HW1

# Solution 1

Describe the principle of Dynamic Programming. Pick a problem of interest in Data Science which can be solved efficiently using Dynamic Programming. Describe the problem and the application of Dynamic Programming on it.

## Principles of Dynamic Programming

* Breakdown a big problem into sub-problems.
* Identify relationship between sub-problems.
* Store the results of the sub-problems.
* In case the sub-problem is encountered again use the stored results instead of solving it again.

## Application

### Sequence Labelling with Hidden Markov Models (HMMs)

* Involves assigning labels to each element in a sequence, based on set of hidden states and observed emissions.
* HMMs are probabilistic model that models sequences of observations, where each observation is associated with a hidden state and the entire system evolves over time.

#### Problem

* HMMs emits sequence of observations from states and estimating its probability.
* Main problem is to find the most probable sequence of states.

### Viterbi Decoding Algorithm

* A dynamic programming algorithm tailored for HMMs solves this problem.
* Parameters of HMM:
  + State transition probability matrix
  + Emission probability matrix
  + Initial probability vector

#### Initialization

* A basic data structure or a table of size T (elements in sequence) \* S (number of states).
* Stores probability of path taken at time t that ends in state s.

#### Start

* For the first element compute transition probability from a start state to each possible state.

#### Recursion

* Compute the probability of reaching each state at time t from all paths from prior step.
* Update the table with the probability of most likely path.

#### Termination

* Find most likely final state using transition probabilities from final to end state.

#### Backtracking

* Utilize the cached data in table to find the most likely sequences of states.

# Solution 2

Describe the main ingredients of local search. Pick a problem of interest in Data Science which can be solved efficiently using local search in a discrete space. Describe the problem and the application of local search on it. What happens if the search space is continuous?

## Ingredients of Local Search

* Hill Climbing/Descent (Greedy Local search):
  + Starts with an arbitrary assignment.
  + Move to next neighbor till reach a max/min where there is no neighbor with a higher/lower value.
  + There are chances that the algorithm may get stuck at neighborhood after finding a local optimum.
  + Quit if no neighbors are better than the current.
* Random Restart (outer loop):
  + To avoid getting stuck at a local max/min periodically run hill climbing multiple times from random location until local optimum if found.
* Noisy Strategy (inner loop):
  + Stochastic approach
  + Randomly choose from uphill moves depending on probability of the amt. of improvement.
  + Converges slower while it may find better solutions.
* Plateau Problem (Random walk)
  + When algorithm gets stuck in a neighborhood such it can neither move upwards or downwards.
  + Use a tabu list with limited use of memory to store the best seen neighbor from a random uniform choice of neighbor.
  + Continue for N iterations, with increase in N the random walk shall lead to global optimum.
* Simulated Annealing (Random Walk + Hill Climbing/Descent)
  + If the problem is set to decay exponentially
  + Accept an upward/downward move and reduce the acceptance rate proceeding further to escape the local optimum.
* In case of continuous search space
  + Utilize gradients to find the best direction.
  + Magnitude of gradient determines the direction of max improvement.

## Application

### Flight Route Planning

* Initialize: Initial set of available flight routes based upon origin, destination, air space regulations, and weather conditions.
* Neighborhood function: Define neighboring solutions depending upon route alternations possible, waypoints, altitudes.
* Objective function: Evaluate solution the maximizes the profit earned from flight depending upon fuel consumption, stop duration, parking costs, safety, and compliance with regulations.
* Selection criteria: Choose a route that maximizes the objective function meeting all the requirements.
* Termination criteria: Stop when a satisfactory solution is discovered.

# Solution 3

Describe the K-means clustering algorithm. Is it an Expectation-Maximization algorithm? Explain. What are some drawbacks of the K-means algorithm?

K-Means Clustering

* an iterative partitioning algorithm
* unsupervised learning method
* Group similar data points into distinct non overlapping clusters

### Algorithm

* Inputs:
  + K clusters
  + N data points
* Initialize: Randomly select K clusters
* Iteration:
  + Assign points to nearest cluster center using a distance metric.
  + Re-estimate the centroid by calculating the mean of points assigned to that cluster.
* Terminate:
  + If none of the points change cluster membership
  + Else re-iterate.

### Expectation-Maximization Algorithm?

* A simplified case of EM algorithm where only hard membership is assigned to the data points.
* In general EM is much more flexible allowing soft membership where a point can have a partial belonging to a cluster.

### Drawbacks

* Produces spherical clusters.
  + Cannot form elongated clusters or clusters of non-globular shapes.
  + Cluster may not be linearly separable.
* Problematic when are there clusters with different densities.
* Sensitive to outliers
* Cluster may overlap.
  + Probability that an object belongs to a cluster.
* Only work with geometric points.

# Solution 4

Describe the FastMap algorithm. What does it achieve and how does it do so? Comment on its efficiency. Describe two applications of FastMap in Data Science.

## FastMap

* A dimensionality reduction technique that projects high dimensional data to low dimensional space preserving the pairwise distances between objects.
* Used to represent objects as points in low dimensional space – creates K dimensional vector embeddings.

### Algorithm

* Inputs:
  + N objects
  + k – dimension of output vector
  + Pairwise distance function
* It heuristically selects two farthest objects in the dataset which acts as initial reference for projection of points into the low dimensional space.
* Runs in K iterations.
* Iteration **i**:
  + Calculate the distance of each data point from the reference line.
  + Each iteration gives the **i** th coordinate for the projected vector for each object.
  + Farthest pair is recomputed.
  + Pairwise distance function is corrected by component obtained by subtracting distance between the reference points.

### Efficiency

* Time Complexity: O(Nk)
  + Run in k iterations.
  + Each iteration in O(N) time.

### Applications

* Data fusion
  + Remote sensing: Combine data from different remote sensing sources like satellite with different resolutions, aerial imagery, LiDAR scans.
  + Speech and Image processing: Combine speech and image data to reduce dimensions of both modalities easing tasks like audio-visual recognition.
  + Efficiently reduces the dimension of data by aligning and fusing.
* Recommendation systems
  + Reduce the dimension of user-item interactions.