
2. What does it mean when we say a Markov chain is stationary?

Hidden Markov Models (HMM)

1. Why is HMM an augmentation of the Markov chain?
2. What is filtering/monitoring?
3. What is smoothing?

Monitoring and Smoothing

1. Derive (filtering/monitoring)

$$\alpha_i = P(S_i \mid o_0, \ldots, o_i) \propto P\left(o_i \mid S_i\right) \sum_{S_{i-1}} P\left(S_i \mid S_{i-1}\right) \alpha_{i-1}$$

2. Derive (smoothing)

$$\beta_{i+1} = P\left(o_{i+1} \ldots, o_T \mid S_i\right) = \sum_{S_{i+1}} P\left(o_{i+1} \mid S_{i+1}\right) P\left(S_{i+1} \mid S_i\right) \beta_{i+2}$$

Dynamic Belief Networks

1. What is a dynamic belief network (DBN)?

9a Slides

1. Describe the idea of Bayesian learning.

   Multiple hypotheses are used at the same time when making predictions.

2. Suppose X is input features, and Y is target feature, $d = \{x_1, y_1, x_2, y_2, ..., x_N, y_N\}$ is the data, x is a new input, and we want to know the corresponding output y. Given models M.

   $P(Y|x, d) = \sum_{m \in M} P(Y|x, m)(m|d)$.

3. Describe the idea of maximum a posteriori (MAP) learning. How is it different from Bayesian learning?

   It uses the hypothesis that maximizes the posterior $P(H|\mathbf{d}) \propto P(\mathbf{d}|H)P(H)$. That is, it makes prediction using $h^* = \arg\max_{h_i} P(h_i|\mathbf{d})$.

4. Describe the idea of maximum likelihood (ML) learning. How is it different from MAP learning?

   It uses the hypothesis that maximizes the likelihood $P(\mathbf{d}|H)$. That is, it makes prediction using $h^* = \arg\max_{h_i} P(\mathbf{d}|h_i)$.

5. Order Bayesian learning, MAP learning and ML learning in terms of their accuracy and susceptibility to overfitting.

   Accuracy: Bayesian > MAP > ML

   Susceptibility to overfitting: ML > MAP > Bayesian

6. Consider the following Naïve Bayes classifier

   a. What are the parameters?

      $\theta_C = P(C = 1)$

      $\theta_{i0} = P(A_i = 1|C = 0)$

      $\theta_{i1} = P(A_i = 1|C = 1)$

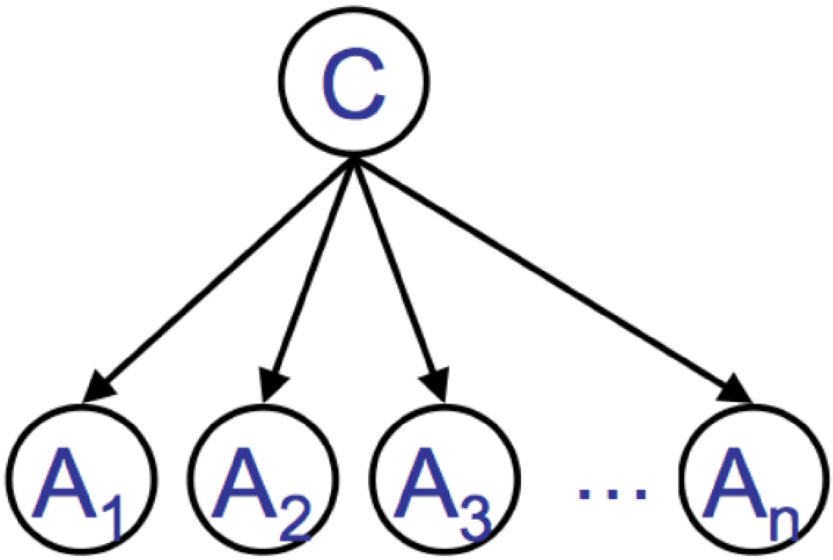
   b. How to make predictions using the parameters?

Calculate $P(A_1, \ldots, A_n|C = 1)P(C = 1)$ and $P(A_1, \ldots, A_n|C = 0)P(C = 0)$

$P(A_1, \ldots, A_n|C = 1)P(C = 1) = \theta_C \prod_i \theta_{i1}^{a_i}(1 - \theta_{i1})^{1-a_i}$

$P(A_1, \ldots, A_n|C = 0)P(C = 0) = (1 - \theta_C) \prod_i \theta_{i0}^{a_i}(1 - \theta_{i0})^{1-a_i}$

Then normalize them.



8. If a feature never occurs in the training set, but does in the test set, ML may assign zero probability to a high likelihood class. How can Laplace correction be used to fix the problem?

   It adds one to the numerator and d (number of possible values in the domain) to the denominator.

9b Slides

1. Write down the weight update formula, given the learning rate η and weights $\vec{w}$

   $$w_i \leftarrow w_i - \eta \frac{\partial(Error, \vec{w})}{\partial w_i}$$

2. What is incremental gradient descent?

   Use one example at a time

3. What is stochastic gradient descent?

   Pick a random example at a time

4. What is batched gradient descent?

   Use n examples at a time

5. Define linearly separable. Are all data linearly separable? If not, give a counterexample.

6. What do support vector machines do?

   Find a boundary that maximizes the distance between the line and the data points.

7. Write the definition of the following activation function and their properties

   a. Step function

   b. Sigmoid or logistic activation function

   c. Rectified linear unit (ReLU)

   d. Leaky ReLU

8. In the backpropagation algorithm, what do these do?

   a. Forward pass

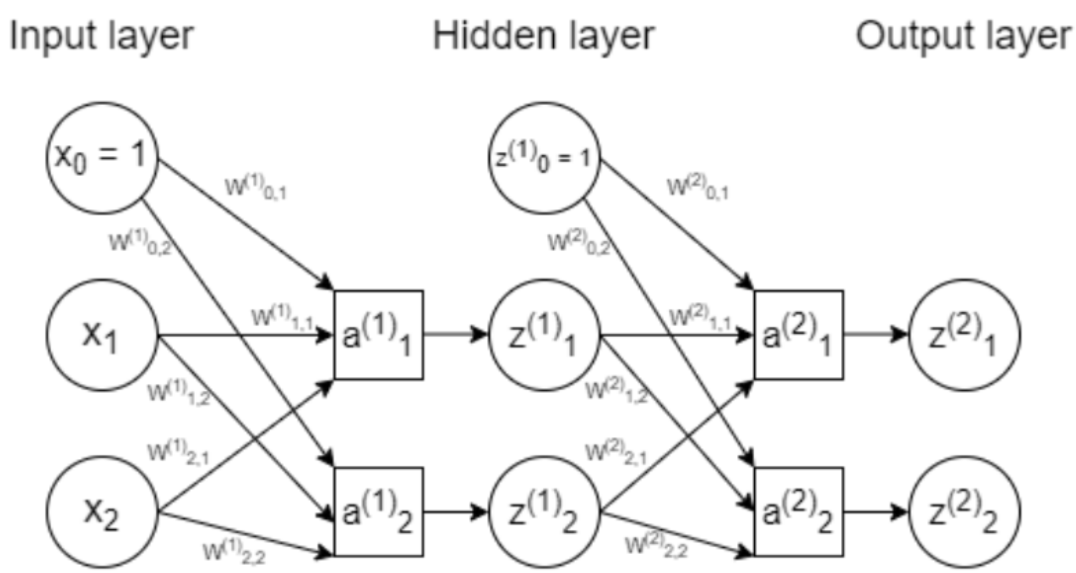
      Compute the a and z-values, and the error.

   b. Backward pass

Compute the gradients $\frac{\partial Error}{\partial W}$

9. Define feedforward network and recurrent network.

10. For the 2-layer network:



a. Show the forward pass

$$\begin{array}{ll} a_j^{(1)} = \sum_i W_{ij} x_i & z_j^{(1)} = f(a_j^{(1)}) \\ a_k^{(2)} = \sum_j W_{jk} z_j^{(1)} & z_k^{(2)} = f(a_k^{(2)}) \end{array}$$

b. Show the backward pass

$$\frac{\partial Error}{\partial W_{j,k}^{(2)}} = \delta_j^{(2)} z_j^{(1)} = \frac{\partial Error}{\partial z_j^{(2)}} f'(a_k^{(2)}) z_j^{(1)}$$
$$\frac{\partial Error}{\partial W_{i,j}^{(1)}} = \delta_j^{(1)} x_i = \sum_k W_{j,k} \delta_k^{(2)} f'(a_j^{(1)}) x_i$$

9. What is the purpose of regularization?

Prevent overfitting

9c Slides

1. Consider the following ways to deal with missing data. What are the problems?
   a. Ignore hidden variables

      The number of possible values for some variables will increase significantly

   b. Ignore records with missing values

      You cannot ignore them unless they are missing at random. Sometimes variables can be missing for specific reasons.

2. How can the k-means algorithm be used?

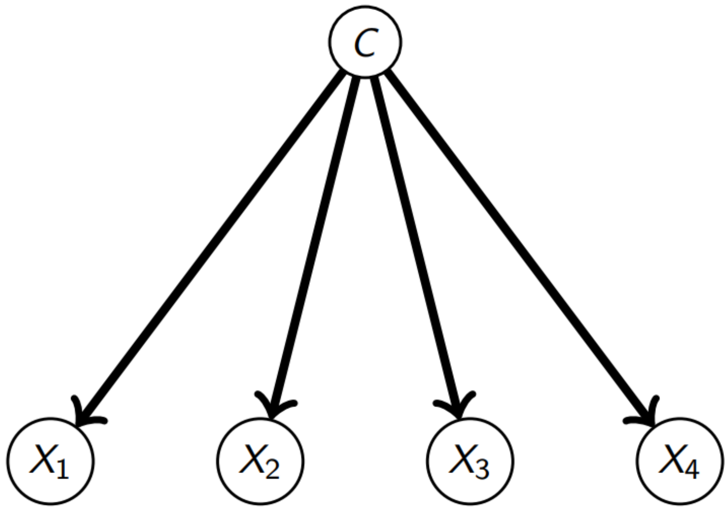
   It can be used for clustering (unsupervised learning).

   Taking training examples and the number of classes k as inputs, it outputs a representative value for each input feature for each class and an assignment of examples to classes.

3. Describe the idea of the k-means algorithm.

   Initialize a random value representing the mean for each class. Assign each input to the closest mean. Then recompute the means.

4. In the following Naive Bayes Net suppose $k = 3$ and $dom(C) = \{1, 2, 3\}$. Describe the idea of EM.



EM computes $P(C|X_1, X_2, X_3, X_4)$ for $C = 1, 2, 3$.

Based on current $P(C)$ and $P(X_i|C)$, compute $P(C|X_1, X_2, X_3, X_4)$ for $C = 1, 2, 3$. Use this as partial data in ML learning.

5. What is an autoencoder? What are the two main components?

   An autoencoder maps inputs to low-dimension representation.

   It includes an encoder which maps inputs to low-dimensional representation and a decoder that maps inputs to its original representation.

6. What is a generative adversarial network (GAN)? What are the two main components?

   It is a generative unsupervised learning algorithm. The goal is to generate unseen examples that look like training examples. It includes generator and a discriminator.

9.1 Preferences and Utility

1. Suppose o_1 and o_2 are outcomes. Define the following
    a. $o_1 \succeq o_2$

       o_1 is at least as desirable as o_2

    b. $o_1 \sim o_2$

       o_1 and o_2 are equally desirable (indifference). $o_1 \succeq o_2$ and $o_2 \succeq o_1$

    c. $o_1 \succ o_2$

       $o_1 \succeq o_2$ and $o_2 \not\succeq o_1$

2. What are the following axioms for preferences and outcomes?
    a. Completeness
    b. Transitivity
    c. Monotonicity
    d. Decomposability ("no fun in gambling")
    e. Continuity
    f. Substitutability
3. What does it mean when we say an agent is rational?

9.5 Decision Processes

Markov Decision Process (MDP)

1. Define the following:
   a. infinite horizon

      The agent will not stop.

   b. indefinite horizon

      There is a stopping point, but we don't know when it will happen.

2. What does it mean when we say a model is stationary?

   When the probability of transition between states are the same at different times.

3. What are the components of a Markov decision process (MDP)

   A set of states S
   A set of actions
   A
   Probabilities/dynamics
   $P(S'|S, A)$
   Rewards
   $R(S, A, S')$

4. What is the difference between a fully observable Markov decision process and a partially observable Markov decision process (POMDP)

   In a POMDP, the states are hidden. Instead, observations are made based on the states.

5. Why does total reward only work when you can guarantee that the sum is finite?

   If the sum is infinite, you cannot compare two policies.

   If the sum is infinite, it does not give any opportunity to compare which sequence of rewards is preferable.

6. What does it mean when we say a policy is stationary?

   When there is a transition for every state.

   It assigns an action to each state.

Dynamic Decision Networks

1. Describe a dynamic decision network (DDN)

   This is like a decision network, but with actions and rewards.

10c Slides

1. Describe the idea of experience replay. What are the advantages?

   Store a buffer of previous experience and choose a batch of them to learn each time. The advantage is to minimize the correlations between successive updates (better learning). Also, fewer interactions with the environment are needed to converge.

2. Describe the idea of target network are needed to

   Maintain a separate network and update it occasionally.

11.3 Reinforcement Learning

Temporal Difference (TD)

1. Derive the temporal difference error $A_k = A_{k-1} + a_k(v_k - A_{k-1})$

Q-learning

1. Q-learning uses ____ to estimate the value of ____. It maintains a table of ____, where S is a set of states and A is the set of actions.

   ?, ?, $Q[S, A]$

   temporal difference, $Q^*[s, a]$

2. What does each entry in that table represent?

   The current estimate of the expected value of the policy

   The current estimate of $Q^*[s, a]$

3. The formula for updating each entry is

   $Q[s, a] \leftarrow (1 - \alpha)Q[s, a] + \alpha(r + \max_{a'} Q[s', a'])$

4. Under what assumption does Q-learning learn the optimal policy no matter which policy the agent is actually following?

   Q-learning learns the optimal policy as long as it tries all actions for a state infinitely, i.e. it does is not limited to a subset of actions

5. Define off-policy learning

   An agent learns the optimal policy no matter which policy the agent is actually following.

Exploration and Exploitation

1. Define exploration and exploitation

Exploration: try an action that is not optimal
Exploitation: perform the action that is known to be have a high reward

2. Describe the idea of ϵ-greedy strategy. What is the problem?

   Choose the optimal action with probability ϵ and explore another action with probability $1 - \epsilon$. The problem is that it treats all actions that are not optimal equally. It is more reasonable to choose the better ones among all sub-optimal actions

3. Describe the idea of the soft-max action selection.

   This is like the ϵ-greedy strategy, but it selects actions with higher Q values with higher probabilities.

4. In a soft-max action selection. What does τ represent?

   It is the temperature. When $\tau \to 0$, it selects the optimal action with higher probability (more like a greedy agent). When $\tau \to \infty$, it selects an action with equal probabilities.

5. Describe the idea of optimism in the face of uncertainty.

   Initialize the Q-values that encourage exploration.

On-Policy Learning

1. Define on-policy learning. Give an example.

   An agent learns the policy that it is actually following. SARSA is an example.

   SARSA uses empirical values for s' and a'

2. When is SARSA better than Q-learning?

   When the exploration might be dangerous.

Model-Based Methods

1. Describe the idea of a model-based reinforcement learning

   Model-based reinforcement learning learns the MDP and interleaves acting and planning. After each experience, it updates probabilities and the reward, then do some steps of asynchronous value iteration.